
RATS for Windows

Informal Manual

Written By "ERROR"

慶應義塾大学経済学部

伊藤幹夫研究会所属

1998年3月初版完成す

戯言

この資料は、「RATS for Windows [v.4.21] ¹」に付属のガイドツアーヘルプ (Ratstour.hlp) の一部を悠々自適自分勝手に意識したものです ²。そのため、数多くの誤訳が含まれていますし、勝手に著者自身の気心で原形をとどめていない部分もあり、さらに手心加えて付け足しなぞしています。また、ここでの解説は RATS の操作方法に関するものであって、決して計量経済学の説明ではないというのが悲しい事実です。それらの知識は各自がそれぞれ自主学習してください。こういう著者自身が計量経済学の知識にかなりの難点がありますので、知らず知らずのうちに嘘を撒き散らしています。本文内にそういったものを発見しましたら、「うわっ、こいつ馬鹿だな。」という優越感に浸りつつ、気にせず先へ読み進んでいってください。

1998 年 3 月
真実は必ず一つしかない
信じきる度胸がありますか

¹現時点 (1998 年 3 月) でのバージョンは 4.30 です。

²一昔前の言葉でいえば「超訳」です。あれは完全な誤訳本です。

目次

戯言	i
I RATS の動かし方	1
1 RATS って何？	2
2 RATS での作業	3
2.1 RATS の実行	3
2.1.1 バッチモード	3
2.1.2 インタラクティブモード	4
2.2 RATS エディタ	4
2.3 命令の実行	5
2.4 簡単な例	5
2.5 その他の簡単な例	6
2.6 プログラムの保存と実行	6
2.6.1 プログラムの保存	6
2.6.2 保存したデータの実行	6
2.6.3 ファイルフォーマット	7
3 RATS の基礎	8
3.1 RATS プログラムの基礎的要素	8
3.2 データセットの定義	8
3.2.1 プログラム例	8
3.2.2 その他の期種	9
3.3 データの読み込み	9
3.3.1 プログラム例	9
3.3.2 その他のデータフォーマット	10
3.3.3 データの期種変換	10
3.4 データの吟味	10
3.4.1 プログラム例	11
3.5 データ変換	12
3.5.1 プログラム例	12
3.5.2 その他の変換	12
3.6 OLS 回帰の推定	12

3.6.1	プログラム例	13
3.6.2	その他の推定例	14
3.7	仮説検定	14
3.7.1	プログラム例	14
3.7.2	その他の検定例	15
3.8	予測	16
3.8.1	プログラム例	16
3.8.2	その他の予測例	17
3.9	関数	18
4	その他の推定技術	19
4.1	副標本回帰	19
4.2	SUR	19
4.3	非線型最小 2 乗法	20
4.4	最尤推定法	20
5	グラフィックス	21
5.1	グラフの例	21
5.1.1	単純な時系列グラフ	22
5.1.2	スタックドバーグラフ	22
5.1.3	2 目盛グラフ	22
5.1.4	複数グラフ	22
6	プロシージャ	24
6.1	プロシージャの実行	24
6.2	プロシージャをつくる	24
7	最後に	27
II	組み込み関数	28
8	はじめに	29
8.1	数値に関する関数	30
8.1.1	一般的な数値論理関数	30
8.1.2	3 角関数	30
8.1.3	その他の数値関数	30
8.2	文字とラベルに関する関数	30
8.3	エントリーなどに関する関数	30
8.4	分布と確率に関する関数	31
8.5	行列に関する関数	31
8.5.1	配列関数	31
8.5.2	特別な乗法	31
8.5.3	配列のスカラー関数	32
8.5.4	スカラーの行列関数	32
8.5.5	スカラーと行列への包括的関数	32

8.6	プログラミングに関する関数	32
8.7	複素数に関する関数	32
9	最後に	34

Part I

RATS の動かし方

Chapter 1

RATS って何？

RATS は、包括的な時系列分析/計量経済学のソフトウェアパッケージです。幅広い範囲の課題に適しています。以下の項目は RATS でできることのほんの一部です。

- データセットを処理し、データの変換をします。
- 数多くの回帰モデルを推定します。線形/非線型最小 2 乗法、最尤法その他ができます。
- 利用方法によって実質的にあらゆる時系列技術を実行します。AR、MA、ARIMA、スペクトル分析その他ができます。
- 予測やシミュレーションができます。
- 高品質のグラフや散布図を得られます。

RATS の他にも、さまざまなソフトウェアパッケージが世に出まわっています。日本では、TSP が主に使われているような気がします¹。また、エクセルなどの表計算ソフトでもかなりの計算が可能です。こうしたものと比べて、RATS はどういうところが優位かというと、次の点が主なものだと思います。

1. 簡単な命令で多くの分析ができます。
2. Windows に完全対応していて、グラフ機能に優れています。

1 は、エクセルのように苛立つマクロを組む必要がありません。2 は、TSP のような貧弱なグラフしか表示できないことがなく、また、複数の分析結果を同時に表示させることができます。

数多くあるソフトウェアパッケージの中で、RATS は PC レベルではかなり優れたものであることには違いありません²。

¹早い時期から利用され、日本語のマニュアルなども数多く利用できるからでしょう。

²著者自身は Estima 社の回し者ではありません。

Chapter 2

RATS での作業

RATS はコマンドで動くプログラムです。つまり、基本的には特別なプログラミング言語を使いません。RATS を動かすには、一連の RATS の命令を実行します。

RATS 言語は、おおむね一般的な計量経済学/統計学的課題に対して、組み込み命令をもっています。例えば、回帰モデルの推定、データの予測、制約検定、その他多数利用できます。つまり、いくつかの単純なコマンドを使うだけで作業ができます。

とはいえ RATS は、より複雑な課題に対しても組み込みブロックとして使える他の命令もあります。

- スカラー計算や行列代数をおこなうための命令や関数があります。
- IF、WHILE、UNTIL 表現やループ命令といったプログラムの特徴をもちます。
- RATS 命令のサブルーチンをつくることができます。これはプロシージャ¹と呼ばれます。
- ユーザに情報を促し、また、ユーザにダイアログボックスからアイテムを選ばせることができる相互作用ルーチンのための命令があります。

こうした構成は、多くの一般的な（そして一般的でない）課題を簡単に実行でき、一方では、多くの柔軟性ももっています。用意されたルーチンの使用に制限がないので、多くの最新技術をすぐに補え、また、各自の新しい分析を展開することができます。

2.1 RATS の実行

RATS の実行には、バッチモードとインタラクティブモードの 2 種類あります²。

2.1.1 バッチモード

バッチモードでは、RATS はプログラムファイルにある命令のすべてを自動的に実行し、その結果を他のファイルに保存します。バッチモードは、実行に長時間かかりそうなプログラムや複数の短いプログラムをすばやく実行するときに特に有効です。

¹PROCEDURE と書きますが、カタカナ読みの正しい？ものが分かりませんのでこう呼んでおきます。

²著者自身は後者しか使ったことがありません。

2.1.2 インタラクティブモード

インタラクティブモードでは、RATS エディタ内で作業します。といってもそれは単なるテキストエディタです。パッチモードとは違い、次のようなことを編集集中にできます。

