

**AREMOS/Windows**

经济统计数据库系统

入门讲义

(第五版)

郑汉川 梁明义 编着

财团法人经济资讯推广中心 编印

**AREMOS/Windows**

经济统计数据库系统

入门讲义

有关 AREMOS 软件包以及数据库系统使用如有疑问

请电话：(02)2366-1944 找 AREMOS 管理师

或传真：(02)2366-0403

或E-mail：[TEDC1@MS12.HINET.NET](mailto:TEDC1@MS12.HINET.NET)

或来信：106 台北市辛亥路一段 96 号 3 楼

财团法人经济资讯推广中心收

本手册亦置于因特网本中心网址上，欢迎在线查询。本中心网址是：

<http://www.edu.tw/moecc/rs/pkg/tedc.htm> 或于各大入口网站(Yam、Yahoo、Pchome...)敲入「财团法人经济资讯推广中心」或「aremos」搜寻到本中心网址。

「AREMOS 经济统计数据库系统」之建立与平时更新承蒙下列诸单位提供数据鼎力协助，在此特表谢忱：

<u>数据库名称</u>	<u>提供单位</u>
台湾地区国民所得统计数据库	行政院主计处第三局
台湾地区工业生产统计数据库	经济部统计处、经济部信息中心
台湾地区进出口贸易统计数据库	财政部统计处 海关总税务司署信息管理中心
台湾地区金融统计数据库	中央银行经济研究处 中央银行信息室
台湾地区资金流量统计数据库	中央银行经济研究处 中央银行信息室
台湾地区国际收支平衡统计数据库	中央银行经济研究处 中央银行信息室
台湾地区资 IMF IFS 格式金融统计数据库	中央银行经济研究处 中央银行信息室
台湾股票市场统计数据库	台湾证券交易所 行政院国家科学委员会 交通大学管理科学研究所 中华民国证券市场发展基金会 台凤证券股份有限公司
台湾上市公司股票报酬率统计数据库	台湾证券交易所 台湾大学财务金融学系、会计学系

台湾股票上市公司财务报表数据库	台湾证券交易所 台湾大学财务金融学系、会计学系
台湾上柜公司股票报酬率统计数据库	财团法人中华民国证券柜台买卖中心
台湾柜台买卖统计数据库	财团法人中华民国证券柜台买卖中心 台凤证券股份有限公司
台湾股票上柜公司财务报表统计数据库	财团法人中华民国证券柜台买卖中心 台湾大学财务金融学系、会计学系
台湾地区人力资源统计数据库	行政院主计处第四局
台湾地区能源资源统计数据库	经济部能源委员会
中华民国教育统计数据库	教育部统计处
台湾地区薪资与生产力统计数据库	行政院主计处第四局
中华民国财税统计数据库	财政部统计处 财政部财税数据中心管理信息作业小组
台湾地区物价统计数据库	行政院主计处第三局
中华民国交通统计数据库	交通部统计处
台湾地区农业统计数据库	行政院农业委员会 台湾省政府农林厅
台湾总体经济预测数据库	行政院主计处第三局 行政院经济建设委员会 中央研究院经济研究所
台湾地区多因素生产力统计数据库	行政院主计处第四局
台湾地区国民经济动向统计季报数据库	行政院主计处第三局

大陆地区经济统计数据库	中国国家统计局
台湾地区保险市场统计数据库	保险业务展基金管理委员会
台湾地区期货市场统计数据库	台湾期货交易所
台湾地区商业动态统计数据库	经济部统计处
台湾地区景气指针统计数据库	经济部投资审议委员会 内政部统计处 经济部统计处 行政院经济建设委员会经济研究处 中央银行金融业务检查处 财政部金融局统计室

财团法人经济资讯推广中心谨上

# 目录

## 第一部份 检索数据与基本操作

第一章	「AREMOS/Windows 经济统计数据库系统」简介	
1.1	系统概述 .....	11
1.2	硬件、数据库传递方式及收费标准 .....	13
1.3	系统特色 .....	14
1.4	AREMOS/Windows 与 AREMOS/UNIX 之不同 .....	15
1.5	如何学习使用 AREMOS/Windows 数据库系统 .....	15
第二章	基本检索数据方法	
2.1	前言 .....	17
2.2	AREMOS/Windows 基本操作程序 .....	17
2.3	范例(一): 基本检索数据方法 .....	20
2.4	范例(二): 如何 Export 或 Send 数据以便用于其它软件 .....	28
2.5	范例(三): 如何检索数据转换频率并做回归分析 .....	36
第三章	有关检索代号(RetrievalCode)	
3.1	数据库检索手册 .....	42
3.2	在线(ONLINE)查询检索代号 .....	45
3.3	「INDEX 在线查询法」 .....	46
3.4	「Seriescode 在线查询法」 .....	50
3.6	利用本中心的 Homepage 网页, 在线查询检索代号 .....	51
第四章	股票市场数据库之特殊用法	

4.1	股票市场数据库使用上的特点 .....	52
4.2	范例(四): 检索时间序列日数据 .....	53
4.3	范例(五): 跨数据库之时间序列检索方法 .....	53
4.4	范例(六): 转换日数据之频率并做回归分析 .....	56
4.5	有关「上市公司股票报酬率统计数据库」 .....	59
第五章	如何使用打印机(PRINTER)	
5.1	使用鼠标按键操作 .....	60
5.2	使用<Alt><Print Screen>键 .....	60
5.3	sf(Save-File)与 sfc(Save-File-Completed).....	60
5.4	有大量结果而不愿在荧光幕上出现者: bf(Bigsave-File)与 bfc(Bigsave-FileCompleted).....	62
第六章	常用指令之说明	
6.1	General 与 Object Commands.....	63
6.2	Global 与 Local Setting.....	64
6.3	跨行下指令或同一行下多数指令 .....	65
6.4	加减乘除与开方 .....	66
6.5	Work Bank 的说明 .....	66
6.6	Series 的「完整名称」 .....	67
6.7	如何产生 Series.....	68
6.8	AREMOS Functions.....	71
6.9	「说明」(Document).....	72
6.10	如何建立 Databank.....	72
6.11	index,delete 与 clear.....	73
6.12	rename....as.....	74
6.13	小数点与 report columns 的设定 .....	74
6.14	help 与 query.....	75
6.15	Option 设定的储存: store 与 restore.....	76
6.16	在 AREMOS/Windows 里进行编辑 .....	76
6.17	「递延」(lag)与「领先」(lead).....	77
第七章	LIST	
7.1	使用 list 指令来产生 list.....	78
7.2	使用 index 指令来产生 list.....	81
7.3	使用 Set Operator 来产生 list.....	85

第八章	PROCEDURE 与 OBEY	
8.1	使用 Procedure Edit and Display 编辑程序产生 Procedure.....	88
8.2	使用 Procedure 指令产生 Procedure.....	91
8.3	CMD file 与 obey.....	91

第九章	stocklist、complist、、compflist、compreport、compfreport 之使用	
9.1	stocklist.....	95
9.2	complist、compflist.....	96
9.3	compreport、compfreport.....	96

## 第二部份 分布图，制表，绘图

第十章	分布图(Scatter&Plot)简介	
10.1	范例.....	98
10.2	范例.....	99

第十一章	AREMOS/Windows Table 制表的简介	
11.1	表格步骤.....	101
11.2	表格画面.....	101
11.3	基本表格的设计.....	103
11.4	数据取得和设定时间.....	104
11.5	编辑报表及数学计算公式.....	107

第十二章	AREMOS/Windows Graphic 绘图简介	
12.1	绘图步骤.....	110
12.2	AREMOS Graph 「Help」.....	114

## 第三部份 循环及计量模型

第十三章	循环的使用(for-end)	
13.1	循环之使用例(甲).....	116

13.2	循环之使用例(乙).....	117
13.3	循环之使用例(丙).....	117
第十四章 计量模型之建立与求解		
14.1	基本概念 .....	120
14.2	模型建立 .....	121
14.2.1	估计方程式 .....	121
14.2.2	编译及求解 .....	125
14.3	样本外预测 .....	128
14.4	外生变数仿真 .....	130
14.5	加入 Addfactors.....	132
14.6	产生 Implicit Addfactors.....	135
14.7	Large Model 解法 .....	138
14.7.1	样本外预测 .....	138
14.7.2	外生变数仿真 .....	140
14.7.3	加入 Addfactors.....	142
14.8	结果陈示 .....	143
14.9	指令摘要 .....	145
14.9.1	check.....	145
14.9.2	solve.....	145
14.9.3	model.....	147
附录(一): 「AREMOS/Windows 经济统计数据库系统」		
	安装使用收费标准 .....	149
附录(二): ※「键入指令法」(在 AREMOSCommand 画面下).....		
		150

# 第一部份

## 检索数据与基本操作

# 第一章 「AREMOS/Windows 经济统计数据库系统」简介

## 1.1 系统概述

教育部自民国 76 年以来，委托国立台湾大学在其 IBM 主机上建立「EPS 经济统计数据库系统」，此系统广为国内大专院校老师、学生使用。然近年来计算机市场变化极快，大型主机逐渐被工作站与个人计算机取代。为顺应此潮流，教育部于民国 84 年 6 月停用 IBM3090 主机，而以 IBM RS6000 工作站来替代。为能继续提供该「数据库系统」的维护与使用，我们选定 AREMOS/UNIX 软件来取代 EPS。此外，为方便在个人计算机上使用这些数据库，并开放这些数据库给工商界使用，乃发展出 PC-Windows 版的「AREMOS/Windows 经济统计数据库系统」，同时也成立了「财团法人经济资讯推广中心」，并提供必要之相关性服务。

\*截至目前，共有下列 29 个数据库可供 PC 上使用：

(一)国内「一般性」统计数据库(共 14 个)

- (1)「台湾地区国民经济动向季报统计数据库」
- (2)「台湾地区国民所得统计数据库」
- (3)「台湾地区工业生产统计数据库」

- (4)「台湾地区进出口贸易统计数据库」
- (5)「台湾地区人力资源统计数据库」
- (6)「台湾地区能源统计数据库」
- (7)「台湾地区教育统计数据库」
- (8)「台湾地区薪资与生产力统计数据库」
- (9)「台湾地区财税统计数据库」
- (10)「台湾地区物价统计数据库」
- (11)「台湾地区交通统计数据库」
- (12)「台湾地区农业统计数据库」
- (13)「台湾总体经济预测数据库」
- (14)「台湾地区多因素生产力统计数据库」

(二)国内「金融性」统计数据库(共 12 个)

- (15)「台湾地区金融统计数据库」
- (16)「台湾地区资金流量统计数据库」
- (17)「台湾地区国际收支平衡表统计数据库」
- (18)「台湾地区 IMFIFS 格式金融统计数据库」
- (19)「台湾股票市场统计数据库」
- (20)「台湾上市公司股票报酬率统计数据库」
- (21)「台湾地区股票上市公司财务报表统计数据库」
- (22)「台湾柜台买卖统计数据库」
- (23)「台湾上柜公司股票报酬率统计数据库」
- (24)「台湾股票上柜公司财务报表数据库」
- (25)「台湾地区保险统计数据库」
- (26)「台湾地区期货统计数据库」

(三)大陆地区经济统计数据库(共 1 个)

(27)「大陆经济统计数据库」

(四)台湾地区商业动态与景气指针统计数据库(共 2 个)

(28)「台湾地区商业动态统计数据库」

(29)「台湾地区景气指针统计数据库」

本入门讲义是介绍 PC-Windows 版的「AREMOS/Windows 经济统计数据库系统」使用方法。如果您已熟悉 AREMOS/UNIX 的操作,那些「键入指令」的使用方法在 AREMOS/Windows 里也都可以适用。但 AREMOS/Windows 可鼠标、窗口等,要简便多了。

## 1.2 硬件、数据库传递方式及收费标准

本系统可装于一般 IBM 兼容性个人计算机。Hardware 586 or above。Memory 至少 32MB。Windows 的 version 95 or above。

数据库约占 600MG 空间。有两种方法存放这些数据:(1)如放在硬盘,则至少需有 700MG 硬盘,以便其中 600MG 空间来放数据。(2)如有只读式光盘(CD-ROM)的光驱则所有数据库可放入一片 CD-ROM。

数据库每天不断更新。每个月一次邮寄一片含最新更新数据之光盘片给 Users。

此系统之安装使用,每年收费新台币 11 至 13 万元不等(学校系所、老师、学生按 70%教育折扣收费),不含国外数据库。此项收费包括:(1)AREMOS/Windows 软件包及 29 个数据库之安装使用。(2)每个月一次更新数据库。(3)提供使用手册及使用上之咨询服务。如果想增加更新数据库次数(例如每周一次),或同一单位想增加安装份数,另有

收费标准，请参考附录(一)。

其中国外数据库亦可透过本中心向国外购买(本中心也是美国 **Global Insight** 环球透视公司代理商)，数据亦是 **AREMOS** 格式可读取，另有收费标准，请询问本中心。

## 1.3 系统特色

本数据库系统特色可综合下列五点：

(1)「检索方法简便」：**AREMOS/Windows** 检索指令简单，大概只需把鼠标移至右下角您所想要 **SET frequency** 及 **period** 的地方去改一改，再把鼠标移至工具列中的打开档案处，去 **OPEN banks**，即可叫取到所需的数据。

(2)「完整时间序列数据」：一般统计期刊常仅列出最近几月或最近几季之数据，往前几年则以年数据出现。如要完整时间序列月数据或季数据，必得从不同书上逐一抄写下来。使用 **AREMOS/Windows** 数据库，可方便地以鼠标把完整时间序列叫取出来。

(3)「更新迅速」：本数据库系统，尽量与发布数据机构建立直接管道。例如「台湾地区国民所得统计数据库」，每次主计处国民所得评审会议通过后，约两、三天内我们的数据库即可更新。而统计期刊常要一、两个月后才能出版寄到各图书馆。

(4)「数据库与运算分析功能的结合」：使用 **AREMOS/Window** 数据库最大特点，在于数据库与运算分析功能的结合。研究者可立即在「工作空间」内就数据库之数据进行各种分析，而不需做任何转换或再输入。**AREMOS/Windows** 的运算功能，从简单的回归、复杂的联立方式等等，均有具备。

(5)「配合使用其它软件」：虽然 AREMOS/Windows 具有相当完整的运算功能，不少研究者仍习惯于自己熟悉的软件。

AREMOS/Windows 亦有现成指令，可把数据输出让其它软件使用。

## 1.4 AREMOS/Windows 与 AREMOS/UNIX 之不同

由于不少使用者已熟悉教育部电算中心 RS 6000 工作站上之「AREMOS/UNIX 数据库系统」，在此特别简列 AREMOS/Windows 和其它两种数据库系统之不同如下：

(1)AREMOS/Windows 是 PC-Windows 版软件，而在 PC 上绘图、制表、窗口、使用打印机等方面之方便，是工作站上的 AREMOS/UNIX 远所不及。

(2)有不少使用者惯常把 AREMOS/UNIX 数据 download 到 PC 来使用(透过 FTP 档案传输方式)。现在 AREMOS/Windows 系统可直接透过窗口工具列(send series data) 把数据送到同一 PC 上，因此不需经过 download 过程。