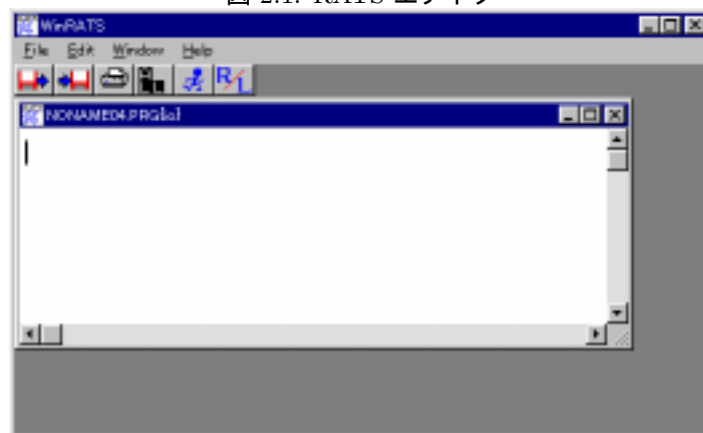
- RATS の命令を打ち込み、実行し、結果のアウトプットを検討します。
- それぞれの命令を修正して、再実行します。
- 完成した RATS プログラムを保存、編集、再実行します。
- アウトプットを検討、プリント、保存します。
- 時系列グラフや散布図を表示、検討、保存、プリント、エクスポートします。
- オンラインヘルプを使えます。

これらの機能によって、RATS と「インタラクティブ」に作業できます。つまり、完全なプログラムでなくても、さまざまな編集をおこなうことにより作業を続けていくことができます。こうしたことによつて、モデルやプログラムを発展させたり、ちょっとすぐに分析したいときなどに、時間の節約ができます。

2.2 RATS エディタ

RATS をインタラクティブモードではじめるためには、単に RATS のアイコンをダブルクリックします³。

図 2.1: RATS エディタ



プログラムがメモリにロードされると、RATS のウィンドウが表示されます。見た目は標準的なテキストエディタです。RATS は自動的に新しいウィンドウを開き⁴、「NONAME00.PRG」というタイトルをつけます。そのすぐ側に {io} というものが見えます。これは、このウィンドウがインプットとアウトプットのウィンドウになることを示しています。つまり、ここに命令を書き込み実行すると、同じウィンドウに結果も表示されます。もちろん、同じウィンドウに結果を表示させないこともできます。新しいウィンドウを開き、メニューバーの「Window」内の「Use for Output」を選択すると⁵アクティブなウィンドウがアウトプット用になります。

³DOS モードの RATS もありますが、DOS モードを使う利点が少ないため、今回ここでの説明は割愛します。

⁴設定によって開かなくすることができます。

⁵メニューバーやツールバーなどでアンダーラインが引いてある文字（ここでは「Window」など）は、「Alt」キーを押しながらそのキーを押すと選択されます。マウスでいちいち選択するのが面倒なときは便利です。

2.3 命令の実行

RATS 命令を実行することで RATS は動きます。RATS エディタ内で命令を実行するにはいくつかの方法があります。

- 命令を実行したいときは、インプットウィンドウ内に命令を打ち込み、<Enter> を押します。すると RATS はすぐに命令を実行し、アウトプットをウィンドウ内に表示します。
- すでに打ち込んでいる行を実行/再実行したいときは、カーソルをその行にもってきて、<Enter> を押します。または、<Enter> の代わりにツールバーの「Run」ボタン⁶を押します。
- 一連の命令を完全にすばやく実行したいときには、キーボードやマウスを使って実行したい命令を範囲指定します。それから <Enter> を押すか、「Run」ボタンを押します。

2.4 簡単な例

それではここで簡単な例をおこなってみます。ここでは簡単な計算と、その結果の表示をします。まず、アウトプットを別ウィンドウにします。そのために、次のようにします。

- 「File」内の「New」を選び、新しいウィンドウを開きます。
- そのウィンドウを前述のようにしてアウトプット用にします。

そしてウィンドウの配置を見やすくするために「Window」内の「Tile Horizontal」を選びます。

さて、そうしたらインプット用のウィンドウをアクティブにします（そのウィンドウの中をクリックします）。そして短いプログラムを書きます。はじめに a という実数値をとる変数をつくり⁷、それに 2.0 を代入します。次のように打ち込み <Enter> を押します。

```
compute a = 2.0
```

次に、さらに 2 つ「COMPUTE」命令を使って $b = 2.5$ をつくり、 c に $a * b$ ⁸ を代入します。最後に 2 つの「DISPLAY」命令で c と \sqrt{c} を表示します。合計で 5 つのステートメントがあり、アウトプットが「DISPLAY」命令により生まれます。⁹

```
*---input---
compute a = 2.0
compute b = 2.5
compute c = a * b
display 'c =' #.#### c
display 'Square root of c =' #.#### sqrt(c)
*---end---
```

⁶走る人間のボタンです。

⁷RATS では大文字と小文字の区別がありません。著者自身は小文字の方が書きやすいので小文字を使います。

⁸掛け算（ \times ）記号は $*$ を使います。

⁹以下の *---input--- という記述の類は、分かりやすくするために後から付け加えています。

```
*---output---  
c = 5.0000  
Square root of c = 2.2361  
*---end---
```

2.5 その他の簡単な例

今度は実際に、次のような短いプログラムを実行してみます。ここでは、日経平均の四半期データを読み込み、ARIMA(1,1,1)を推定するために「BOXJENK」命令を使っています¹⁰。

```
calendar 1955 1 4  
allocate 1996:4  
open data a:/japan.xls  
date(format=xls,org=obs) / jsrspa  
boxjenk(ar=1,diffs=1,ma=1) jsrspa / resids
```

さて、「BOXJENK」推定で得られたアウトプットを見た後、今度はAR(1)でなくAR(2)として推定したいとします。そのとき、わざわざからcalendar、allocate...などを実行しなくて良いのです。単に、「BOXJENK」命令内の適切なオプションを指定して実行すれば良いのです。

すべてを打ち直すのではなく、「BOXJENK」命令のみを直してから実行すればそれで終わりです。この場合、ただ単に「AR=1」オプションを「AR=2」に変えて<Enter>か「Run」ボタンを押せば、RATSが自動的にARIMA(2,1,1)を推定し、結果をアウトプットウィンドウに付け加えます。つまり、以前の結果と素早く比較することが出来ます。

2.6 プログラムの保存と実行

2.6.1 プログラムの保存

メニューバーの「File」内の「Save」「SaveAs」を使えば、後の利用のためにそのプログラムを保存できます。次のような点でとても便利です。

- 必要に応じて結果を再利用できます。
- 利用できるデータを使って同一のプログラムを実行できます。
- 新しい研究のためにプログラムの断片を再利用できます。

例えば、数ヶ月間隔のデータ用に回帰モデルと予測プログラムを組んでいるとします。ここでデータが月次で利用できることになったときは、必要な部分を書き換えて実行させれば済みます。

2.6.2 保存したデータの実行

インタラクティブモードで、すでに実行して存在するプログラムがあるときは、若干のキーボード操作やマウスクリックが必要です。

¹⁰ヘルプでは違うデータを利用していますが、これからは著者自身の利用できるデータを使い、同様の命令を試していきます。

1. プログラムファイルを開きます (「 File 」 内の 「 Open 」 を使います) 。
2. 「 Edit 」 メニューから 「 Select All 」 を選びファイル内のすべての行を選択します。
3. <Enter> か 「 Run 」 で実行します。

もちろん、バッチモードでもプログラムを実行できます。

2.6.3 ファイルフォーマット

RATS は、プログラムやアウトプットについて、標準的なテキストフォーマットを使用しています。分析結果を公開したいときも、プログラムやアウトプットを他のどんなワープロやページレイアウトソフトでも編集することができます。

もちろん、RATS 言語は基本的にすべてのプラットフォームで使えるので、Mac 上で打ち込んだプログラムを UNIX 上の RATS で実行することもできます¹¹。

¹¹ といってもたいていは一つのプラットフォームでしか作業しませんが ... 。

Chapter 3

RATS の基礎

この章では、もっともおなじみの RATS 命令をいくつか紹介し、それらを使って短いプログラム例をつくります。その一方で、これらの命令の詳細にはふれません。ここでは RATS 言語がどのように機能するか、そして、さまざまな課題を実行するためにどういう命令を組み合わせしていくかを提供していきます。

この先は RATS マニュアルの第 1 章にある基礎的段階に従います¹。

3.1 RATS プログラムの基礎的要素

ほとんどの RATS プログラムは次の段階に従っています。

1. データを記述します (例えば、期種、データの始点/終点などです)。
2. データをメモリに読み込みます。
3. データを吟味します。
4. 予備的変換をしておきます。
5. 分析を開始します (例えば、モデルの推定、仮説検定、予測その他です)。

3.2 データセットの定義

これから示すプログラム例では、いくつかのデータを読み込み、新しい系列をつくり、OLS 回帰をし、そして予測をします。データを用いるプログラムすべての第 1 段階は、RATS のためにデータセットを書きます。そのためには「CALENDAR」「ALLOCATE」命令を使います。

「CALENDAR」はデータの配置を決め、データの始点を特定化します。「ALLOCATE」はデータ系列の初期期間を決めます (クロスセクションデータを使用しているときは「CALENDAR」命令は省略できます)。

3.2.1 プログラム例

例えば 1955 年から 1996 年までの四半期データセットがあります。このときは次のように書きはじめます。

¹といっても、本当の紙のマニュアルはありませんので著者自身海のものとも山のものとも分かりません。あしからず。

```
calendar 1955 1 4
allocate 1996:4
```

1 行目は、RATS に 1955 年第 1 四半期から四半期データを使うことを教えています。2 行目はデータの終点が 1996 年第 4 四半期ということを示しています。

RATS でのデータ参照は *year:period* (年次、月次、四半期用)、*year:month:day* (その他ほとんどの期種) というようになります。年次データなら、期間は常に 1 です。1 年に 1 期しかないから当然です。

3.2.2 その他の期種

RATS は事実上いかなる時系列の期種も支援しています。年次、半年次、月次、四半期、週次、日次、さらにはパネルデータ (クロスセクション時系列データ) や 1 日に複数観測される日次データなどです。次のプログラムはいくつかの例です。

```
*---月次データ、2月から、1965年から1995年12月まで---
calendar 1965 2 12
allocate 1995:12
```

```
*---日次データ(月-金曜日まで)、1995年1月第2週から1995年12月29日まで---
calendar(daily) 1995 1 2
allocate 1995:12:29
```

```
*---上記と同じです。ただし週7日---
calendar(sevenday) 1995 1 2
allocate 1995:12:29
```

```
*---1年24期間、1950年始点、1995年終点---
calendar 1950 1 24
allocate 1995:24
```

RATS 命令の記述に関して、はじめの 3 文字だけで十分機能します²。例えば「CALENDAR」を「CAL」とし、「ALLOCATE」を「ALL」と打ち込んで同じ意味になります。慣れるととても便利です。

3.3 データの読み込み

次はディスクファイルからデータをメモリに読み込む必要があり、そのためには「OPEN DATA」と「DATA」命令を使います。「OPEN DATA」はデータを含んでいるファイル名を特定し、「DATA」によってそのファイルから実際に読み込み作業をおこないます。

3.3.1 プログラム例

「OPEN DATA」と「DATA」を付け加えると次のようになります。

²DOS からの名残でしょう。昔はメモリの制約がとてつもないものでした。

```
calendar 1955 1 4
allocate 1996:4
open data a:\japan.xls
data(format=xls,organization=observation) / gdp90 cp90 in90 cg90 mnqmacd cmbemtu
```

この場合、ここで使うファイルは japan.xls という名前で、 a:\japan.xls にあります。「 DATA 」命令内の「 FORMAT 」オプションは、このファイルは xls (EXCEL) 形式ということを示しています。「 ORGANIZATION 」オプションは、データが縦並びか横並びかという構成を示します。ここでは縦並びなので「 OBSERVATION 」を選んでいきます。記号 / は 1955 年から 1996 年までの初期期間データ範囲の代わりになります。 gdp90 cp90 ... はファイル内にある読み込みたい系列の名前です³。

もし「 OPEN DATA 」を省略すると、 RATS は「 DATA 」命令を実行するときにファイル名をプロンプトで尋ねてきます。

3.3.2 その他のデータフォーマット

RATS フォーマットは特別なデータフォーマットです。特に時系列データを扱うようにできています。ファイル内のデータを RATS フォーマットに変換するのは簡単ですが、特にそうする必要はありません。

RATS は他の多くのデータフォーマットからデータを読み込みます。テキスト、 1-2-3 / EXCEL スプレッドシート、 dBase ファイル、 Micro TSP ファイルなどです。「 COPY 」命令を使うと、これらのデータフォーマットにデータを書き出すこともできます。

次の例は、同じデータセットの中身を違うデータフォーマットから読みとります。はじめの例では系列名を省略してます。そうすると、 RATS はファイル内のすべての系列を読み込みます。

```
open data japan.wks
data(for=wks,org=obs)
```

```
open data japan.dat
data(for=free,org=obs) / gdp90 cp90
```

3.3.3 データの期種変換

「 DATA 」命令によって、短い期間から長い期間への期種変換をすることができます。初期段階では、 RATS はある期間のために数期間の平均をとります。数期間の合計や幾何平均をとったり、特定の数期間を選んだり、その他いろいろできます。例えば RATS は、月次を四半期に変換するとき、 3 カ月分を 1 四半期分にします。また、長い期間から短い期間へ補間するプロシージャ⁴も 2 つあります⁵。

3.4 データの吟味

データセットや打ち込んだプログラムに問題がないかを確認するために、メモリに読み込んだデータを吟味することを推奨します。これはデータ分析をはじめるときにとっても重要なことです。

³ちなみにデータの意味は、 gdp90 が国内総生産、 cp90 が消費支出、 in90 が総固定資本形成、 cg90 が政府支出、 mnqmacd は M2+CD、 cmbemtu はコールレート有担保翌日ものを示します。 90 という添え字があるものは、その要素が 1990 年基準で図られていることを示しています。 ... でもこれらの名前は著者自身の勝手な命名なので気にしないでください。

⁴サブルーチンの RATS での呼び方です。

⁵著者自身は知りません。

次の例では「TABLE」命令を使っています。これは現在メモリ内にあるすべての系列について、いくつかの基礎統計量を表示します。そして「PRINT」命令では系列の実際の値を表示します。

3.4.1 プログラム例

次の例⁶は「TABLE」と「PRINT」を含んでいます。

```
cal 1955 1 4
all 1996:4
open data a:\japan.xls
data(for=xls,org=obs) / gdp90 cp90 in90 cg90
table
print
```

この結果は次のようになります。ただし、かなりの量になっているので途中を省略します。

Series	Obs	Mean	Std Error	Minimum	Maximum
GDP90	168	61457.780357	34591.516165	10667.200000	130731.100000
CP90	168	36552.862500	19830.546092	7065.600000	75743.200000
IN90	168	17600.032738	11152.099807	1271.300000	41427.600000
CG90	168	6395.412500	2762.464696	2471.800000	11424.700000