(3)使用 AREMOS/Windows 系统可在自己的环境下编辑 (AREMOS Editor)，但 AREMOS/UNIX 必须学习 UNIX 里的 vi 编辑程序。

(4)教育部免费提供 AREMOS/UNIX 数据库系统供大家使用(但先决条件您必须是有教育部账号的研究生及教授们)，而安装此 AREMOS/Windows 系统则需收取费用。请参考附录(一)至(四)。

## 1.5 如何学习使用 AREMOS/Windows 数据

# 库系统

要学习使用 **AREMOS/Windows** 数据库系统有下列几种方式：

(1)使用 **AREMOS/Windows** ，一定要有一本

「**AREMOS/Windows** 入门」讲义或速成讲义在手边。此书含基本操作方法、计量回归分析、**AREMOS/Windows** 操作方法说明等等，是学 **AREMOS/Windows** 最基础的入门读物，也是操作时必须的参考手册。

(2)除了「**AREMOS/Windows** 入门」讲义外，每个数据库均有一本「检索手册」(注：本中心已将所有检索手册放置在中心网站，供大众免费下载，无锁任何 IP)。这些手册简单介绍各个数据库内容，并把数据库所含每笔数据之「检索代号」(retrieval code)列举出来。查出您所需数据之「检索代号」，再依「**AREMOS/Windows** 入门」所介绍之检索数据方法，便可上机操作。

(3)本中心为您安装 **AREMOS/Windows** 软件包与数据库系统后，也会安排一个研讨会来讲解 **AREMOS/Windows** 基本操作。

(4)**AREMOS/Windows** 有二本原文手册，使用者可自行研修：

「**AREMOS 5.2 for Windows User's Guide**」，「**AREMOS 5.2 for Windows Training Manual**」或自行进入计算机 **AREMOS** 系统环境的主画面下，用鼠标点取「**Help**」栏项，如 **Homepage** 一页页的查询。或把鼠标停放在某个窗口上后，按键盘「**F1**」功能键，亦可一目了然。

(5)平时使用 **AREMOS/Windows** 数据库或 **AREMOS/Windows** 软件如有疑问或建议，请来电话(02)2366-1944 或传真 (02)2366-0403 找 **AREMOS** 管理师。或来信：「106 台北市辛亥路一段 96 号 3 楼财团法人经济资讯推广中心 收」。

# 第二章 基本检索数据方法

## 2.1 前言

本章将已三个实际范例说明如何此用 **AREMOS/Windows** 软件包检索数据，并进行简单回归分析。大部分研究者的需求，可依这些范例套上自己所需数据来进行。如果您不在乎了解 **AREMOS/Windows** 语言架构，本章的初步说明应该足够。如果您有兴趣进一步了解 **AREMOS/Windows**，在第六章将回头有系统地解释这些指令。

## 2.2 **AREMOS/Windows** 基本操作程序

在未开始介绍基本检索数据的方法前，先说明 **AREMOS/Windows** 软件之基本操作程序：

### 1. 进入及离开 **AREMOS/Windows**

当 **AREMOS/Windows** 安装(install) 好了以后，即可进入使用操作。首先进入 **Windows** 环境，找找桌面 **AREMOS** 接口，按下鼠标左键两下，即可进入 **AREMOS/Windows** 环境下，即屏幕出现如图 2.2:

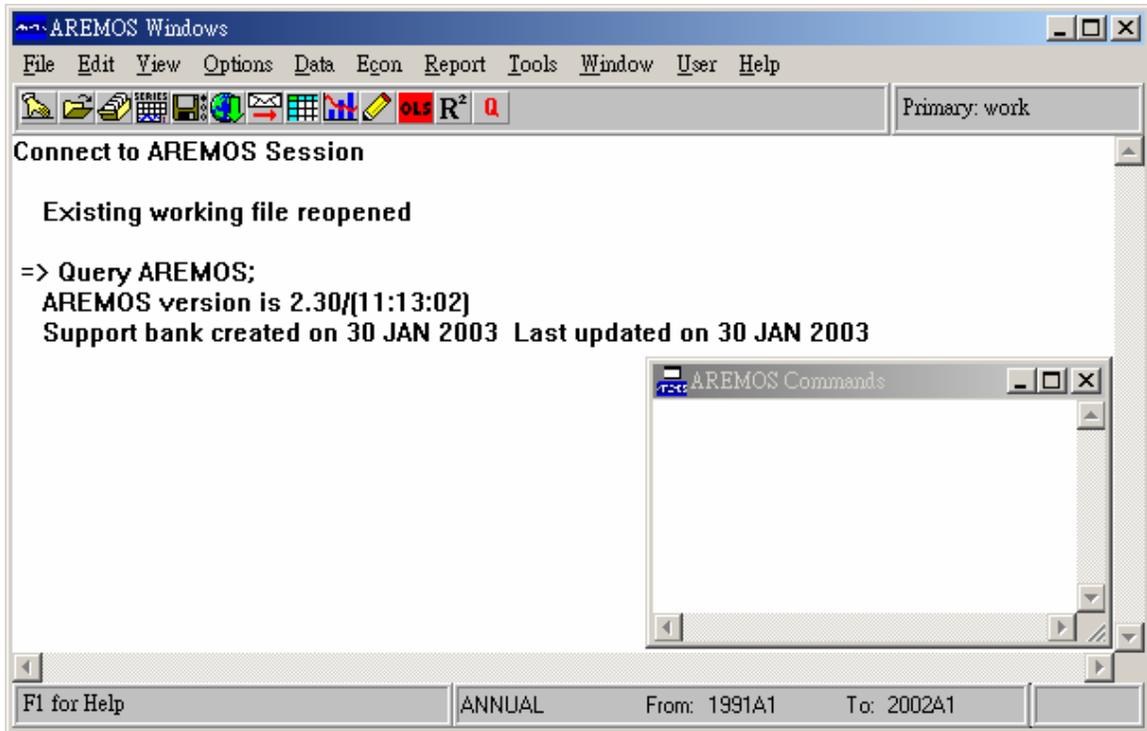


图 2.2

操作上几乎都用「鼠标」来叫取(除了某些 period, 如 1989 年到 1993 年或一些 AREMOS 的 Function 需要 key-in 指令外), 非常简便。一般熟悉 Windows 的人皆知鼠标操作按左键「一下」即是叫取, 按「两下」是执行的意思。而在图面上的右端或下端有( 图 ), 这是屏幕画面的滚动轴, 可看上一行, 上一页, 下一行及下一页, 或是左有两边的画面。

要停止使用或离开 AREMOS/Windows, 和一般离开 Windows 环境一样, 只要把鼠标移到最右上角( 图 )的地方, 按两次就是跳离的意思。

## 2. AREMOS/Windows 几个常用的工具

开始谈下一个范例前，先介绍几个 AREMOS/Windows 环境下常用的工具，可方便读者操作。当然，首先您必须知道如何打开「工具列」：先把鼠标移至 View 处按一下左键，找到 Toolbar 按一下，打「~」。若您不想用鼠标操作检索数据，而想 key-in 指令快速叫取数据，可按 Esc 键，即出现 AREMOS Command 的画面(在右下角)，或先把鼠标移至 View 处按一下，找到 Command Window 按一下，打「~」，即可出现 AREMOS Command 的画面，让您 key-in 指令。

若您已打开工具列，只要把鼠标移至各个图表工具停着不按键，就明了此工具代表的意思(在左下角有英文简介)。若您鼠标移至某个图表工具按一下鼠标左键，就直接进入(执行)此工具的用途。



: **Open banks** 打开数据库的意思



: **Index DataBank** 查询数据里的项目 (items)。



: **Series Editor** 可编辑或显示数据库 items 里的 data，且可以 update 资料，也可绘图及输出数据。



: **Set Savefile Options** 对储存的档案设定一些 option。



: **Receive** 是接收 AREMOS 格式项目的数据。



: **Send Series Data** 可输出数据库 items 里的 data 至 Clipboard(暂存区)或送至文字文件(file)下或 Excel 软件下，等等。



: **Tables** 是制表的工具，其格式和 Microsoft 里的 Excel (电子表格)类似。



: **Graph** 是绘图的工具，记得操作方式必须要去寻找 **Format**

里的 **Data Area**，找到所需的 **data Properties** 告诉计算机您要的 **Series Name**，才可画图。

 : **AREMOS Editor** 这是 **AREMOS** 的文书编辑系统，可写程序(**procedure**)，或作 **obey** 档。(有关 **Procedure** 与 **Obey** 档请参读第八章)。

 : **Ordinary Least Squares** 作计量经济统计回归分析的工具；这是采用普通最小平方法(**OLS**)去跑回归方程式。

 : **Residual Display** 根据以上 **OLS** 的回归分析后所作的残差值显示，有实际值及预测值，同时也可绘出残差值的图形。

 : **QuickData** 是连上至国外 **Global Insight** 公司在线叫取实时订购的国外数据库。

## 2.3 范例(一)：基本检索数据方法

在 **AREMOS** 环境下，所谓「检索数据」，即是要「开启」(**open**)某些数据库，而对数据库里的某些数据来做「印出」(**print**)，或「分析」(**analyze**)，或「说明」(**write**)等操作。而操作过程，必须「设定」(**set**)所要操作的数据「频率」(**frequency**)与「期间」(**period**)，以及所要显示「小数点以下位数」(**decimal places**)等。

假定我们要叫取台湾地区「国内生产毛额」、「民间消费」及「(以1996年价格表示之)实质国内生产毛额」等三笔资料。进行步骤如下：

(一)首先从「**AREMOS** 台湾地区国民所得统计数据库检索手册」一书中查知这三笔数据检索代号(**retrieval code**)分别为 **gdp**，**cp** 与 **gdp96**。同时也查知这三笔数据有两种频率：年 (**a**) 与季 (**q**)。年数据起始期间为 1951 年，季资料起始期间为 1961 年第 1 季。

(二)我们决定要叫取季数据,期间(period)是 1982 年第 1 季到 2002 年第 4 季(1982Q1 2002Q4)。

(三)我们也已知国民所得季数据库计算机文件名是: **niaq.bnk**。有了以上三步骤的准备,便可上机检索资料。开机后,操作鼠标按键如下:

- (1) **AREMOS Windows**
- (2) **AREMOS Windows**
- (3) **Options → Set → Period (Frequency : Quarterly, From 1982Q1 to 2002Q4 → OK**
- (4) **Options → Set → Report (Decimal Places : 0) → OK**
- (5) **Data → Bank → Open → Files → 磁盘驱动器 (看看数据库放在哪一槽) → niaq.bnk → 确定 → Open → cancel 跳回主画面**
- (6) **Report → Print → 键入 CP,GDP,GDP96 (在 Expression 处) → Execute**
- (7) **Option → Set → Report (Decimal Places:2) → OK**
- (8) **Report → Print → 键入 pchya(gdp),pchya(cp),pchya(gdp96) (在 Expression 处) → Execute**
- (9) **Data → Series → Analyze → 键入 gdp96,pchya(gdp96) (在 Expression 处) → Execute**
- (10) **Report → Write → Index → NIAQ (在 Bank 里) → Q**

(在 Version 选 Q) → Find → Ctrl-鼠标左键, 选  
CP,GDP,GDP96 → OK → OK

(11) Data → Bank → Index → Index → NIAQ (在 Bank 里)  
→ Q (在 Version 选 Q) → Find → Ctrl-鼠标左键, 选  
CP,GDP,GDP96 → OK → OK

(12) 主画面的右上、下角及下面, 有滚动轴, 只要把鼠标移至  
( 图 )处, 按多次, 即可看上一页, 下一页, 或左右两边  
的文字。

(13) Data → Bank → Close → NIAQ → OK

(14) File → disconnect → 是(Y)

(15) 用鼠标在右上角( 图 )处按一次跳出 Windows

说明:

- (1)(2) : 此指令为典型的 PC-Windows 执行方法, 先进入 AREMOS Windows 项目, 然后再选择 AREMOS Windows 项, 而进入 AREMOS Windows 的环境, 其画面如图 2.2。若您的右下角没有出现 AREMOS Command 画面, 只需按 Esc 键即出现此画面。若您不想用鼠标叫取数据, 直接在此 AREMOS Command 的画面里 key-in 指令后按 Enter 键, 其结果是完全相似于您用鼠标所检索的东西。(而您若想在 AREMOS Command 的画面下使用 key-in 指令, 请参考附录(二)和 AREMOS/UNIX 入门讲义第二章里的所有指令及说明)。
- (3) : 用鼠标选 Options 的地方, 再选 Set, 后选 Period, 在此可设定所要检索数据的时间频率 (frequency) 以及期间 (period)。在此设定季频率 Quarterly。期间如果是年数据,

需改成 **Annual**。如果是季数据，则需指定那一季到那一季，例如 **1982Q1 2002Q4** (您必须 **key-in**)代表 1982 年第 1 季到 2002 年第 4 季。按 **OK** 结束跳出，如图 2.1.3。

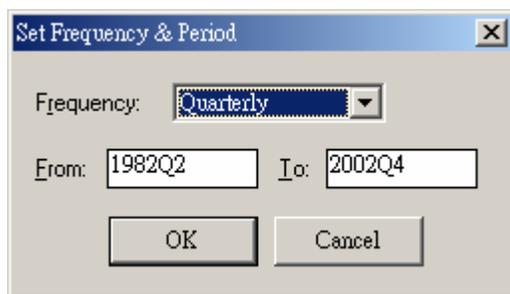


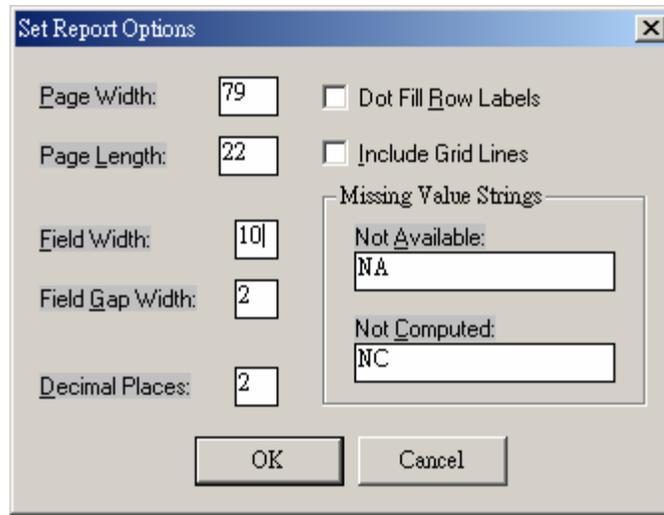
图 2.1.3

(4)：我们选 **Options** 里的 **Set** → **Report**，在 **Decimal Places** 处，改此 **global** 里的小数点以后几位数 (如图 2.1.4)。往后您若在 **AREMOS Command** 画面下 (**key-in**) 键入 **print**，**analyze** 等所显示出 **data** 小数点会被影响。