ENTRY	GDP90	CP90	IN90	CG90
1955:01	10667.20000000	7178.600000000	1573.800000000	2554.400000000
1955:02	10685.10000000	7065.600000000	1271.300000000	2558.500000000
1955:03	11258.80000000	7404.400000000	1589.400000000	2482.700000000
1955:04	14463.60000000	9047.900000000	1734.100000000	2554.100000000
1956:01	11531.60000000	7831.900000000	1654.100000000	2558.900000000
1956:02	11769.60000000	7884.100000000	1559.600000000	2540.900000000
1956:03	12016.30000000	8082.300000000	1877.800000000	2480.300000000
1956:04	15285.00000000	9634.700000000	2026.800000000	2550.500000000
*---省略---				
1995:01	110715.20000000	65125.800000000	33441.200000000	11129.100000000
1995:02	108356.50000000	65768.500000000	29125.900000000	11219.000000000
1995:03	115358.40000000	69044.000000000	34764.400000000	10222.400000000
1995:04	127025.70000000	74230.700000000	39401.200000000	11065.400000000
1996:01	116114.10000000	68162.600000000	36881.900000000	11363.400000000
1996:02	111584.50000000	67769.900000000	31956.800000000	11424.700000000
1996:03	119440.70000000	70300.400000000	38326.100000000	10461.100000000
1996:04	130731.10000000	75743.200000000	41427.600000000	11384.000000000

もちろん、こうした系列はグラフにして吟味することもできます。

⁶以前も述べましたが、命令やオプションなどで省略可能なものは省略形で書いていきます。

3.5 データ変換

RATS での作業は、かなりの部分がデータの変換に関係します。これらはたいてい「SET」命令でおこないます。これによって新しい系列もつくれます。組み合わせは無量大にあるので、ここではいくつかの単純な例だけを示します。

3.5.1 プログラム例

例えば、線形のトレンド系列と、政府支出について 2 期間移動平均の系列が必要だとします。トレンドをつくるために、ここでは特別な変数「T」を使います。これは単に期間ごと数字を返します。この命令によって、第 1 期の TREND に 1.0 を、第 2 期の TREND に 2.0 を ... というようにセットされています。

```
set trend = t
```

平均を得るには次のようにします。

```
set avgcg = ( cg90 - cg90{1} ) / 2.0
```

ここで {1} という記述は 1 期のラグを示します。よって avgcg のそれぞれの期は、cg90 について現在の期と 1 期前を足したものを 2 で割ったものと等しくなります。

3.5.2 その他の変換

その他いくつかのおなじみの変換を示します。

```
*--- y の 1 階差---  
set dy = y - y{1}
```

```
*--- x の自然対数---  
set lx = log(x)
```

```
*--- .and. を使ったダミーで、第 2 次世界大戦近辺は 1、その他は 0 となる変数---  
set wardummy = t>=1942:1.and.t<=1946:1
```

3.6 OLS 回帰の推定

OLS (とそれに関係する) 回帰の推定についての基礎的的命令は、「LINREG」です。「LINREG」の一般的な構文は次の通りです。

```
linreg(option) dependentvariable start end residuals coefficients  
# list of regressors
```

ここでイタリックになっている項目はパラメタと呼ばれます。パラメタは、主に系列名を与えたりデータの始点や終点を示すために使われます。2 行目は # 記号ではじまります。これは補助カードと呼ばれ、

追加的情報を与えるために使われます。「LINREG」のために補助カードの右側の変数を特定します。

3.6.1 プログラム例

例えば、国内総生産を定数項/トレンド/消費支出/総固定資本形成/政府支出の 5 項目で OLS 回帰したいとします。そのときは次のように打ち込みます⁷。

```
linreg gdp90
# constant trend cp90 in90 cg90
```

このとき、gdp90 は被説明変数です。始点と終点を省略すると、RATS は回帰のためにすべてのデータを使います。residuals や coefficients も省略できます。これらは系列の残差や係数を保存しておきたいときに使います。そして補助カードに説明変数名を与えています。これらの命令を実行した結果の出力は次の通りです。

```
Dependent Variable GDP90 - Estimation by Least Squares
Quarterly Data From 1955:01 To 1996:04
Usable Observations      168      Degrees of Freedom      164
Centered R**2            0.999344      R Bar **2            0.999332
Uncentered R**2          0.999843      T x R**2             167.974
Mean of Dependent Variable      61457.780357
Std Error of Dependent Variable 34591.516165
Standard Error of Estimate      894.078401
Sum of Squared Residuals      131097694.82
Regression F(3,164)           83271.8457
Significance Level of F        0.00000000
Durbin-Watson Statistic       0.765728
Q(36-0)                       347.064499
Significance Level of Q        0.00000000

      Variable              Coeff      Std Error      T-Stat      Signif
*****
1.  Constant              -1114.560487      249.077793      -4.47475      0.00001425
2.  CP90                   1.440414         0.043607       33.03203      0.00000000
3.  IN90                   0.507618         0.045404       11.18012      0.00000000
4.  CG90                   0.154329         0.175933        0.87720      0.38165851
```

ぱっと見て分かるように、RATS は多くのアウトプットを表示します。ここに表示された統計量のほとんどは予約変数に貯えられます。これらはその後の計算で使えます。例えば、自由度は変数 %NDF に貯えられ、自由度修正済み決定係数は %RBARSQ で使えます。

RATS は回帰モデルでラグを特定化するのが簡単です。次の命令では、消費支出を定数/国内総生産の同期と 1 期ラグとで回帰しています。

⁷constant はすべての期が 1 である定数項になりますので、特に他のものを用意する必要はありません。

```
linreg cp90
# constant gdp90{0 1}
```

3.6.2 その他の推定例

例えば国内総生産を、定数項と政府支出の 0-15 期まで使って回帰したいとします。もちろんラグを 1 つ 1 つ別々に書くこともできます。けれども、表現「TO」を使うことで、より簡単にラグの範囲を打ち込めます。

```
linreg gdp90
# constant cg90{0 to 15}
```

すべての推定命令は多くのオプションをもっています。異なる種類の推定をおこない、出力を制御し、方程式を定義したりなどです。「LINREG」は多くのオプションを提供します。例えばその中の 1 つの「ROBUSTERRORES」というものは、不均一分散についての共分散行列を補正し、「INSTRUMENTS」は 2 段階最小 2 乗法に使用します。

```
linreg(robusterrors) gdp90 1986:1 1990:4
# constant mnqmacd{-4 to 8}
```

上記のプログラムでは、GDP を定数項/ 4 期リードから 8 期ラグをとる M2+CD で回帰しています。そして起こりうる不均一分散に対して共分散を補正しています⁸。

次のプログラムでは、まず「INSTRUMENTS」を用いて誘導形をつくり、cp90 に対して gdp90 と cp90 の 1 期ラグを用いた 2 段階最小 2 乗法推定をおこなっています。

```
instruments constant cp90{1} gdp90{1} cmbemt{1} gdp90
linreg(instruments) cp90
# constant gdp90 cp90{1}
```

3.7 仮説検定

RATS は多くの仮説検定についての組み込み命令をもっています。そして、その柔軟性によって他の多くの検定を遂行することもできます。ここではいくつかの単純な例を見ていきます。そして、より複雑な検定のためにどうやってプログラムを組んでいくかを示していきます。

3.7.1 プログラム例

次のプログラムでは、政府支出が回帰からはずせるか（係数が 0 となるか）の仮説検定を説明していきます。

⁸著者自身にこの意味は分かりません。

```

cal 1955 1 4
all 1996:4
open data a:\japan.xls
data(for=xls,org=obs) / gdp90 cp90 in90 cg90
table
print
set trend = t
set avgcg = ( cg90 - cg90{1} ) / 2.0
linreg gdp90
# constant trend cp90 in90 cg90{0 1}

```

まずはすべての説明変数を含んだもので回帰分析をおこないます。その後、「EXCLUDE」命令を使い、排除制約検定をおこないます。次のような命令を打ち込みます⁹。

```

exclude
# cg90{0 1}

```

結果は次のような形式で表示されます。見ると、cg90 と cg90{1} の係数が同時に 0 となることは、有意水準 0.00100184% で有意になるという結果になりました。つまり帰無仮説は強く棄却されています。

```

Null Hypothesis : The Following Coefficients Are Zero
CG90              Lag(s) 0 to 1
F(2,161)=         7.21080 with Significance Level 0.00100184

```

3.7.2 その他の検定例

RATS は他にもいくつかの命令をもっています。「TEST」命令は係数が特定のをとるかどうかを検定します。「RESTRICT」「MRESTRICT」は一般的な線形仮説を検定します。例えば次の例では、被説明変数 Y を X1 / X2 / X3 の 3 つで回帰したとき、「X1 と X2 の係数を足して 1.0 になる」「X2 の係数から X3 の係数を引くと 0 になる」という検定をおこないます。

```

linreg y
# x1 x2 x3
restrict 2
# 1 2
# 1.0 1.0 1.0
# 2 3
# 1.0 -1.0 0.0

```

次の例は、誤差項の不均一分散を検定するための方法です¹⁰。まずは OLS 回帰をおこない、その誤差を REDISD という系列に保存します。次に、誤差の 2 乗をそれ自身の 1 階差で回帰します。それを元に検定統計量を計算します。これは誤差の 2 乗の回帰から得られた「(観測数) × (決定係数)」です。最

⁹ここで「cg90{0 1}」と書かずに、「5 6」というように何番目の変数かという表現もできます。慣れるとこちらの方が使いやすいです。

¹⁰これは ARCH モデルというらしいです。詳しくは他の参考書などを見てください。

後に、「CDF」命令を使います。これは検定統計量の限界有意水準を計算します。「CDF」は $F/t/\chi^2$ /正規分布を扱えます。ここでは自由度 1 の χ^2 分布を使います。

```
linreg(noprint) cp90 / resids
# constant gdp90
set ressqr = resids**2
linreg(noprint) ressqr
# constant ressqr{1}
compute test_stat = %nobs * %rsquared
cdf chisqr test_stat 1
```

結果は次の通りです。どうやら均一分散ではないようでした。

```
Chi-Squared(1)=      14.341805 with Significance Level 0.00015244
```

3.8 予測

推定、検定の他にもなじみの課題はデータの予測です。RATS では、線形/非線型方程式、相互依存体系、方程式の完全同時体系といったものの予測ができます。ほとんどの予測では「FORECAST」命令を使います。ここではいくつかの例を簡単にふれます。

3.8.1 プログラム例

例えば、総固定資本形成/国内総生産/コールレートのデータ 1980:1 から 1989:4 を用いて、このデータから 1990:1 1990:4 の総固定資本形成を正確に予測できるかどうか知りたいとします。まずは総固定資本形成を定数項/国内総生産/コールレートで回帰したモデルをつくることにします。

まずはじめに回帰を推定します。ここで「LINREG」内で「DEFINE」オプションを使って方程式をつくり、変数名と推定係数を保存します。「FORECAST」はこれらの結果を予測計算のために使います。

```
linreg(noprint,define=in90eq) in90 1980:1 1989:4
# constant gdp90 cmbemtu
```

次に、gdp90 と cmbemtu の 1990:1 から 1990:4 までのデータを入力する必要があります。一度データファイルに書き込んでからまた読み直しても良いのですが、「DATA」命令の「UNIT=INPUT」オプションを使ってみます。これは RATS に直接データを打ち込めます。

```
data(unit=input) 1990:1 1990:4 gdp90 cmbemtu
101303.0 101312.9 108223.1 119146.3
8.40 7.18 6.45 5.84
```

それでは予測を開始します。次の命令では、前につくった方程式 (in90eq) を 1990:1 から 1990:4 までの 4 四半期予測します。予測したものは、ここでは in_forecast に保存されます。

```
forecast 1 4 1990:1
# in90eq in_forecast
```

さて、次に結果確認のために「PRINT」を使って観測地と予測値を表示してみます¹¹。

```
print 1989:1 1990:4 in_forecast in90 gdp90 cmbemtu
```

ENTRY	IN_FORECAST	IN90	GDP90	CMBEMTU
1989:01	NA	31005.600000000	97366.000000000	3.9091600000000
1989:02	NA	26575.000000000	95564.000000000	4.3608400000000
1989:03	NA	32438.700000000	102706.100000000	5.1832600000000
1989:04	NA	35986.100000000	113547.200000000	6.0353600000000
1990:01	31168.126018444	33234.700000000	101303.000000000	8.4000000000000
1990:02	30817.625175230	29362.600000000	101312.900000000	7.1800000000000
1990:03	33434.864263216	35168.500000000	108223.100000000	6.4500000000000
1990:04	37730.105694933	38918.800000000	119146.300000000	5.8400000000000

3.8.2 その他の予測例

「BOXJENK」を使って ARIMA モデルを推定してみます。そして 1990:1 から 1995:4 までの予測をおこなってみます。

```
boxjenk(noprint,define=bjeq,diffs=1,ar=1,ma=1) gdp90 1980:1 1989:4 resids
forecast 1 24 1990:1
# bjeq bjgdp
print 1990:1 1995:4 bjgdp gdp90
```

次のプログラムは「SYSTEM」命令を使って、4変数12期ラグのVARモデルを推定し、8期分の予測をします。ここでは、方程式は名前ではなく数字で参照します。RATSは、VARモデルの推定を単純にする命令をいくつかあります。「FORECAST」内の「PRINT」オプションを使うことで、RATSに予測値を表示させます。なお推定のパラメタのみ関心があるときは、「ESTIMATE」の「NOPRINT」オプションをつけることをお勧めします。そうしないと大量の分析結果がアウトプットに表示されてしまいます。

```
system 1 to 4
variables gdp90 cp90 in90 cg90
lags 1 to 12
deterministic constant
end(system)
estimate(noprint)
forecast(print) 4 8 1990:1
# 1 f_gdp90
# 2 f_cp90
# 3 f_in90
# 4 f_cg90
```

¹¹NA は Not Available の意味で、データが存在しないことを意味します。

3.9 関数

組み込み命令に加えて、RATS では 80 種類以上の関数を用意しています。配列の逆/転置、自然対数/平方根、観測地から年/月/日の取得、乱数発生などです。以下はその例です。

```
*---系列 X の平方根を系列 SQ_X の変換---  
set sq_x = sqrt(x)
```

```
*---行列 A について、A'A の逆行列を逆行列関数「 INV 」と転置関数「 TR 」で求めます---  
compute apainv = inv(a*tr(a))
```

```
*--- Y の COSINE を求めます---  
compute cosy = cos(y)
```

```
*--- 2 つの配列のクロネッカー積を求めます---  
compute kp = %kroneker(h,k)
```

Chapter 4

その他の推定技術

ここでは、RATS で利用できるその他の推定技術をいくつか簡単に紹介していきます¹。さらに詳しい情報は、RATS のマニュアル、または、オンラインヘルプの「[List of Instructions](#)」を参照してください。