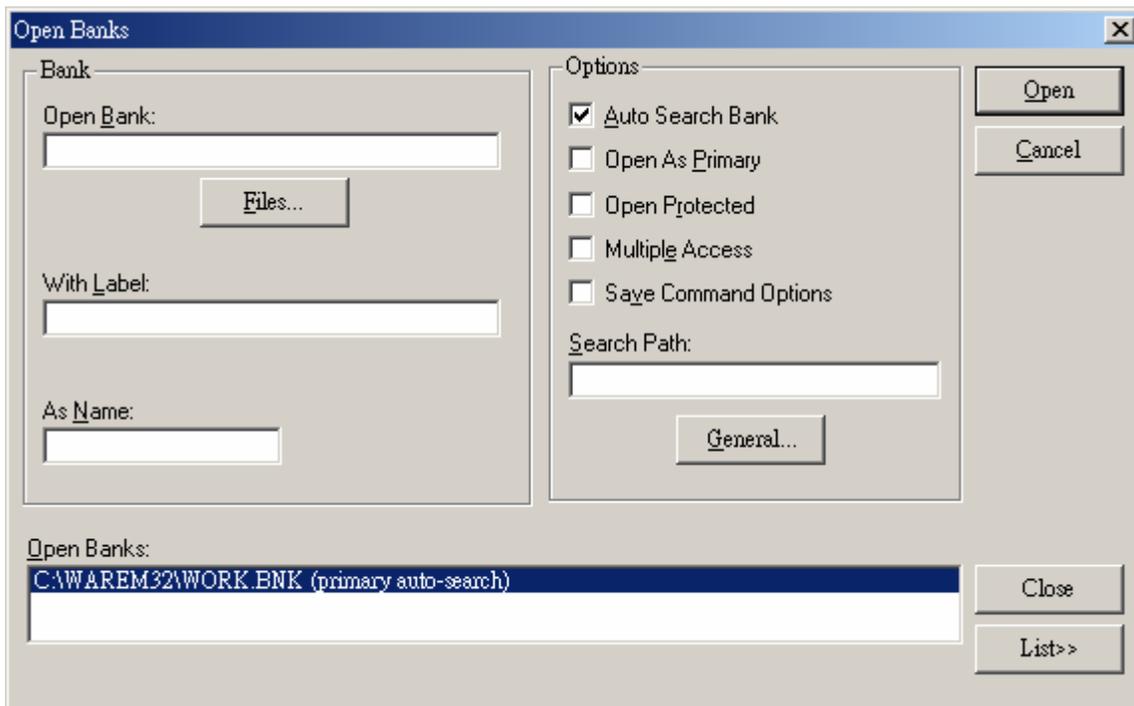
(5)：此步骤是在开启 (**open**) **databank** (数据库)。当鼠标按至 **Files** 处，屏幕会出现 **Open Bank File** 的画面，您先去磁盘驱动器处找寻您数据库所放的槽，然后选择您所要叫取的数据库文件名 **niaq.bnk**，按确定后，就如图 2.1.5。一旦开启某个数据库即可叫取该数据库里的数据。同一时间可开启多个数据库。**niaq** 是 **National Income Accounts Quarterly** 的简称。

(6)：在 **Report** 里的 **Print** 是可把资料在荧光幕上印出来。在 **Print Expressions** 处，您可使用 **Index** 去检索数据，也可自己在此

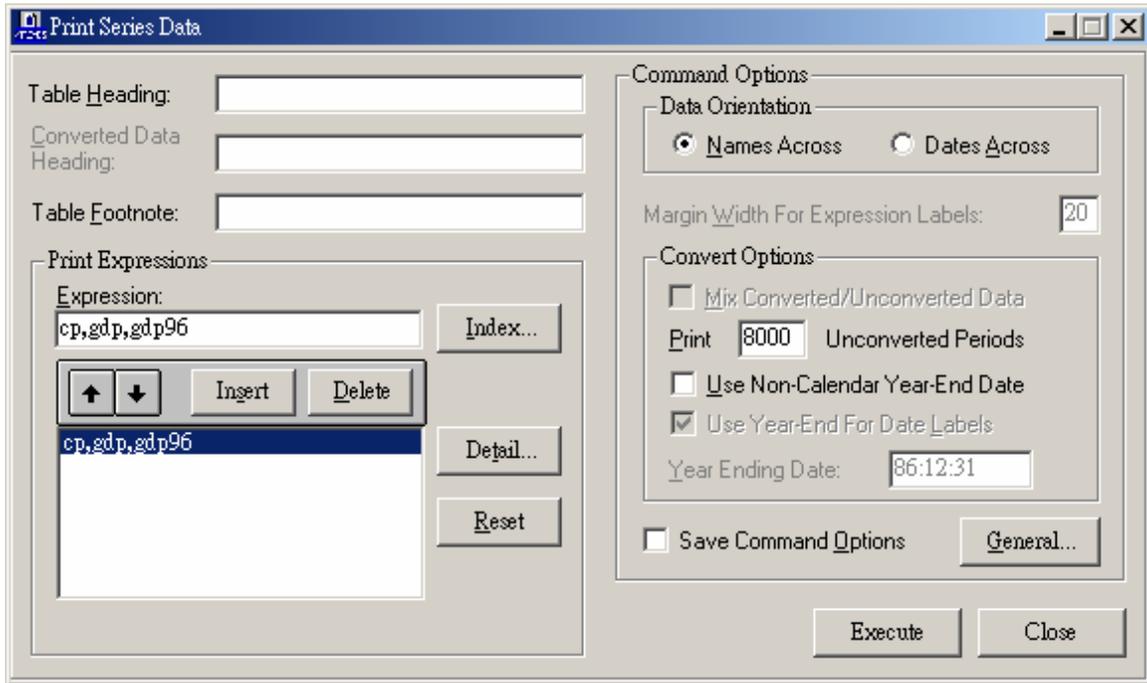
key-in 键入 items 的名字, 然后用鼠标按 **Execute** 执行 (如图 2.1.6)。但如果您之前忘了 **open bank**, 您可在 **Print Series Data** 画面下, 找到 **General** 键钮, 在此按鼠标一下, 即可进入 **General Options** 画面, 然后按 **Open** 去打开您所要的 banks(如图 2.1.7)。



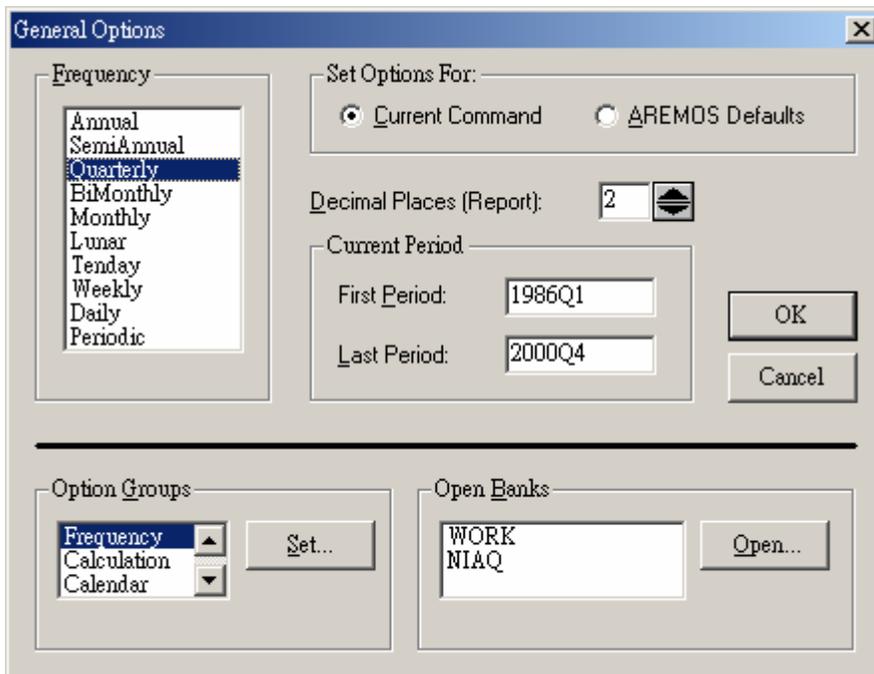
(图 2.1.4)



(图 2.1.5)



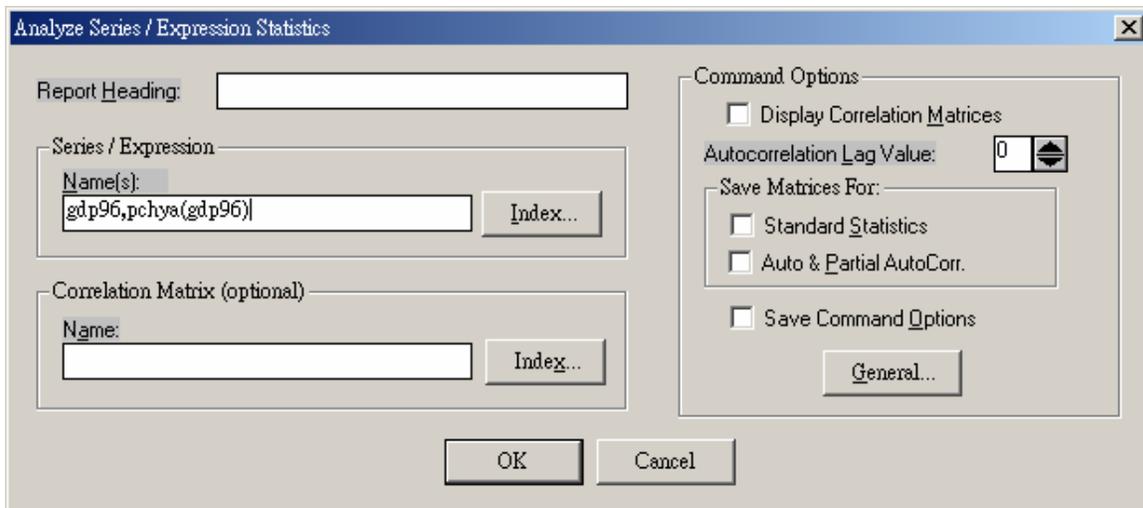
(图 2.1.6)



(图 2.1.7)

(7)(8) : 此指令可印出这三笔数据「与去年同期比之成长率」。Pchya 是一个 AREMOS function, 代表「Percentage CHange From a Year Ago」。在此 pchya(gdp96)即是一般所谓「经济成长率」。而印出成长率, 一般习惯印出小数点后两位, 故 (7) 先设定 report decimal 为 2。事实上「设定」option 也可以用 local 来设, 故 (8) 在其内部设定上, 已是个 local 设定了。有关 local 与 global 设定 option 请参考第六章第 6.2 节。

(9) : analyze 可印出资料的 mean, standard deviation, maximum 及 minimum 等基本统计指标。操作过程 (如图 2.1.9)。但如果之前您忘了 open bank, 您可在此 Analyze Series/Expression Statistics 画面下, 找到 General 键钮, 在此按鼠标一下, 即可进入 General Options 画面, 然后按 open 您所要的 banks(如图 2.1.7 一样)



(图 2.1.9)

- (10) : 再 AREMOS 数据库里每一笔数据均有三至五行的「说明」(document)以供查询。这些说明通常包括该数据的全名、单位、来源等等。Write 即可印出这些说明。
- (11) : 我们国内自己建立的数据库，每笔数据一律建有「说明」(document)，可用 write 指令来印出 document。而您也可用 index 指令来印出该数据的全名与单位，并显出该笔数据在数据库中 available 的总期间。大致说来，write 与 index 所显示的信息不太一样，读者可习惯选用较适合自己所需要的。
- (12) : AREMOS/Windows 会把荧光幕上的右边和下面有滚动轴可看上一个 page，下一个 page，及左右两边的文字。
- (13) : 此操作可把开启的数据库「关闭」起来。没用的数据库最好 close 起来。同时开启太多数据库会影响运算速度。
- (14) : disconnect 是离开 AREMOS/Windows 环境，跳回 Windows 原本的画面。
- (15) : 完全跳出 Windows。

\*补充范例(一)

您在设定 frequency(频率)及 period 或 open bank 时，使用工具列上的图表工具或是把鼠标移至最右下角去，按左键一下(在 frequency 和 period 处)，就可直接进入 set Frequency & Period 画面去修改 frequency 和 period。而把鼠标移至右上角 Primary Bank:work 处按”一下”，亦可进入 Open Banks 的画面，可去 open 您所要的 banks。

至于往后的范例(二)(三), 若您熟悉工具列的图表工具, 可多利用图表工具去检索数据, 非常方便。

## 2.4 范例(二): 如何 **Export** 或 **Send** 数据以便用于其它软件

虽然 **AREMOS/Windows** 有相当完整的运算分析功能, 有不少读者仍习惯于把数据放入 **PC**, 并使用自己熟悉的软件来做运算、绘图等。本范例介绍如何把 **AREMOS/Windows** 数据库里的数据以

「**Export**」方法及「**Send**」方法转入其它软件使用;前者是转入 **PC-DOS** 下, 而后者是送到暂存区(**Clipboard**), 以便其它软件使用。若您熟悉 **Microsoft** 里的 **Excel**, 把数据放在暂存区后(**Clipboard**), 只要「贴上」即可。此外我们亦设计一 **procedure**, 取名为 **df(Data-file)**之简称, 可执行与 **export<prn>**同样功能, 但格式略有不同。读者可自行选用。

假定我们要叫取台湾地区「工业生产总指数」、「总出口」与「总人口」等三项统计之月资料, 期间从 **1991** 年 **1** 月到 **1992** 年 **12** 月, 同时要把这些数据 **export** 出来形成一个 **file** 以便进一步使用。则如同范例(一)之进行步骤, 我们首先由「**AREMOS** 台湾地区工业生产统计数据库检索手册」、「**AREMOS** 台湾地区进出口贸易统计数据库检索手册」、「**AREMOS** 台湾地区人力资源统计数据库检索手册」等三本书中查知这三笔数据检索代号及数据库计算机文件名如下:

	<u>检索代号</u>	<u>数据库文件</u>
名		
「工业生产总指数」	<b>jqind</b>	<b>ind.bnk</b>
「总出口」	<b>ex</b>	<b>trade.bnk</b>

「总人口」

n

man.bnk

此外，我们也要范例(一)里的 gdp、cp、gdp96 等三笔季资料，期间 1982 年第 1 季到 2002 年第 4 季，直接 send 到 Windows Clipboard(暂存区)。我们使用鼠标操作方法如下：

(1)AREMOS Windows

(2)AREMOS Windows

(3)Options → Set → Period (Frequency : Monthly, From 1991M1 to 1992M12) → OK

(4)Options → Set → Report (Decimal Places: 2) → OK

(5)Data → Bank → Open → Files → 磁盘驱动器 (看看数据库放

在哪一槽) → ind.bnk  
trade.bnk → 确定 → Open →  
分别叫取 man.bnk

Cancel 跳回主画面

(6)Report → Print → 键入 jqind,ex,n(在 Expression 处) →  
Execute

(7)Data → List → 键入 n1(在 Name 处) → 键入 jqind,ex,n(在  
List Item Text 处) → OK

(8)Data → Transfer → Export → 键入 #n1(在 Export Series  
Name 处) → 键入 aaa(在 Export file Name 处) → 选 Names  
Across → OK

(9)键入指令 view aaa.prn;

- (10) **Data** → **Transfer** → **Export** → 键入 #n1(在 **Export Series Name** 处) → 键入 bbb(在 **Export file Name** 处) → 选 **Names Across** → 选  **Exclude Dates** → **OK**
- (11) 键入指令 **view bbb.prn**;
- (12) **Data** → **Transfer** → **Export** → 键入 #n1(在 **Export Series Name** 处) → 键入 ccc(在 **Export file Name** 处) → 选 **Names Across** → 选  **Exclude Dates** 及  **Exclude Names** → **OK**
- (13) 键入指令 **view ccc.prn**;
- (14) **Tools** → **AREMOS Editor** → **File** → **Open** → 文件名称 → **All Files** → **aaa.prn** → 确定
- (15) **Options** → **Set** → **Period (Frequency : Quarterly, From 1982:1 to 1992:4)** → **OK**
- (16) **Options** → **Set** → **Report(Decimal Places: 0)** → **OK**
- (17) **Data** → **Band** → **Open** → **Files** → 磁盘驱动器(看看数据库放在哪一槽) → **niaq.bnk** → 确定 → **Open** → **Cancel** 跳回主画面
- (18) **Tools** → **Send Series Data** → 键入 **gdp,cp,gdp96**(在 **Name** 处) → **Retrieve** → 选 **Clipboard** → **send** → 确定 → **Close**
- (19) **Data** → **Series** → **Edit** → 键入 **gdp,cp,gdp96**(在 **Enter Series Name** 处) → **OK** → **Mark All** → 按绘图工具 → **Close**
- (20) **Data** → **Bank** → **Close** → **Ctrl-鼠标左键**, 选

**IND,MAN,NIAQ,TRADE → Close → Cancel**

**(21) File → disconnect → 是(Y)**

说明：

**(1)(2)(3)(4)(5)(6)：**见范例(一)之说明。在此设定频率为 M(月)。其它常用的频率有 A(年)，W(周)，D(日)。

**(7)：**此指令可把 **jqind,ex,n** 3 笔数据名称形成一个 list, 名叫 **n1**。接着可用 **#n1** 代表者 3 笔资料(如图 2.2.7)。有关 list 更进一步介绍，请参考本讲义第七章。

**(8)：**Export 传送出去在 C 碟软件子目录 **WAREM32** 下形成一个 file。 **#n1** 代表所要 Export 的三笔资料， **aaa** 则为随意取的 filename(如图 2.2.8)。执行完此动作，您的 **WAREM32** 子目录即有一个 file, filename 是 **aaa.prn**。计算机自动加上 **.prn** 为其 filename 的 extension。在 **Command Options** 里的 **File Format**，共有五种格式可选择：

**PRN: Tabular Format**

**TSD: Time Series Data Format**

**DIF: Data Information Format**

**XOF: XSIM-XOFF Format**

**XNC: 也是 XSIM-XOFF Format**

有关进一步说明，请参考原文手册「**Reference Manual**」**P.85 ~86**。或将光标停放在 **AREMOS/Winows** 接口上某一画面，按键盘上的 **F1** 功能键，即刻显现原文的说明。

**(9)：**键入指令 **view** 可看到 C 槽下 **WAREM32** 子目录下的文字文件。

(10)(11): 在 **Export** 传送的过程中, 选  **Exclude Dates**, 所形成资料文件会少掉代表期间(date)的那一行。

(12)(13): 在 **Export** 传送的过程中, 选  **Exclude Dates** 后, 再加选  **Exclude names**, 所表示的是, 形成的资料文件不再有 **jqind,ex,n** 等数据名称。以上为三种 **Export** 不同的设定, 读者可自己选用最适当的方式。

(14): 在 **AREMOS/Windows** 环境下编辑, 先找(8)所做出的 **aaa.prn** 看看, 在此可修改编辑。**AREMOS/Windows** 是用 **Tools** 里的 **AREMOS Editor** 来编辑文字、数据、程序。在第八章的 **Procedure** 与 **Obey** 的介绍, 我们就是用 **AREMOS Editor** 来编辑的。

(15)(16)(17): 见范例(一)说明。

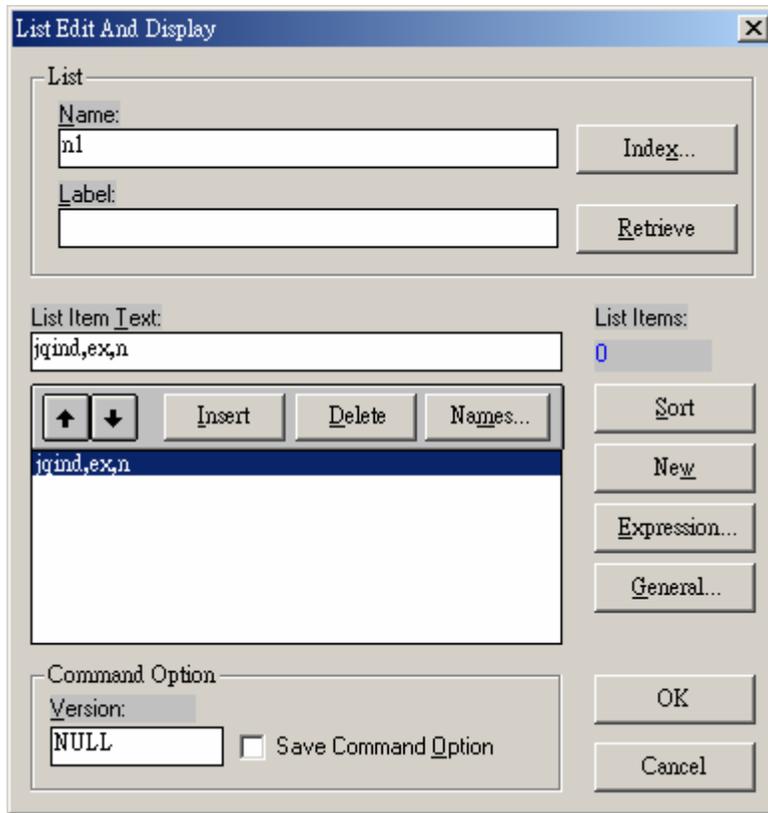


图 2.2.7

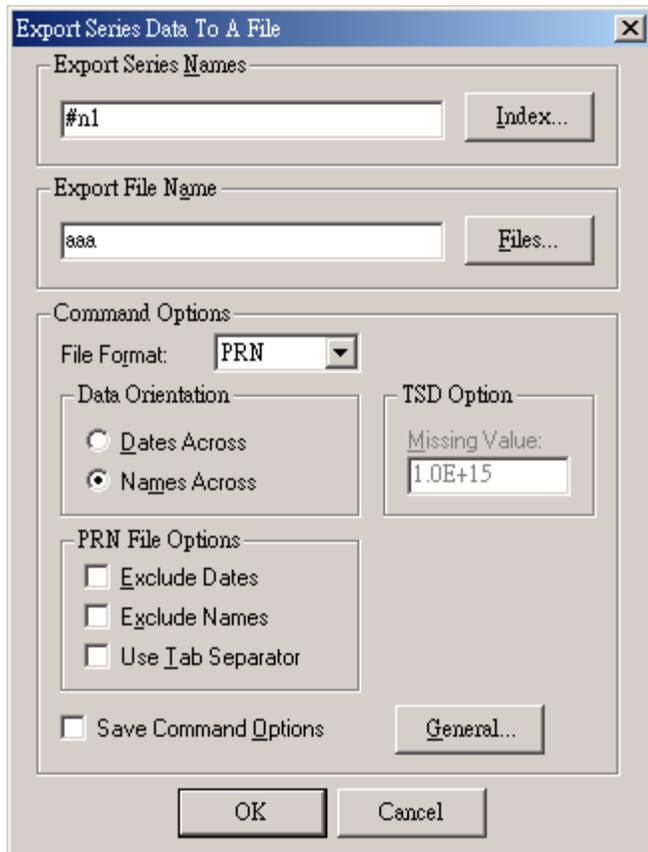


图 2.2.8

(18): AREMOS/Windows 允许我们将数据直接送到 Windows Clipboard, 然后使用其它在 Windows 里的软件来操作运算等。此三笔 gdp、cp、gdp96 键入后(在 Send Series Data 画面下), 按 Send, 这三笔数据会送到 Windows Clipboard 去(如图 2.2.18)。若您是在主画面下(注: 必需是 AREMOS 5.301 以下的版本才有此功能), 键入指令 `Sendclip gdp,cp,gdp96;`, 不但三笔数据会送到 Windows Clipboard 去, 同时在 C 槽 WAREM32 子目录下形成一个档叫 `sendclip.prn`。此时, 若我们离开 AREMOS/Windows 而回到 Windows 的环境, 即可在 Clipboard Viewer 看到 `gdp,cp,gdp96` 这三笔资料。只要打开

Excel 软件，按「贴上」的图表工具，数据马上显现。当然我们也可加以储存以用于其它 Windows 的软件。

(19): 此方法和(18)类似，但他多了 edit 编辑及 graphic 绘图的功能。先键入变量名字后，按 OK 即进入 Series Edit and Display，您要 update 新的 series items 皆可行。若要绘图先要按 Mark All，再把鼠标移至绘图工具的钮(在右下角)。如图 2.2.19。

(20)(21): 见范例(一)说明。

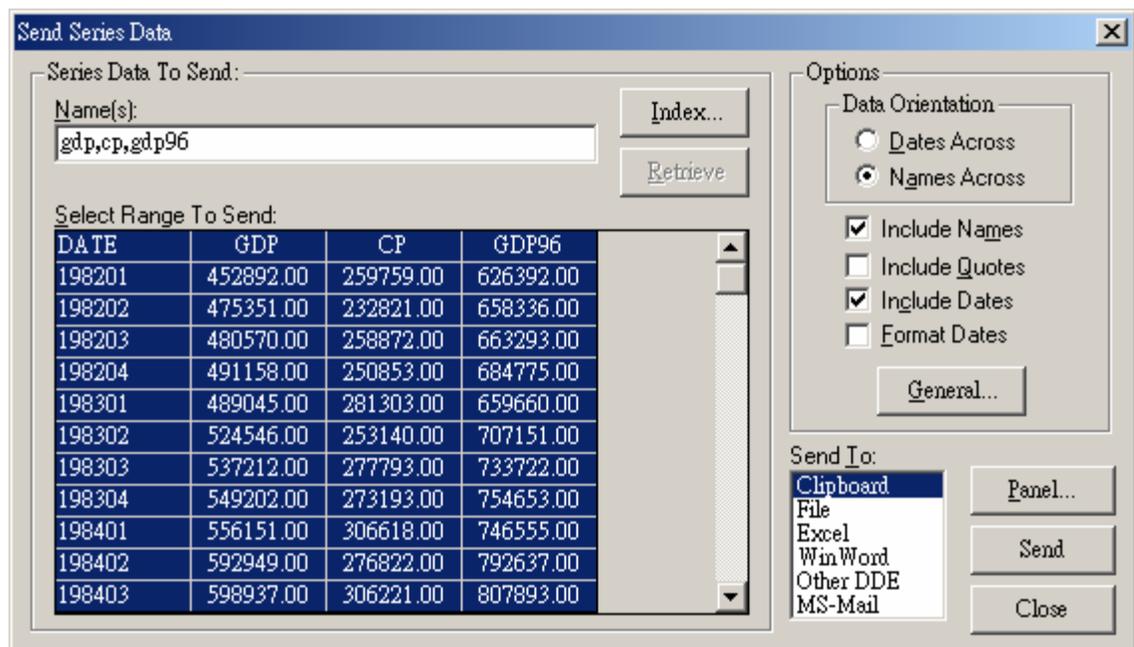


图 2.2.18

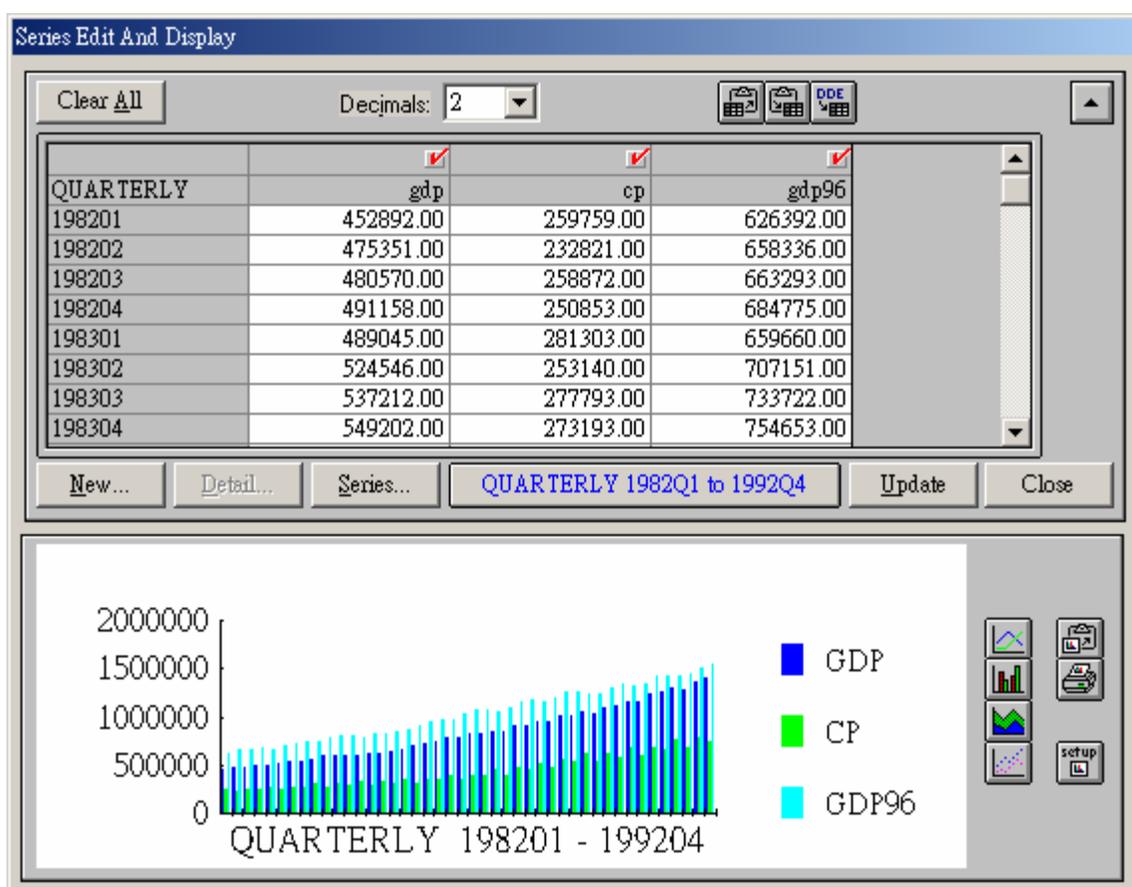
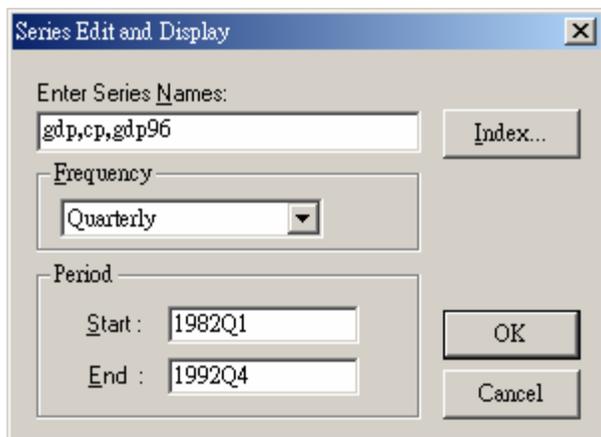