4.1 副標本回帰

ほとんどの推定命令で利用できる「SMPL」オプションによって、推定のための副標本を簡単に選択できます。例えば、ある調査の項目で年齢があり、変数 AGE に格納されています。今度の課題では、30 歳を区切りにして回帰に差異があるかを比較したいとします。次のようにします。

```
set under = age<=30
set over = age>30
linreg(smpl=under) y
# x1 x2 x3
linreg(smpl=over) y
# x1 x2 x3
```

4.2 SUR

次の例では、2 つの方程式 (easteq と westeq) を定義し、それらを SUR (Seemingly Unrelated Regressions²) により推定します。

```
equation easteq yeast
# constant x1east x2east x3east
equation westeq ywest
# constant x1west x2west x3west
sur 2
# easteq
# westeq
```

¹この章については、内容的にかなり厳しいものがあります。翻訳も著者自身調べもせずに勝手に命名しているものもあり、信憑性にかけてます。

²—見無相関回帰とかいう訳しか浮かばなかったなのでこのままにしました。

4.3 非線型最小 2 乗法

次の例では CES 生産関数を推定するために³、非線型最小 2 乗法を用います。「NONLIN」命令を使い推定に使うパラメタを定義し、「FRML」命令を使い非線型モデルを定義します。そして「NLLS」で推定をおこないます（ここで Q は被説明変数です）。なお、プログラム内で \$ を使うことによって式を複数行に分割して書くことができます。

```
nonlin lgamma delta nu rho
frml ces = lgamma - ( nu / rho ) * $
          log(delta * k**(-rho) + ( 1.0 - delta ) * l**(-rho) )
nlls(frml=ces) q
```

4.4 最尤推定法

最尤推定法を用いて GARCH(1,1) を推定します。はじめの「MAXIMIZE」命令はいくつかの初期推定量を得るためにシンプレックス法を用います。次の「MAXIMIZE」では最終推定量を得るために BHHH 法 (Berndt、Hall、Hall、Hausman) を用います。モデルと対数尤度関数を特定化するため、ここでは単に親しみのある ARCH / GARCH モデルバリエーションについて示します⁴。

```
declare series u h
nonlin b0 vc va vb
frml resid = y - b0
frml hf = vc + va * h{1} + vb * u{1}**2
frml logl = ( h(t) = hf(t) ) , ( u(t) = resid(t) ) , $
           -0.5 * ( log(h(t)) + u(t) * ( u(t) / h(t) ) )
linreg(noprint) y / u
# constant
compute b0 = %beta(1)
compute vc = %seesq , va = 0.05 , vb = 0.05
set h = %seesq
maximize(method=simplex, iters=5, noprint) logl gstart gend
maximize(method=bhhh, recursive, iters=100) logl gstart gend
```

³ $\ln Y = \ln \gamma - \left(\frac{\rho}{\rho}\right) \ln [\delta K^{-\rho} + (1 - \delta)L^{-\rho}]$ とかいう有名な生産関数です ... とか説明しなくてもよいですね。でも著者自身は詳しく分かりません。

⁴この節については、まったくもって難解極まりないところなので何をどうしているか不明です。

Chapter 5

グラフィックス

RATS でグラフを描く際には、主に 2 つの命令があります。「GRAPH」では時系列グラフを扱い、「SCATTER」では XY 散布図を扱います。他にも「SPGRAPH」「GRTEXT」「GRPARM」がありますが、これらの命令は柔軟性が高く、次のようなことができます。

- 「GRAPH」によって、数種類の線画が引けます。
- 「SCATTER」によってシンボルのプロット、線付きシンボルプロットができます。
- カラー印刷ができます（ただし DOS 版はポストスクリプトデバイスカラーのみできます）。
- 白黒印刷のときに、カラーパターンを自動的に変換します。
- Y 軸に 2 つの異なる目盛をつけられます。
- 複数のグラフ/散布図を含むページをつくれます。
- ラベル/サブラベル/XY 軸ラベル/キーをつけられます。これらのラベルは大きさや見た目を制御でき、フォントも選べます（DOS 版でのフォントは、ポストスクリプトデバイスに従います）。
- RATS のグラフを他のソフトウェアで利用するために、多くのフォーマットへの書き出しができます。扱えるフォーマットは、ポストスクリプト/ウィンドウズメタファイル/マッキントッシュ PICT /ロータス PIC / HP プロットファイルフォーマットです。いくつかのフォーマットは特定のプラットフォーム用です。ウィンドウズ/マッキントッシュ版では、クリップボード経由でもグラフをコピーできます。

5.1 グラフの例

ここでは RATS の扱えるグラフのうち「GRAPH」が扱えるもののいくつかを紹介します。ちなみにここで使用したデータは次の通りです。

```
cal 1955 1 4
all 1996:4
open data a:\japan.xls
data(for=xls,org=obs) / gdp90 cp90 in90 cg90 jsrspa land6 mnqmacd cmbemtu
```

5.1.1 単純な時系列グラフ

```
graph(key=lolleft) 2
# gdp90
# cp90
```

5.1.2 スタックドバーグラフ

```
graph(style=stackedbar,header='INVESTMENTS & GOVERNMENT',$
      subheader='from 1955:1 to 1996:4',key=upleft) 2
# in90
# cg90
```

5.1.3 2 目盛グラフ

```
label jsrspa land6
# 'NIKKEI 225' 'LAND PRICES'
graph(patterns,key=below,header='STOCKS & LANDS',$
      style=polygonal,overlay=line,min=0.0) 2
# jsrspa / 2
# land6 / 1
```

5.1.4 複数グラフ

```
spgraph(header='Multiple Graphs',hfield=2,vfield=1,$
        vlabel='M2CD & Call Rate')
graph(key=upleft) 1 1 1
# mnqmacd
graph(key=upleft) 1 1 2
# cmbemtu
spgraph(done)
```

このように、RATS では簡単にグラフを作成することができます。グラフ機能についての詳細は、オンラインヘルプを参照してください。

図 5.1: 単純な時系列グラフ

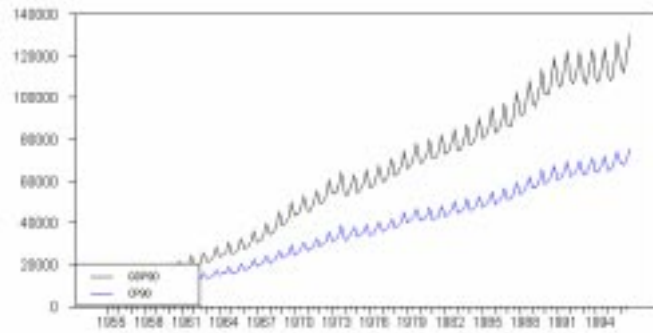


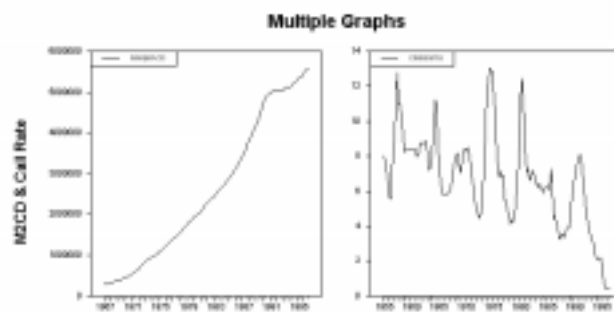
図 5.2: スタックドバーグラフ



図 5.3: 2 目盛グラフ



図 5.4: 複数グラフ



Chapter 6

プロシージャ

RATS プロシージャは RATS 命令のサブルーチンで、パスカルやその他プログラミング言語でのプロシージャやサブルーチンと同じです。実際に新しい命令を、他の一連の RATS コマンドを使って定義することができます。