图 2.2.19

## 2.5 范例(三): 如何检索数据转换频率并做回归分析

假定我们要对台湾地区狭义货币供给额 **m1a** 进行回归分析。分析期间是季资料, 从 1993 年第 1 季到 2003 年第 4 季。解释变量是台湾地区「国内生产毛额」及「一月期定期存款固定利率」。

首先, 从「AREMOS 台湾地区金融统计数据库检索手册」查知 **m1a** 与「一月期定期存款固定利率(第一银行)」之检索代号分别为 **m1a** 与 **rmtd1@1st@f**, 数据库计算机文件名为 **fsm**。此二笔数据均为月数据。另外已知「国内生产毛额」检索代号为 **gdp**, 是季数据, 数据库计算机文件名为 **niaq**。用鼠标操作如下:

(1)AREMOS Windows

(2)AREMOS Windows

(3)Options → Set → Period(Frequency : Quarterly, From  
1993:1 to 2003:4) → OK

(4)Data → Bank → Open → Files → 磁盘驱动器 (看看数据库放在那一槽) → **niaq.bnk**  
分别叫取 **fsm.bnk** → 确定 → Open  
→ Cancel 跳回主画面

(5)Data → Series → Collapse → 键入 **m1a.m** (在 Collapse Existing Series 处) → 键入 **m1a.q** (在 To Lower Frequency Series 处) → 选 Average (在 Use Conversion Method: 处) → OK

(6)Data → Series → Collapse → 键入 **rmtd1@1st@f.m** (在 Collapse Existing Series 处) → 键入 **rmtd1@1st@f.q** (在 To Lower Frequency Series 处) → 选 Average (在 Use Conversion Method: 处) → OK

- (7) **Econ** → **Equation** → **OLS** → 先点 **New** → 键入 **equ1**(在 **Name** 处) → 键入 **m1a.q**(在 **Dependent** 处) → 键入 **gdp** 及 **rmtd1@1st@f**(在 **Independent** 处的格子里) → **Execute** → 按 **R<sup>2</sup>** 图表工具 (在主画面下) → **Close**
- (8) **Econ** → **Equation** → **OLS** → 键入 **equ2**(在 **Name** 处) → 键入 **m1a.q**(在 **Dependent** 处) → 键入 **gdp** 及 **rmtd1@1st@f**(在 **Independent** 处的格子里) → 键入 **1**(在 **AR Elements** 处) → **Execute** → 按 **R<sup>2</sup>** 图表工具 (在主画面下) → **Close**
- (9) **Econ** → **Equation** → **OLS** → 键入 **equ3**(在 **Name** 处) → 键入 **m1a.q**(在 **Dependent** 处) → 键入 **gdp** 及 **m1a[-1]**及 **rmtd@1st@f**(在 **Independent** 处的格子里) → **Execute** → 按 **R<sup>2</sup>** 图表工具(在主画面下) → **Close**
- (10) **Econ** → **Equation** → **OLS** → 键入 **equ4**(在 **Name** 处) → 键入 **m1a.q**(在 **Dependent** 处) → 键入 **log(gdp)**及 **log(rmtd1@1st@f)**(在 **Independent** 处的格子里) → **Execute** → 按 **R<sup>2</sup>** 图表工具(在主画面下) → **Close**
- (11) **File** → **disconnect** → 是 (Y)

说明：

- (1)(2)(3)(4)：见范例(一)之说明。
- (5)：此 **collapse** 窗口是转换数据(从高频率转成低频率)的意思；而相对地是 **Interpolate** 窗口也是转换数据(从低频率转成高频率)的意思，可点取 **Data** → **Series** → **Interpolate** 查看。当然除了以上点取窗口的方法外，也可以用「键入指令」的方法如 **collapse** 或 **Interpolate** (在 **AREMOS Command** 窗

口下), 而 **collapse** (用 **conv** 也行, 如此是 **convert** 之简称) 指令可让我们把数据库里之「月」数据(在此 **m1a.m** 之 **.m** 代表属原始数据「月频率 **version**」之 **m1a**。有关 **version** 之进一步说明请参读第三章 3.3 节及第六章 6.6 节。) 转换成目前所设定之「季」频率数据。选 **average** 乃指定转换方法为取三个月之平均为季数据。货币数量与利率通常取平均值 **ave(average)**来做转换。其它常用的转换方法尚有:

- total** : 取三个月之加总
- first** : 取三个月之第一个月
- last** : 取三个月之最后一个月
- minimum** : 取三个月中之最小值
- maximum** : 取三个月中最大值

注: 若您是要从「月数据」转换成「季数据」, 首先在时间频率的设定上是必须 **set** 「季」 **frequency** 频率才对(**Global Setting**), 以下的转换或分析数据皆以「季」为主。

(6): 见(5)之说明。执行完此指令, 我们已有 **m1a** 与 **rmtd1@1st@f** 两笔转换成的季数据。另外 **gdp** 在数据库中已是季数据, 不需转换。

(7): 这是在做回归分析, 采 **OLS**(最小普通平方法), 进入 **OLS Equations** 画面后, 先设定名字 **equ1**, 再设定 **Dependent**(被解释变量)的名字, 及 **Independent**(解释变量)的名字, 用鼠标按 **Execute** 键。回主画面后, 按 **R<sup>2</sup>** 图表工具可看出此 **Equation** 的 **Residual** 情况及图表分配(如图 2.3.7)。

(8): 和(7)类似, 不过加上键入 **1**(在 **AR Terms** 处), 如图 2.3.8。

这是 **1st-order autocorrelation correction**。在此乃采用 **Cochrane-Orcutt** 方法。

**(9)**: 此回归取被解释变量之递延一期(**lagged dependent variable**)为解释变量。**AREMOS** 以 [ ] 代表递延(**lag**)。 **m1a[-1]** 是 **m1a** 之递延一期(注: **m1a.1** 计算机也接受是 **m1a** 之递延一期); **m1a[-2]** 是 **m1a** 之递延两期。 **m1a[+1]** 则是 **m1a** 之领先(**lead**)一期。

**(10)**: 对每个变量取自然对数(**log**)后再进行回归。

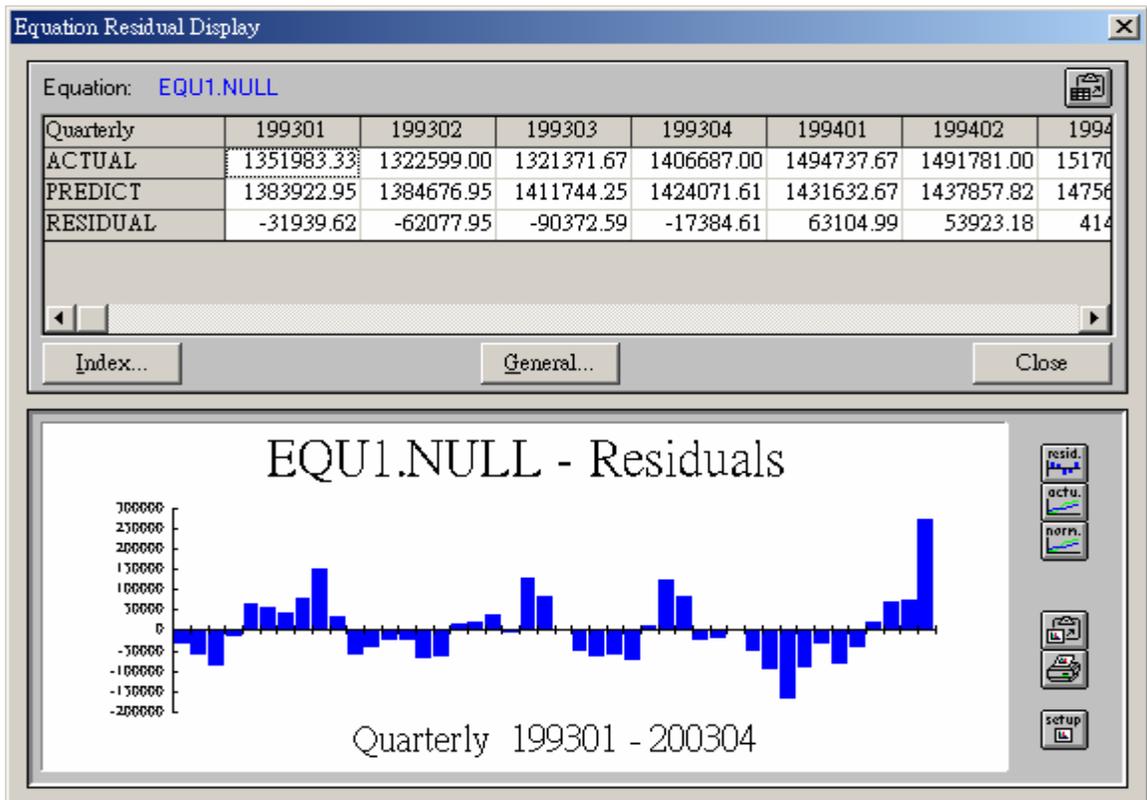
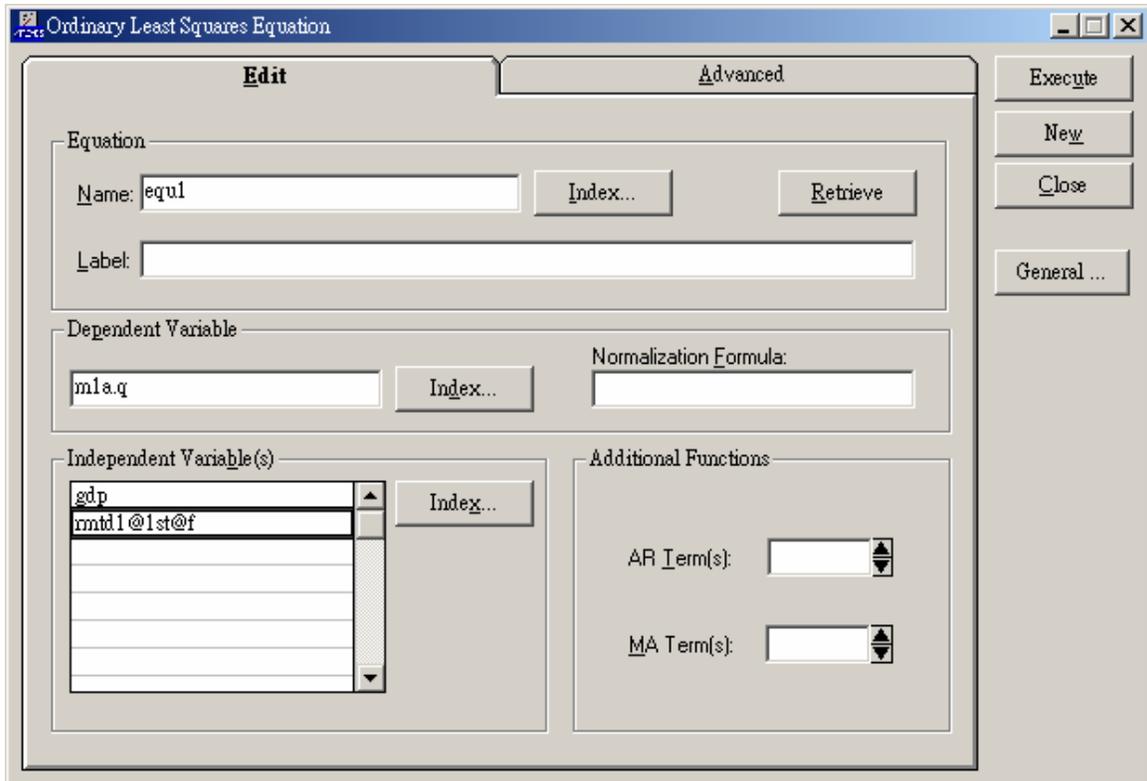


图 2.3.7

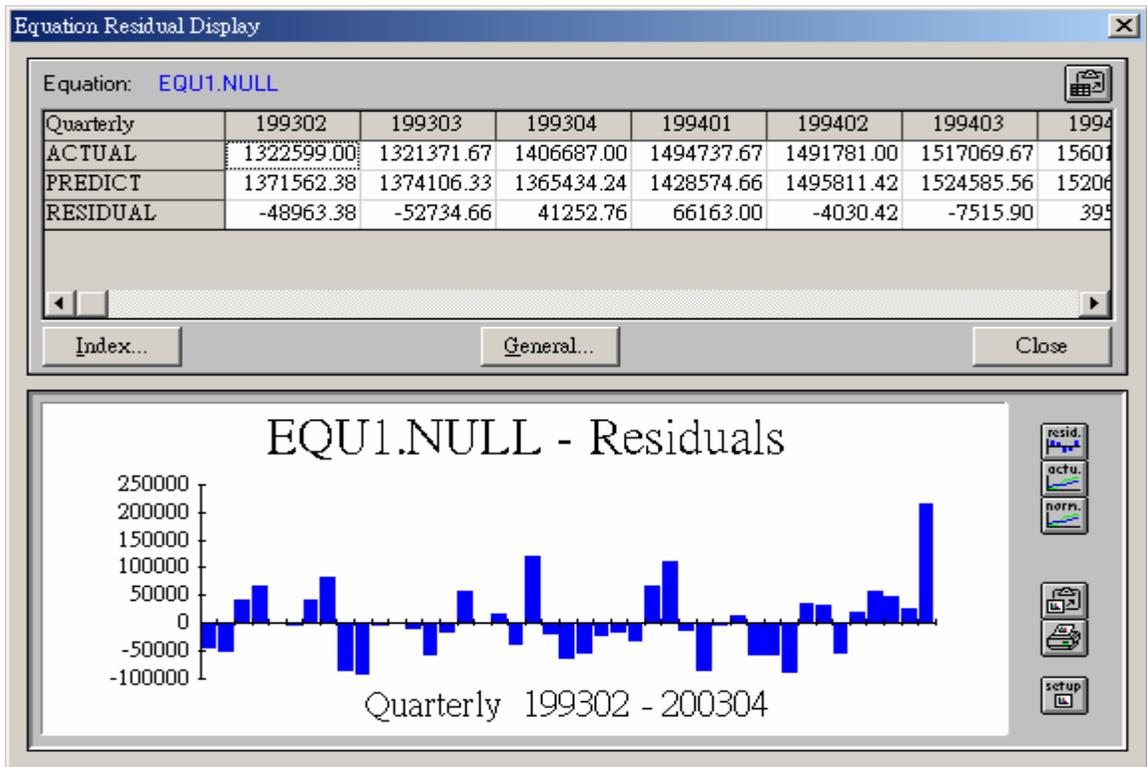
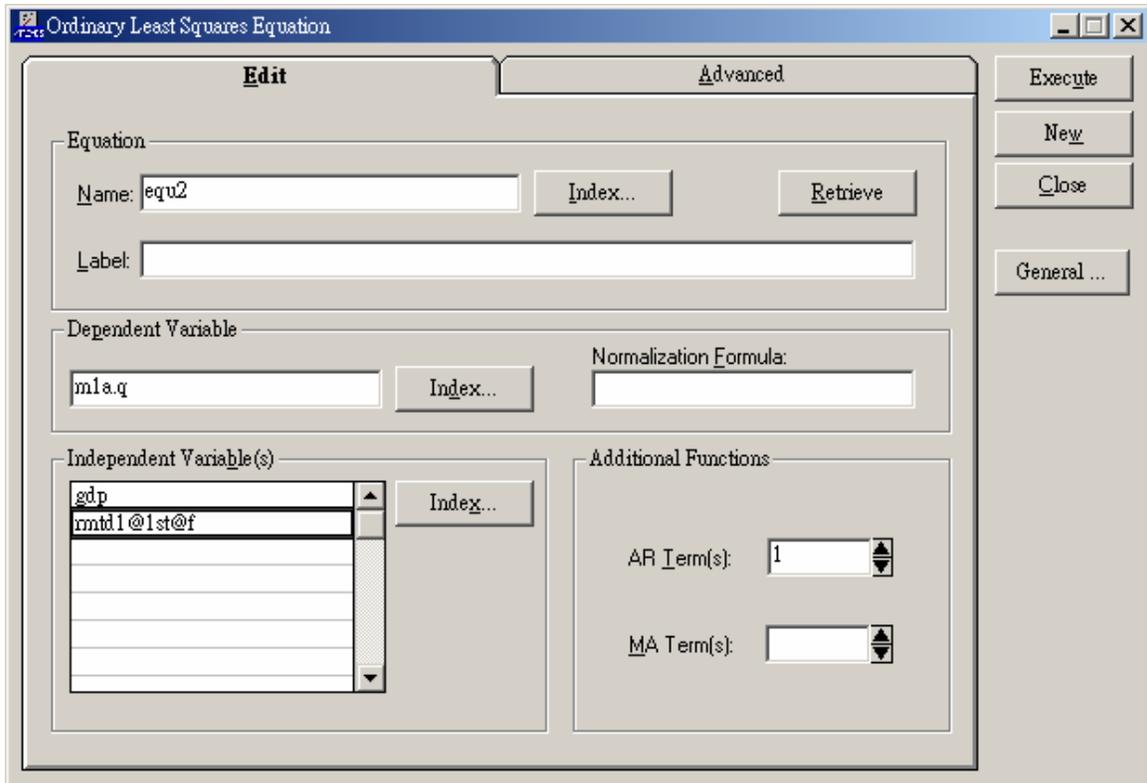