プロシージャは、さまざまな種類の反復課題を単純化するのにとっても便利です。特に、いくつかの異なるモデルや変数に一連の命令を適応するような状況では有効です。また、別のユーザがプログラムを組むときに、プロシージャを使うために 1 つの「命令」を覚えるだけで済みます。

はじめから、RATS には約 30 種類のプロシージャがあらかじめ用意されています。それらの中には ARIMA モデルの推定/予測、単位根検定、その他多くのスペクトル分析技術がそろっています。これらは、各自がプロシージャをつくりたいときの良い見本にもなります。その他多くのユーザによって書かれたプロシージャは、WEB や BBS 経由でダウンロードすることができます。

6.1 プロシージャの実行

すでにあるプロシージャを実行するには、まずプロシージャを含む命令を RATS に実行させます。RATS では、プロシージャを「コンパイル」するといいます。今使っているファイルにプロシージャがなくて、プロシージャは別々のファイルに保存されているときには、「SOURCE」命令で読み込みます。その後プロシージャを指定します。

```
*---ソースを読み込む---  
source procedure.src  
*---プロシージャ指定---  
@procedure(option) parameters  
# supplementary cards
```

このように、プロシージャは普通の RATS 命令と同様にパラメタとオプションをとれます。補助カードも使えます。1 回プロシージャを読み込めば、それ以降いつでもどこでもなんどでも実行できます。

6.2 プロシージャをつくる

ここではプロシージャをつくってみましょう。仮説検定の節でふれた ARCH 検定の例をプロシージャにしてみます。はじめのものはそのまま打ち込んだ場合です。後のものは同じ検定をプロシージャにしたものです。なおプログラム内の行頭の * は、それ以降の文字を行末までコメントアウトするという意味です。

```

linreg(noprint) y / resids
# constant x1 x2 x3 x4
set ressqr = resids**2
linreg(noprint) ressqr
# constant ressqr{1}
compute test_stat = %nobs * %rsquared
cdf chisqr test_stat 1

```

```

procedure archtest depvar start end
type series depvar
type integer start end
local vector[integers] reglist
local series resids ressqr
local real test_stat
*
enter(varying) reglist
linreg(noprint) depvar start end resids
# reglist
set ressqr = resids**2
linreg(noprint) ressqr
# constant ressqr{1}
compute test_stat = %nobs * %rsquared
cdf chisqr test_stat 1
*
end procedure

```

この例にはプロシージャの基礎要素のほとんどが含まれています。

- 一行目の「PROCEDURE」命令によって、プロシージャの名前（ここでは ARCHTEST）とパラメタ（DEPVAR / START / END）を定義します。
- 「TYPE」命令によって、パラメタの変数タイプ（real / integer / vector / series など）を定義します。
- 「LOCAL」命令によって、変数の有効範囲をローカルにします。こうすると、変数の値はプロシージャ内のみで有効になり、メインプログラムにグローバル変数として渡されることはありません。
- 「ENTER」命令によって、補助カードに指定されていた変数名リストを読み込み、REGLIST という変数リストに保存しています。プロシージャ内では補助カードそれ自体がはじめにあるわけではないので、RATS はプロシージャを実行させる @procedure コマンドに続くものを代わりに使います。このことによって、プロシージャを実行させるたびに異なった変数セットを特定化することができます。
- はじめの「LINREG」のところ、明確な変数名を使わずに、パラメタ名と REGLIST 変数を使っています。

このプロシージャが書かれたファイルを ARCHTEST.SRC という名前で保存したとします。次のプログラムでは、いくつかの変数を読み込み、プロシージャも読み込み、それからいくつか異なるモデルに対してプロシージャを実行しています。

```
cal 1955 1 4
all 1996:4
open data a:\japan.xls
data(for=xls,org=obs) / gdp90 cp90 in90 cg90 mnqmacd land6
source(noecho) a:\archtest.src
@archtest gdp90
# constant cp90 in90 cg90
@archtest mnqmacd 1980:1 1996:4
# constant gdp90{0 1}
@archtest land6 1980:1 1996:4
# constant gdp90{-4 to 4}
```

このように、いくつかの異なるモデルに対して検定を簡単に適応できます。

Chapter 7

最後に

ここでガイドツアーはひとまず終わりです。この説明書が RATS に役立つことを願います。これから先の詳しい解説はオンラインヘルプを参照してください。

この資料では、RATS とは何かからはじまり、読み終わった後には RATS を用いて一通りの回帰分析ができることを目標にして書かれています。一番重要な章は「RATS の基礎」の部分です。後の部分はそれなりに読み飛ばしても差し支えありません、と信じています。

超訳にあたって、著者自身 RATS について大変勉強になりました。しかし、どうみてもいいかげんな説明や誤訳な部分を数多く出てしまいました。そういうところに関しては、まあ若気の至りということで見過ごしてやっておくんなまし。

1998 年 3 月
ふと辺りを見回してみると
誤差の多さに唖然とします

Part II

組み込み関数

Chapter 8

はじめに

ここでは、RATS が提供する多くの関数を簡単に紹介していきます。計算、情報の表示、複素数の課題のプログラミング、などです。適切な表現をすればどこでもこれらの関数を使えます。すべての関数の構文は次のようになっています。

関数名 (引数 1, 引数 2, ...)

関数名とその隣 (の間にスペースを入れてはいけません。それ以外に、これからの関数の表記には、特に注意がないときは基本的に次のような決まりを使います。

X and Y : 実数または実数値表現
M,N,and NUMBER : 整数または整数値表現
A and B : 配列または行列
Z : 複素数または複素数値表現

関数のカテゴリーは次のように分けられています¹。

数値に関する関数

文字列に関する関数

エントリーなどに関する関数

分布と確率に関する関数

行列に関する関数

プログラミングに関する関数

複素数に関する関数

この順番に簡単な説明をしていきます²。

¹カテゴリーの日本語訳は結構いいかげんですのであしからず。

²説明になっていないものもあります。

8.1 数値に関する関数

8.1.1 一般的な数値論理関数

ABS(X)	X の絶対値を返します。
%IABS(N)	整数 N の絶対値を返します。
EXP(X)	X の累乗を返します。
FIX(X)	実数値 X の切り捨てをして整数値を返します。
FLOAT(N)	整数値を実数値にします。
%IF(X,Y,Z)	X が非ゼロ (真) なら Y、ゼロ (偽) なら Z を返します。
LOG(X)	X の自然対数を返します。
SQRT(X)	X の平方根を返します。
%VALID(X)	X に値があれば 1、値がなければ 0 を返します。

8.1.2 3 角関数

COS(X)	コサイン X を返します。ただし X ラジアンです。
SIN(X)	サイン X を返します。ただし X ラジアンです。
TAN(X)	タンジェント X を返します。ただし X ラジアンです。