图 2.3.8

## 第三章 有关检索代号 (Retrieval Code)

### 3.1 数据库检索手册

从第二章所介绍检索数据方法可见查出「检索代号」是最重要步骤。一旦知道所需数据之检索代号及其所属数据库计算机文件名，则操作指令并不难。为了方便查考，我们每一数据库都有一本「检索手册」把该数据库检索代号列举出来。截至目前，已出版检索手册及数据库计算机文件名如下：

#### (一) 国内「一般性」统计数据库(共 14 个)

##### (1) 「台湾地区国民经济动向统计季报数据库检索手册」

QNET: 年与月资料 391 笔

##### (2) 「台湾地区国民所得统计数据库检索手册」

NIAA: 年资料 900 笔

NIAQ: 季资料 130 笔

##### (3) 「台湾地区工业生产统计数据库检索手册」

IND: 年与月资料 4,028 笔

##### (4) 「台湾地区进出口贸易统计数据库检索手册」

TRADE: 年与月资料 2,644 笔

##### (5) 「台湾地区人力资源统计数据库检索手册」

MAN: 年与月资料 999 笔

##### (6) 「台湾地区能源统计数据库检索手册」

OECDR, OECDC, OECDO, ENG: 年与月资料约 19,323 笔

##### (7) 「中华民国教育统计数据库检索手册」

EDU: 年资料 410 笔

(8) 『台湾地区薪资与生产力统计数据库检索手册』

WAGE: 年与月资料 518 笔

(9) 『中华民国财税统计数据库检索手册』

GOVT: 年资料 91 笔

TAX: 年与月资料 155 笔

(10) 『台湾地区物价统计数据库检索手册』

PRICE: 年与月资料 1,042 笔

(11) 『中华民国交通统计数据库检索手册』

TRANS: 年与月资料 408 笔

(12) 『台湾地区农业统计数据库检索手册』

AGR: 年与月资料 494 笔

(13) 『台湾总体经济预测数据库检索手册』

DGBAS: 年与季资料 110 笔

CEPD: 年资料 25 笔

IEAS: 年资料 28 笔

(14) 『台湾地区多因素生产里统计数据库检索手册』

MFP: 年资料 435 笔

(二) 国内「金融性」统计数据库(共 12 个)

(15) 『台湾地区金融统计数据库检索手册』

FSM: 年、季、月与日资料 4,023 笔

(16) 『台湾地区资金流量统计数据库检索手册』

AAL: 年与半年资料 1,445 笔

FOF: 年与半年资料 531 笔

(17) 『台湾地区国际收支平衡表统计数据库检索手册』

BOP: 年与季资料 526 笔

- (18) **「台湾地区 IMF IFS 格式金融统计数据库检索手册」**  
IFS: 年、季与月资料 321 笔
- (19) **「台湾股票市场统计数据库检索手册」**  
STOCK70, STOCK80, STOCK90, STOCK00: 日资料数  
万笔  
TSE: 年、月与日资料 1,870 笔
- (20) **「台湾上市公司股票报酬率统计数据库简介」**  
STOCKRR7, STOCKRR8, STOCKRR9: 日资料 668 笔  
STOCKRR: 年、季、月与周资料 2,672 笔
- (21) **「台湾股票上市公司财务报表数据库检索手册」**  
「一般产业股」部分 COMP:  
年资料, 有上百家公司, 每家公司 174 笔  
「金融保险股」部分 COMPF:  
年资料, 有数十家公司, 每家公司 191 笔  
「金控股」部分 COMPH:  
年资料, 有数家公司, 每家公司 124 笔
- (22) **「台湾柜台买卖统计数据库检索手册」**  
OTC: 日资料 2,612 笔
- (23) **「台湾上柜公司股票报酬率统计数据库检索手册」**  
OTCRR: 年、季、月、周与日资料 1,550 笔
- (24) **「台湾股票上柜公司财务报表数据库检索手册」**  
「一般产业股」部分 OTCCOMP:  
年资料, 有上百家公司, 每家公司 174 笔  
「金融保险股」部分 OTCCOMPF:  
年资料, 有数家公司, 每家公司 191 笔  
「金控股」部分 OTCCOMPH:

年资料，有数家公司，每家公司 124 笔

(25) 『台湾地区保险市场统计数据库检索手册』

INS: 年资料 1,254 笔

(26) 『台湾地区期货市场统计数据库检索手册』

FEX: 年、月、日资料 280 笔

(三) 大陆地区经济统计数据库(共 1 个)

(27) 『大陆经济统计数据库检索手册』

CHINA: 年、月资料 569 笔

(四) 台湾地区商业动态与景气指针统计数据库(共 2 个)

(28) 『台湾地区商业动态统计数据库检索手册』

COMM: 年、季、月资料 179 笔

(29) 『台湾地区景气指针统计数据库检索手册』

BCI: 年、季、月资料 262 笔

## 3.2 在线(ON LINE)查询检索代号

虽然各数据库检索手册已把检索代号列举出来，事实上，这些检索代号也可在上机后从计算机里印出来。本节介绍在线(on line)查询检索代号方法。

既然各手册已列举检索代号，为何需要在线查询？有两种情形可能会用得上：

第一、有时手头上刚好没有检索手册。如果记得住本节所介绍在线查询法，则可以派得上用场。

第二、这些检索代号随时可能会更动，而出版新的检索手册难免落后一段时间。

在计算机里则可查询出最新、最正确的检索代号。

在线查询有二种方法：(1)「INDEX 在线查询法」

(2)「Seriescode 在线查询法」

第(1)项是在 AREMOS/Windows 环境下操作，而第(2)项必须在数据库光盘片的子目录下检索，进入<seriescode>子目录，点取「seriescode.htm」超文件文件即可看到所有的数据库检索手册。当然您也可直接上财团法人经济资讯推广中心网站，在 3.5 有介绍。

### 3.3 「INDEX 在线查询法」

使用 INDEX 在线查询，我们可以用鼠标点取「Data→Bank→Index...」或进入「AREMOS Command」的窗口操作(在 AREMOS/Windows 主画面下直接 key-in 键入任何指令，即出现「AREMOS Command」的画面)。在介绍 index 操作方法前，首先我们必须了解 AREMOS/Windows 数据名称结构。一笔时间序列数据，如 gdp96 或 j22，是一个 object。在 AREMOS/Windows 里，每一个



(4)Data → Bank → Index → 键入指令 **gdp\***(在 Name(s) or Wildcard 处) → OK

(5)Data → Bank → Index → 键入指令 **g\***(在 Name(s) or Wildcard 处) → OK

(6)Data → Bank → Index → 键入指令 **g\*a\***(在 Name(s) or Wildcard 处) → OK

(7)Data → Bank → Index → 键入指令 **g??**(在 Name(s) or Wildcard 处) → OK

(8)Data → Bank → Index → 键入指令 **gn-h**(在 Name(s) or Wildcard 处) → OK

(9)Data → Bank → Index → 键入指令 **niaq:**(在 Name(s) or Wildcard 处) → OK

(10)Data → Bank → Index → 键入指令 **j22**(在 Name(s) or Wildcard 处) → OK

说明：

- (1): 开启 **niaq** 与 **ind** 两个数据库为查询对象。
- (2): 此操作是在寻找「**gdp**」为 **objectname** 之数据(如图 3.3.1)。结果只有一笔，即是 **niaq** 里之 **gdp**。
- (3): 此操作是限制在 **niaq** 数据库寻找名字为 **gdp** 之数据。与指令(2)不同处在于，指令(2)不限制在 **niaq** 里寻找，而是所有已 **open** 的数据库均可。但因只有 **niaq** 存 **gdp** 这笔数据，(2)与(3)的结果是相同的。

- (4): 「\*」是一个 **wildcard character**，代表任何英文字母或字符串皆可。此指令要寻找出前 3 个字母为 **gdp**，而后面接任何英文字母或字符串皆可的数据。结果有 **36** 笔资料合乎此条件。
- (5): 此操作比(4)更广，只要第一个英文字母为「g」的数据皆可。结果有 **39** 笔资料。
- (6): \*亦可出现在两个以上的地方。合于此条件的有 **6** 笔资料。
- (7): 「?」亦是一个 **wildcard character**，与「\*」不同处在于一个「?」代表「一个」英文字母，两个「??」代表「最多两个」英文字母。因此「?」比「\*」寻找范围较窄。此指令寻找所有第一个英文字母为 **g**，但 **objectname** 仅是 **2** 个或 **3** 个英文字母之数据。
- (8): 此操作寻找 **alphabetical range** 在 **gd** 与 **h** 之间的数据。
- (9): 此操作则列出所有 **niaq** 的数据，共有 **130** 笔。
- (10): 此操作寻找名字为 **j22** 之数据，结果有两笔，「**ind: j22.a**」与「**ind: j22.m**」。不同频率之同一笔数据在 **AREMOS** 里算是两笔不同资料。

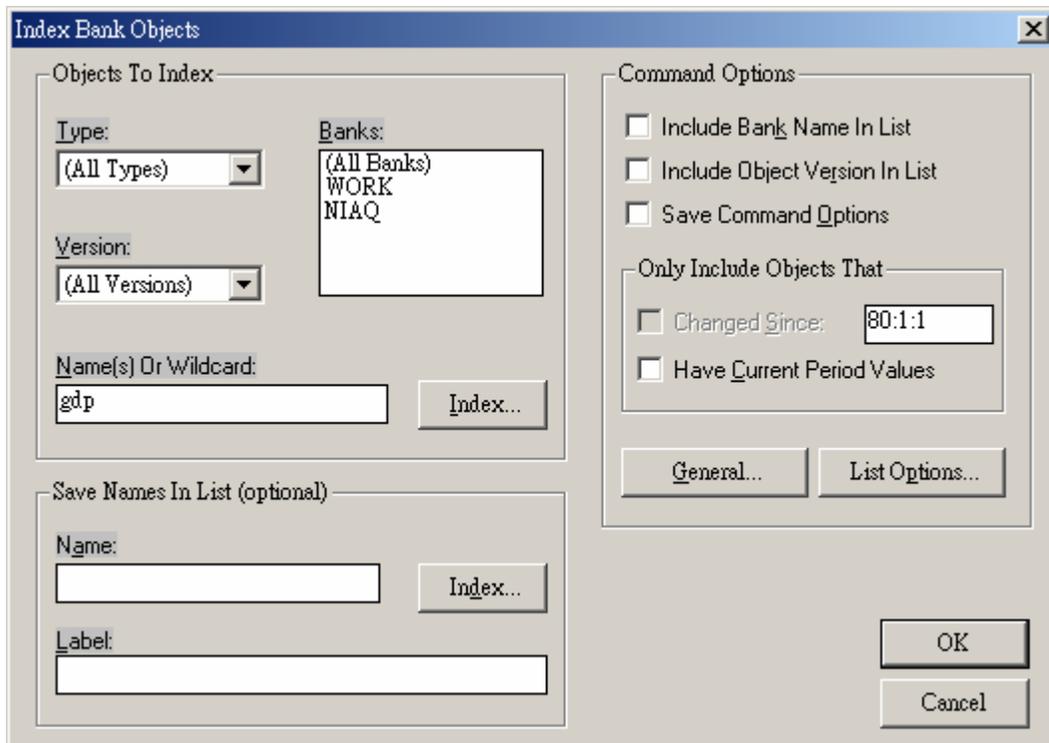


图 3.3.1

### 3.4 「Seriescode 在线查询法」

前 3.3 节所叙述的 **Index** 方法，是 AREMOS 软件里的基本检索在线(on line)查询方法，大概出现的检索代号，皆是没有设定格式的英文说明。而若使用本节的 **Seriescode** 在线查询法，可看到检索代号，是有格式的中文说明。操作如下：

在数据库光盘片的子目录下检索，进入<seriescode>子目录，点取「seriescode.htm」超文件文件即可看到所有的数据库检索手册。

注：若是查询国外检索手册，请您看看您的光盘有否

<Foreign\_doc>? 原则上是有加装 AREMOS 国外数据库的单位, 其光盘片才有此子目录。当然不管是<seriescode>子目录或 <Foreign\_doc>子目录皆可在财团法人经济资讯推广中心网站自由下载。

### 3.5 利用本中心的 Homepage 网页, 在线查询检索代号

财团法人经济资讯推广中心的 homepage 网址是:

[ <http://www.edu.tw/moecc/rs/pkg/tedc.htm> ]。当然有可能将来网站地址会更改, 但您只要在各大入口网站(如 Yam, Yahoo, Pchome...)敲入「财团法人经济资讯推广中心」或「aremos」即可找到。

由于近来 Internet 盛行, 本中心网站内已将国内外 30 几个数据库的检索手册及检索代号皆挂上去, 欢迎进入查询及下载(不锁任何 IP), 而且本中心也建立了一套「AREMOS 国内数据库检索代号搜寻引擎」方便 usrs。同时也可利用此网站查询其它关于「AREMOS 统计数据库」的最新相关讯息, 请多加利用。

# 第四章 股票市场数据库之特殊用法

## 4.1 股票市场数据库使用上的特点

第二章介绍的基本检索数据方法可适用于一般 AREMOS/Windows 数据库。但「台湾股票市场数据库」较特殊，在使用上稍有一点变化。特别在本章中讲解。

就数据检索而言，股票市场数据库有三个特点：

(1) 「日数据之检索」：股票市场资料主要是「日」 (d) 频率在操作上有关 per 的设定略有不同。

(2) 「跨数据库之检索」：由于股票市场数据库数据量过于庞大，我们分三个计算机文件储存：

**STOCK70**：含 1971 年 1 月 1 日—1979 年 12 月 31 日。

**STOCK80**：含 1980 年 1 月 1 日—1989 年 12 月 31 日。

**STOCK90**：含 1990 年 1 月 1 日—1999 年 12 月 31 日。

**STOCK00**：含 2000 年 1 月 1 日—目前。

因此，如果需要检索跨越不同计算机文件的时间序列，在方法上很不相同。

(3) 「collapse(or conv)方法的设定」：在第二章范例(三)已介绍 collapse 指令可叫取数据并同时转换成不同频率。但日数据一定要先 collapse 成月资料，才能再继续 collapse 成季或年资料。

AREMOS/Windows 无法由日数据直接 conv 成季或年数据。

以下我们已三个实际范例来说明股票市场数据库的使用方法。

## 4.2 范例(四): 检索时间序列日数据

假定我们要叫取 1976 年 1 月 1 日至 3 月 15 日之「台泥收盘价」、「台泥成交量」及「发行量加权股价指数」等三笔资料。由「EPS/AREMOS 台湾股票市场数据库检索手册」查知其检索代号分别为: s1101@cl, s1101@vol, js。进入 AREMOS/Windows 后, 其操作鼠标如下:

(1)Options → Set → Period → (Frequency:Daily, From 1976:1:1 to 1976:3:15) → OK

(2)Data → Bank → Open → Files → 磁盘驱动器(看看自己的数据库在哪一槽) → stock70.bnk → 确定 → Open → Cancel 跳回主画面

(3)Report → Print → 键入指令 s1101@cl, s1101@vol, js(在 Expression 处) → Execute

说明:

(1)此指令设定频率为「日」(d), 期间是 1976 年 1 月 1 日至 1976 年 3 月 15 日(1976:1:1 1976:3:15)。这是 AREMOS/Windows 特定要求的期间设定方法, 一定要年: 月: 日。此外, AREMOS/Windows 也接受 1976d1 1976d75, 代表 1976 年第 1 天至 1976 年第 75 天。

## 4.3 范例(五): 跨数据库之时间序列检索方法

假定我们要叫取「台风收盘价」与「国寿收盘价」两笔资料，期间是 1989 年 12 月 1 日到 1990 年 1 月 20 日。由「检索手册」查知此二笔数据之检索代号为 s1206@cl 与 s2805@cl。由于所要叫取的期间跨越 1989 年与 1990 年，1989 年部分数据存于 stock80，而 1990 年部分则存于 stock90，因此检索方法略有变化。假定我们已进入 AREMOS/Windows，操作鼠标如下：

(1) Options → Set → Period → (Frequency: Daily, From 1989:12:1 to 1990:1:20) → OK

(2) Data → Bank → Open → Files → 磁盘驱动器(看看数据那一槽) → stock80.bnk  
分别叫取 stock90.bnk → 确定 → Open → Cancel 跳回主画面

(3) 键入指令 series<1989:12:1

1989:12:31>x1=stock80:s1206@cl;(在主画面 Aremos Command 下)

(4) 键入指令 series<1990:1:1 1990:1:20>x1=stock90:s1206@cl;

(5) 键入指令 series<1989:12:1 1989:12:31>x2=stock80:s2805@cl;

(6) 键入指令 series<1990:1:1 1990:1:20>x2=stock90:s2805@cl;

(7) 键入指令 print<dec 2>x1,x2;

说明：

(1)：此操作是在设定频率与期间。见范例(四)之说明。

(3)：series 指令可让我们产生一个新的 series，在此这个新 series 命名为 x1，而所产生的 x1 会自动存在我们「目前所运作」的 bank 里(AREMOS 术语称此为 primary bank)。通常我们

一进入 AREMOS/Windows, 即会进入一个 **primary bank**。除非您别有指定, 此 **primary bank** 通常叫做 **work bank**。在运作过程中, 如有产生任何新 **series**(或任何新的 **object**), 都会自动存放在 **work bank** 里。series **x1=stock80:s1206@cl** 乃是产生一个新 **series**, 名叫 **x1**, 而其值在所设定的期间内等于 **stock80** 里的 **s1206@cl**。为何要特别指定 **stock80**? 因为 **stock90** 数据库里亦有 **s1206@cl**, 故需以「**bankname:**」来指定那一个 **bank** 里的 **s1206@cl**。在<>里指定期间为 **stock80** 所涵盖之部分, 及 **1989:12:1** 到 **1989:12:31**。因为 **stock80** 仅含我们所需要之 **1989** 年部分数据, 如果不把期间如此设定来执行 **series** 指令, 会产生一大堆 **missing(m)**值。使用<>来设定 **per**, 是 **local setting**, 其设定时效仅限于此指令。在第(1)式里使用 **set** 方式设定 **per** 则是 **global setting**, 其时效会继续为维持下去, 除非你再到 **Set Frequency & Period** 画面去重新设定。(有关 **global** 与 **local setting** 请参考第 6 章 6.2 节)。注意:**local** 设定 **period**, 不需打 **per 1989:12:1 1989:12:31**, 只打 **1989:12:1 1989:12:31** 即可。

(4): 此指令在 **local** 所设定的期间内(即 **1990:1:1** 到 **1990:1:20**) 给予 **x1** 值, 而令其值为 **stock90** 里之 **s1206@cl**。我们以<>设定期间为 **1990:1:1** 到 **1990:1:20** 乃 **stock90** 所涵盖部分。执行完此指令, 我们在 **work bank** 里产生了一个完整的时间序叫 **x1**, 其在 **global** 所设定的期间内(**1989** 年 **12** 月 **1** 日至 **1990** 年 **1** 月 **20** 日)含有由数据库 **stock80** 与 **stock90** 所叫取出之台风收盘价(**s1206@cl**)。

(5)(6): 见(3)(4)之说明。执行完此 2 个指令, 我们产生了一个完整的时间序列叫 **x2**, 代表国寿收盘价(**s2805@cl**)。

(7): 此指令可印出数据。由于我们已回到 **global setting**, 因此所印出的期间为第(1)式所 **set** 之 **1989:12:1** 到 **1990:1:20**。在此以 **<dex 2>** 设定印出小数点后两位。

## 4.4 范例(六): 转换日数据之频率并做回归分析

假定我们要对发行量加权股价指数(检索代号 **js**)与国内生产毛额(**gdp**)、狭义货币供给额(**m1a**)与一月期定期存款利率(**rmtd@1st**)进行回归计量分析。分析频率是季(**q**), 期间为 1971 年第 1 季到 1991 年第 4 季(**1971:1 1991:4**)。我们也已知 **gdp**, **m1a** 与 **rmtd1@1st** 分别存于 **AREMOS/Windows** 经济统计数据库系统里另外两个数据库: **gdp** 在国民所得季数据库(库名: **niaq**), **m1a** 与 **rmtd1@1st** 则在金融统计数据库(库名: **fsm**)。其操作鼠标如下:

(1) **Options** → **Set** → **Period** → (**Frequency:Monthly, From 1971:1 to 1991:12**) → **OK**

(2) **Data** → **Bank** → **Files** → 磁盘驱动器(看看数据库放在那一槽)

—————→ **stock70.bnk , stock80.bnk, stock90.bnk**

→ **确定** → **Open** → **Cancel** 跳回主画面

(3) 键入指令 **conv<1971:1 1979:12 tolerate=99>js=stock70:js.dave;**

(4)键入指令 **conv<1980:1 1989:12 tolerate=99>js=stock80:js.d  
ave;**

(5)键入指令 **conv<1990:1 1991:12 tolerate=99>js=stock90:js.d  
ave;**

(6)**Options → Set → Period → (Frequency:Quarterly, From  
1971:1 to 1991:4) → OK**

(7)键入指令 **conv js=js.m ave;**

(8)**Data → Bank → Open → Files → 磁盘驱动器(看看数据库  
放在那一槽)**

**niaq.bnk**

—————→  
    分别叫取 **fsm.bnk** → 确定 → **Open** → **Cancel** 跳回  
主画面

(9)键入指令 **conv m1a=m1a.m ave;**

(10) **Econ → Equation → OLS → 键入 equ1(在 Name 处) →  
键入 js(在 Dependent 处) → 键入 gdp(在 Independent 处) →  
Execute**

(11) **Econ → Equation → OLS → 键入 equ2(在 Name 处) →  
键入 js(在 Dependent 处) → 键入 m1a(在 Independent 处) →  
Execute**

(12) 键入指令 **conv rmtd1@1st@f=rmtd1@1st@f.m ave;**

(13) **Econ → Equation → OLS → 键入 equ3(在 Name 处) →  
键入 js(在 Dependent 处) → 键入 m1a,rmtd1@1st@f(在**

## Independent 处) → Execute

说明:

- (1): **js** 在数据库里是日数据。**AREMOS/Windows** 无法直接由日数据 **conv** 成季数据, 需先 **conv** 成月数据, 再由月数据 **conv** 成季数据。此指令先 **globally** 设定月的频率与期间。
- (3): **conv** 指令把数据库里的日数据找取出来转换成目前所设定的月频率, 而再 **work bank** 里形成一个新的月资料 **series**, 也取名叫 **js**。由于是跨数据库数据, 正如范例(五), 必须分 3 个指令来执行。本指令限于从 **stock70** 转换出 1971 年 1 月到 1979 年 12 月(1971:1 1979:12)期间的资料。在<>里 **locally** 设定 1971:1 1979:12 期间乃是 **stock70** 所涵盖部分。

在此指令中, 数据库里的原始日数据必须写明其「完整名称」**bankname:objectname.versionname**(亦即 **stock70:js.d**)。有两个原因: (i)我们需要标明 **bankname: (stock70: )**, 因为在已 **open** 的另外两个数据库(即 **stock80, stock90**)中存放有同一 **objectname(js)**之数据。(ii)我们也必须标明 **versionname**(即频率 **.d**), 因为目前所设 **freq** 是 **m**, 而原始数据是日数据。如果不标明 **js.d**, 计算机会当作 **js.m** 来处理, 但 **stock70** 里并无 **js.m** 这笔资料。一般说来, 使用 **collapse** 指令一定要标明「被 **collapse**」数据的 **versionname**。在范例(三)已碰过此种情形。

注: **tolerate=99** 是因转换日数据当中可能期间会有 **na** 的情况(如星期天不交易, 所以没有值); 如此计算机语言可略过 **na** 做转换, 最大的容忍度是 **99** 设定值。

- (4)(5): 此二指令各从 **stock80**, **stock90** 转换出所需的 **js** 数据。  
在 < > 里 **local** 设定 **stock80** 与 **stock90** 各自所涵盖的期间。执行至此, 我们的 **work bank** 里已存在一笔月数据, 名叫 **js**, 期间是 1971 年 1 月到 1991 年 12 月。
- (6): **global** 设定季频率与期间。
- (7): 把 **work bank** 里的 **js** 月资料(以 **js.m** 代表)转换成季数据。  
执行完此指令, 我们 **work bank** 里多了一笔 **series**, 即是 **js.q** 季资料。对 **AREMOS** 而言, **js.m** 与 **js.q** 是两个不同的 **objects**。
- (8)(9)(10)(11)(12)(13): 参考前面范例(三)之说明。

## 4.5 有关「上市公司股票报酬率统计数据库」

由于 **AREMOS/Windows** 能处理日资料之长度有限制, 因此「上市公司股票报酬率统计数据库」也分 4 个档案储存:

- (1)**STOCKRR**: 含所有年、季、月与周数据。
- (2)**STOCKRR7**: 含 1972 年 1 月 1 日-1979 年 12 月 31 日之日资料。
- (3)**STOCKRR8**: 含 1980 年 1 月 1 日-1989 年 12 月 31 日之日资料。
- (4)**STOCKRR9**: 含 1990 年 1 月 1 日-目前更新之日资料。

读者如要对日数据作跨数据库之检索, 可参考本章 4.3 节范例(五)的操作方法。

此数据库数据之检索代号亦采组合方式, 例入「台积电的周报率」为 **s2330@wrr**, 其中 **2330** 为「台积电」的公司代号, **wrr** 代表 **weekly rate of return**。又如「南亚的日报酬率」为 **s1303@drr** 其中 **1303** 为「南亚」的公司代号, **drr** 则代表 **daily rate of return**, 以此类推。详细可参考本数据库之检索手册简介。

## 第五章 如何使用打印机 (PRINTER)

以上介绍之操作均限于屏幕上的运作，在屏幕显现结果后即消失。本章讲解如何把操作结果送到打印机打印出来。有四种不同方法，分别介绍如下。

### 5.1 使用鼠标按键操作

**File** → **Print output**，即可印出你下前个指令所得的 **output**。若您觉得数据太多，只想段章印几列资料，可 **mark(block)**你所要的东西(按住鼠标左键不动，拖曳至你要的范围截止处，放开鼠标左键)，然后选 **File** → **Print output** 即可。

### 5.2 使用<Alt><Print Screen>键

按<Alt><Print Screen>可把目前看到的屏幕画面送到 **Clipboard**(暂存区)，然后您可用 **Microsoft** 里的 **Word or Excel** 软件，在剪贴簿上贴上。(这和 5.1 是不大相同)

以上两种方法均是直接把屏幕上显示的从打印机印出。除此之外，我们也可把所要的结果存放在一个 C 碟上的 **file**。以下将介绍之 **sf(Save-File)**与 **bf(Bigsave-File)**则是把出现在屏幕上的结果，包括 **input** 与 **output**，形成一个 **file**。然后就在 **Windows** 记事本下或 **Microsoft Word** 来文书编排修正，且可把它从打印机印出。

### 5.3 sf (Save-File) 与 sfc (Save-File-Completed)

要把屏幕上出现之结果存在一个 file, 可使用 sf(Save-File 之简称)与 sfc(Save-File-Completed 之简称)系列指令。以下面范例来说明。假定已进入 AREMOS Windows:

(1)Options → Set → Period → (Frequency:Quarterly, From 71:1 to 94:4) → OK

(2)Data → Bank → Open → Files → 磁盘驱动器(看看数据库放在那一槽)