8.1.3 その他の数値関数

%ANNUITY(P,R,N)	年金の現在価値を返します。支払い P、利子率 R、期間 N は実数です。
%BOXCOX(X,Y)	Box-Cox 変換値を返します。
%FRAC(X)	[X] を返します。これは X をこえない最初の整数です。
%MAX(X,Y)	2 つの実数 X、Y の小さくない方を返します。
%IMAX(M,N)	2 つの整数 M、N の小さくない方を返します。
%MIN(X,Y)	2 つの実数 X、Y の大きくない方を返します。
%IMIN(M,N)	2 つの整数 X、Y の大きくない方を返します。
%PAYMENT(A,R,N)	年金の必要支払額を求めます。元金 P、利子率 R、期間 N は実数です。

8.2 文字とラベルに関する関数

%CONCAT(FIRST, LAST)	2 つのラベル変数を連結します。
%LEFT(LABEL, N)	LABEL の N 番目の文字を返します。
%MID(LABEL, M, N)	LABEL の M 番目から文字を N 個返します。
%RIGHT(LABEL, N)	最後から N 番目の文字を返します。
%STRING(N)	整数 N を文字として返します。

8.3 エントリーなどに関する関数

%CAL(YEAR, PERIOD)	YEAR の PERIOD に対する整数のエントリーナンバーを返します。
%CLOCK(M, N)	$[(M-1) \bmod N] + 1$ によって調整された値を返します。
%DATELABEL(T)	整数のエントリー T と対応する日付文字 (1991:10:1 など) を返します。
%DAY(T)	エントリー T の日付 (1-31) を返します。
%INDIV(T)	エントリー T がメンバー (パネルデータのみ) の個々の数字を返します。

%JULIAN(T)	1901年1月1日からエントリー T まで何日あるかを返します。
%MONTH(T)	エントリー T の月 (整数値) を返します。
%PERIOD(T)	個々に対応する時期 (パネルデータ)、日付 (イントラデイ) を返します。
%WEEKDAY(T)	エントリー T の曜日を数字で返します。
%YEAR(T)	エントリー T の年を返します。

8.4 分布と確率に関する関数

%RAN(X)	正規分布から乱数を返します。
%UNIFORM(L,H)	実数の下限 L、上限 H の間の一様分布から乱数を返します。
%CDF(X)	標準正規分布の $\Phi(X)$ ($P(X < z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$) を返します。
%CHISQR(X,R)	自由度 R の χ^2 分布の片側確率を返します。
%DENSITY(X)	標準正規密度関数です。
%LOGDENSITY(A,B)	対数標準正規密度関数です。
%FTEST(X,N1,N2)	F 検定です。
%INVCHISQR(X,R)	逆 χ^2 関数です。
%INVNORMAL(X)	逆標準正規分布関数です。
%LNGAMMA(X)	X のガンマ関数の自然対数値を返します。
%RANGAMMA(X)	シャープパラメタ X のガンマ分布から乱数を返します。
%MILLS(X)	ミル比を計算します。
%DMILLS(X)	逆ミル比のデリバティブを計算します。
%POISSON(MEAN,K)	ポアソン分布の値を返します。
%TTEST(X,N)	自由度 N の t 分布の両側確率を返します。
%ZTEST(X)	標準正規分布の両側検定をします。

8.5 行列に関する関数

8.5.1 配列関数

INV(A)	A の逆行列を返します。
TR(A)	A の転置行列を返します。
%DECOMP(A)	コレスキー分解をします。
%DIAG(A)	ベクトルから対角行列を返します。
%KRONEKER(A,B)	A と B のクロネッカー積を返します。
%XDIAG(A)	配列 A の対角成分を $n \times 1$ 長方形配列にします。
%XROW(A,R)	配列 A の列 R を $n \times 1$ 長方形配列にします。
%XCOL(A,C)	配列 A の行 C を $n \times 1$ 長方形配列にします。

8.5.2 特別な乗法

%KRONID(A,B)	$(A \otimes I) B$ を返します。
%MQFORM(A,B)	$B'AB$ を返します。
%MQFORMDIAG(A,B)	$B'AB$ の対角成分を返します。

8.5.3 配列のスカラー関数

%DET(A)	行列式を返します。
%SUM(A)	要素の和を返します。
%TRACE(A)	A のトレースを返します。
%MAXVALUE(A)	A の最大値を返します。
%MINVALUE(A)	A の最小値を返します。
%DOT(A,B)	A と B の内積を返します。
%COV(A,B)	A と B の共分散を返します。
%CORR(A,B)	A と B の相関係数を返します。
%QFORM(A,B)	$B'AB$ を返します。B は $N \times 1$ です。
%SCALAR(A)	$A(1,1)$ または $A(1)$ を返します。

8.5.4 スカラーの行列関数

$A = \%RAN(x)$	平均 0、標準偏差 x の標準正規分布からの値をうめます。
$A = \%UNIFORM(x1, x2)$	$x1$ と $x2$ の一様分布からの値をうめます。
$A = \%CONST(x)$	x ですべてうめます。
$A = \%MSCALAR(x)$	単位行列 $\times x$
$A = \%IDENTITY(n)$	$n \times n$ の単位行列をつくります。

8.5.5 スカラーと行列への包括的関数

%LOG(A)	要素ごとに自然対数をとります。
%EXP(A)	要素ごとに指数乗をとります。
%ABS(A)	要素ごとに絶対値をとります。
%SQRT(A)	要素ごとに平方根をとります。
%ROWS(A)	A の列の数を返します。
%COLS(A)	A の行の数を返します。
%SWEEP(A,k)	ピボット k について A をスイープします。

8.6 プログラミングに関する関数

%DEFINED(NAME)	プロシージャのパラメタやオプションの状態を返します。
%EQNCOEFFS(EQ)	方程式 EQ の推定係数ベクトルを返します。
%EQNDEPVAR(EQ)	方程式 EQ の被説明変数を返します。
%EQNSIZE(EQ)	方程式 EQ の # にある説明変数を返します。
%EQNTABLE(EQ)	方程式 EQ の説明変数のリストの $2 \times N$ 配列を返します。
%LABEL(variable)	ナンバー系列を使っているなら、特定の変数にあたるラベルを返します。

8.7 複素数に関する関数

%ARG(Z)	複素数 Z の引数を返します。
%CABS(Z)	複素数 Z の絶対値を返します。
%CEXP(Z)	e^Z を返します。
%CLOG(Z)	Z の複素数自然対数を返します。

%CMPLX(R,I)	実数 R と I を複素数に変換します。 R が実部で I が虚数部です。
%CONJG(Z)	Z の複素数結合を返します。
%CSQRT(Z)	Z の複素数平方根を返します。
%FREQSIZE(N)	「 FREQUENCY 」命令のために縦軸の数を返します。
%IMAG(Z)	Z の虚数部を返します。
%REAL(Z)	Z の実部を返します。
%UNIT(X)	単位円値 $\text{EXP}(iX)$ を返します。
%UNIT2(T1,T2)	単位円値 $\text{EXP}\left(\frac{i(T1-1)}{T2}\right)$ を返します。

Chapter 9

最後に

最後の最後ですがお詫び申し上げます。

この第 2 部で関数の説明をしようと思い、いざ書きはじめてみるとまるでわからない部分ができました。分布と確率以降の説明は、著者自身読み返してみても非常に理解に苦しむものになってしまいました。やはり力量不足は否めませんでした。そして本来なら索引なんかもつけるべきなのでしょうが、そこまで無償でするのはつらいということで勘弁してください。

ともかくにも、「RATS for Windows Informal Manual」の初版が完成を迎えることになりました。これも偏に著者自身の暇の賜物です。... 就職活動はどうしたんだって？ 聞きっこなしですよ旦那 ... 。

1998 年 3 月
世界の終わりも知らぬ間に
通り過ぎていきそうですな