————→ niaq.bnk  
分别叫取 fsm.bnk, setup.bnk → 确定 → Open →

Cancel 跳回主画面

(3)键入指令 `conv m1a=m1a.m ave;`

(4)键入指令 `conv m1b=m1b.m ave;`

(5)键入指令 `sf;`

随着问题回答, 输入: aaa

(6)Econ → Equation → OLS → 键入 equ1(在 Name 处) → 键入 m1a(在 Dependent 处) → 键入=gdp(在 Independent 处) → Estimate

(7)键入指令 `rmsf;`

输入 aaa

(8)键入指令 `sf;`

输入 bbb

(9)Econ → Equation → OLS → 键入 equ1(在 Name 处) → 键入 m1a(在 Dependent 处) → 键入=gdp(在 Independent 处) → Estimate

(10) 键入指令 `sfc;`

以上操作，(6)的结果在(7)被 **removed** 掉，因此<WAREMOS>子目录里不再有 **aaa.lst** 这个 **file**。最后只有(9)的结果存在 **bb.lst** 档案里。注意：在第二章范例(二)已提醒过，在这些「形成 **file**」的运作中，如果取了相同的 **filename**，则新形成的会完全盖过现存的。因此读者要小心，免得失去了想保存的档案。

此外，尚有「非交谈式」**sf** 指令可使用。**sf** 是「交谈式」指令：下了 **sf**，计算机间您 **filename**，紧接着输入 **filename**。读者亦可直接下 **sf filename**，例如：键入指令 **sf abc**；其结果与下了 **sf** 指令后，再输入 **abc** 完全一样。

## 5.4 有大量结果而不愿在荧光幕上出现者：**bf**

### (**Bigsave-File**) 与 **bfc** (**Bigsave-File Completed**)

有时读者只想把执行结果形成一个 **file**，而希望这些结果不在屏幕上出现。此种需求可能是要印大量数据(四、五十页或上百页)。如果在荧光幕上出现，必须不断按 **enter** 键，很麻烦。

要把执行结果形成一个 **file**，而不在屏幕上出现，需使用 **bf**(**Bigsave-File** 之简称)与 **bfc**(**Bigsave-File Completed** 之简称)系列指令。其使用方法与上节 **sf**，**sfc** 一样，只不过下了 **bf** 指令后，屏幕及空白(**mute**)。一直要等到下了 **bfc** 后，屏幕才会重新出现。

下了 **bf** 指令，执行了一阵子，如果读者改变主意，想要 **remove** 掉刚所保留的 **file**，则正如上节之 **rmsf**，在此亦可下 **rmbf**(**Remove Bigsave-File**)之指令来达到相同效果。

此外，亦可用「非交谈式」**bfl** 指令。例如：

```
bfl abc;
```

其结果与下了 **bf** 指令后，再输入 **abc** 完全一样。

※注：sf,sfc,bf,bfc 的功能皆存在 setup.bnk 里，所以别忘了 open setup.bnk 数据库。

## 第六章 常用指令之说明

前几章从「实用」观点来介绍如何使用 AREMOS/Windows 检索数据并做简单回归分析。如果您不在乎了解 AREMOS/Windows 语言结构，依那些范例，套上自己研究所需数据，应可顺利完成操作。

本章将回头来对那些指令进一步补充说明。

### 6.1 General 与 Object Commands

AREMOS/Windows 的 commands(指令)分两大类：**general command** 与 **object command**。其中 **object command** 专用为产生某种 **object**(例如 **series**)；而其它非用来产生 **object** 之一般性指令均属 **general command**。

例如范例(一)中：

(3)Options → Set → Period(Frequency : Quarterly, From 1982:1 to 1992:4) → OK

(4)Options → Set → Report(Decimal Places: 0) → OK

(5)Data → Bank → Open → Files → 磁盘驱动器(看看数据库放在那一槽) → niaq.bnk → 确定 → Open → Cancel 跳回主画面

(6)Report → Print → 键入 gdp,cp ,gdp96(在 Expression 处) → Execute

其中(5)(6)均属 **general command** 语句。其语句一般型式是：

**commandname<options>argumentlist;**

以(6)式而言, **print** 是 **command**; **gdp**、**cp**、**gdp96** 是 **arguments**。而针对某些 **arguments**(以逗点分开), 来执行某一动作(在此是 **print**)。在 AREMOS/Windows 下, 在此(6)式有以<>设定 **options**, 当你再 **Execute** 处按下鼠标后, 在主画面上, 可发现一段指令 **print<Format=Column Horizon=8000 shift No>gdp,cp,gdp96 Width:10 Decimal=2**; 如此(4)式的 **global** 设定就没有影响; 而(3)式的 **global** 设定依然存在。

其次范例(五)中:

(3)键入指令 **series<1989:12:1 1989:12:31>x1=stock80:s1206@cl;**

(4)键入指令 **series<1990:1:1 1990:1:20>x1=stock90:s1206@cl;**

此二式均属 **object command** 语句, 其一般型式是:

**objecttype<options>objextname=definition;**

以(3)式而言, 是以 **series** 指令(**objecttype**)产生一个 **series**, 名叫 **x1(objectname)**, 而 **x1** 的内容则定义为与 **stock80** 数据库里之 **s1206@cl** 一样。

## 6.2 Global 与 Local Setting

执行某一个 **command**, 不管是 **general command** 或 **object command**, 都需要有 **options** 来设定执行的环境(**environments**)。

例如对于 **print** 指令, 我们需要设定频率(**freq**)与期间(**per**)。而 **print gdp,cp,gdp96** 则再所设定的频率与期间下打印出那三笔数据。

**Options** 设定有两种方式: 以 **options → set** 方法来设定叫 **global setting**, 其有效期间会继续维持到下次再 **options → set** 为止; 以三角形括号<>来设定是 **local setting**, 其有效仅限于执行该语句, 一执行完又回到原先的 **global setting**。以上面范例(一)(3)(4)(5)(6)为

例，当执行：

**(3)Options → Set → Period(Frequency : Quarterly, From  
1982:1 to 1992:4) → OK**

即是 globally 设定频率为 q，期间为 1982:1 到 1992:4。接着下来执行 (6)式即是在该频率与期间下 print 指令。如果我们加上：

**(7)键入指令 print<1985:1 1986:4>pchya(gdp96);**

**(8)键入指令 analyze pchya(gdp),pchya(gdp96);**

则第(7)是仅会印出 1985:1 到 1986:4 期间之资料(这是典型的 local setting)。但第(8)是又回复到 global 设定之 1982:1 到 1992:4 来执行 analyze 指令。

### 6.3 跨行下指令或同一行下多数指令

AREMOS/Windows 里的 AREMOS Command 环境下指令最多可有 127 个英文字母。有时一个指令语句太长一行打不完，需接第二行或第三行。此时与剧中若有逗点，可以逗点做结尾按 ENTER 键，计算机变了解指令尚未完全而需接下行。例如：

**(1)Data → Bank → Open → Files → 磁盘驱动器(看看数据库放在那一槽) → man.bnk → 确定 → Open → Cancel 跳回主画面**

**(2)键入指令 print ru15@19,ru20@24,ru25@29,ru30@34,**

**(3)键入指令 ru35@39,ru40@44,ru45@49;**

若不以逗点做一行之结尾，则计算机便视语句以完整而开始执行。记住，每下一行指令完，结束后的动作是用分号(;)来代表。如果语句中没有逗点可供结尾，则打「!」为结尾，计算机亦会「续下行」。

例如：

(1)键入指令 `series<1989:12:1 1989:12:31>x1=!`

(2)键入指令 `stock80:1206@cl;`

但记得，不可使用「!」符号把一个 `objectname` 分开在两行，否则会  
得到 `error message`。

我们亦可再同一行下两个以上之 `AREMOS/Windows` 语句。这些  
语句必须以分号( ; )分开。例如：

(1)键入指令 `set freq m ; set per 1990:1 1991:12;`

(2)键入指令 `open man ; print n;`

## 6.4 加减乘除与开方

`AREMOS/Windows` 加减乘除符号为 `+ - * /`。开方则为 `**`。例如  
`2**3` 是 2 的 3 次方。

## 6.5 Work Bank 的说明

`AREMOS/Windows` 的 `object` 一定得存在于某个 `bank` 里。我们  
所建好的数据库，如 `niaq`, `ind`, `fsm` 等等，都是一些 `banks` 放在 C  
碟里(其在 C 碟的 `filename` 为 `niaq.bnk`, `ind.bnk`, `fsm.bnk` 等等)。进  
入 `AREMOS/Windows` 后，一旦 `open` 某个 `bank`，即可对该 `bank` 里  
的 `series` 做操作，例如 `print`，或进行回归分析。

当我们进入 `AREMOS/Windows` 时，事实上我们是进入一个  
`primary bank` 里，这个 `primary bank` 除非您特别指定，计算机自  
动取名为 `work bank`(在 C 碟里 `filename` 即为 `work.bnk`)。

刚开始进入 `AREMOS/Windows` 时，这个 `work bank` 应该是空空  
的，没有任何 `object`。在操作过程中，如有产生任何 `object`(如 `series`,  
`conv`, `equation` 等指令都会产生 `object`)，则这些 `objects` 都会自动存  
放在此 `work bank`。任何时候，如果在主画面下：

键入指令 **index work;**

可查知目前 **work bank** 里有什么 **objects**。

离开 **AREMOS/Windows** 时，如果键入指令「**stop;**」或按活属 **File** → **disconnect**，计算机会把目前的 **work.bnk** 保留起来，下次再进入 **AREMOS/Windows** 时，仍是以此 **work.bnk** 为 **primary bank**，里面的 **objects** 也都还存在可供使用。但如果离开 **AREMOS/Windows** 只打「**stop**」，没有分号「**;**」，则计算机会问是否要 **erase work bank**。您可以打 **y** 或 **n** 来决定是否要杀掉目前的 **work bank**。

在操作过程中，如果想要清除目前 **work bank** 里所有 **objects**，可键入指令「**clear;**」或把鼠标移至 **Data** → **Bank** → **Clear** → **Work**，如此，便可以有一个干净的 **work bank** 重新开始。

## 6.6 Series 的「完整名称」

在第三章 3.3 节已提及，一个 **series** 的「完整名称」事实上由 3 部分组成：

**bankname:objectname.versionname**

例如「工业生产总指数(月数据)」的完整名称是：

**ind:jqind.m**

在此，**versionname** 其实就是「频率」。

我们可用「完整名称」来下指令，例如：

键入指令 **print ind:jqind.m;**

如此，很明确没有任何混淆。但如果我们下指令：

键入指令 **print jqind;**

此时，有可能不同 **banks** 里同时存在 **jqind**，或同一 **bank** 里存有不同 **version**(即频率)之 **jqind**。计算机如何执行呢？其原则如下：

(1)首先看目前的频率设定。如果是 **m**，则计算机会执行 **print**

**jqind.m**, 如果是 **a**, 则计算机执行 **print jqind.a**。

(2) 假定目前频率设定是 **m**, 计算机即开始在所 **open** 的数据库群中寻找 **qind.m**。其寻找的优先级: 第一是 **primary bank**(即 **work bank**), 其次是最近一次 **open** 的 **bank**, 接着再往前, 越晚 **open** 的 **bank** 越优先。此优先级可以键入 **q banks** 指令看出或在主画面下的 **Primary Bank: Work** 处, 按鼠标左键一下, 看看 **Query Banks** 处, 即知。

(3) 当计算机第一次找到 **jqind.m**, 便立刻执行 **print jqind.m**, 而结束此指令的运作, 不再继续寻找是否其它 **banks** 也存 **jqind.m**。

由上述可见, 如果有必要, 应该用「完整名称」来下指令才不会引起混淆。

## 6.7 如何产生 Series

**AREMOS/Windows** 的 **objects** 可分不同 **type**(种类)。到目前我们接触最多的 **object type** 是 **series**(即时间序列数据)。第七、八章中会讲解另外两种 **types of objects: list** 与 **procedure**。

当我们进入 **AREMOS**, 即是进入一个 **primary bank(work bank)**, 里面空空没有任何 **object**。我们可以 **open** 一些 **banks**, 利用那些 **banks** 里的 **objects** 来进行运算分析。但有时我们会需要「产生」的新 **objects**。这些自己产生出来的 **objects** 都会自动存放在 **primary bank** 里。第七、八章将介绍如何产生 **list** 与 **procedure**。本节则谈谈如何产生 **series**。

依需求之不同, 我们有「五」种方法可以在 **primary bank** 里产生 **series**:

(一) 有时读者所需之数据在现有数据库群中并没有, 必须自己输入。可利用 **series** 指令来进行。例如键入指令(在主画面下):

**(1) Options → Set → Period(Frequency:Quarterly, From**

**1985:1 to 1986:4) → OK**

**(2)键入指令 series y1=301,324.5,355,398,458 repeat 3,550;**

如此可产生一个新的 series, 名叫 y1, 其数值则是等号后所输入的数字。所设定季频率从 1985 年第 1 季到 1986 年第 4 季, 一共需要 8 个数字, 因此要输入刚好 8 个数字才是一个完整的语句。输入数字可用 repeat 来简化。458 repeat 3 代表连续个 458。如果 repeat \*, 则代表 repeat all。如果是 missing value, 打 m 即可。

(二) 使用 series 指令亦可利用现有的 series 来产生新的 series。例如范例(五)跨数据库之时间序列检索方法:

**(3)键入指令 series<1989:12:1 1989:12:31>x1=stock80:s1206@cl;**

**(4)键入指令 series<1990:1:1 1990:1:20>x1=stock90:s1206@cl;**

在此乃利用已 open 之 stock80, stock90 里之 s2610@cl 来产生一个新的 series x1。或者=后面也可直接使用数学符号, 例如:

**(1)Options → Set → Period → (Frequency:Quarterly, From 1980:1 to 1992:4) → OK**

**(2)Data → Bank → Open → Files → 磁盘驱动器(看看数据库放在那一槽) → niaq.bnk → 确定 → Open → Cancel 跳回主画面**

**(3)键入指令 series s=gnp-cp-cg;**

**(4)键入指令 display s;**

在此定义一个新的 series, 叫 s, 它的值即是 gnp-cp-cg。

(三) collapse(or conv)指令可转换数据频率。在 AREMOS/Windows 里, 不同频率(version)的数据算是不同 object, 因此 conv 指令应视为「产生」新的 series, 而产生的新频率 series 也存放在 primary bank 里。例如范例(三):

(3)Options → Set → Period → (Frequency:Quarterly, From 1971:1 to 1994:4) → OK

(4)Data → Bank → Open → Files → 磁盘驱动器(看看数据库放在那一槽)

————→            niaq.bnk  
      分别叫取        fsm.bnk → 确定 → Open → Cancel 跳回  
主画面

(5)键入指令 `conv m1a=m1a.m ave;`

在此以 `conv` 指令所产生之 `m1a`,其「完整名称」应该是「`work:m1a.q`」,与数据库里被用来转换之原始数据「`fsm:m1a.m`」是不同 object。

(四)`copy` 方法可把数据原封不动从数据库拷到 `primary bank`。例如:

(1)Options → Set → Frequency (Quarterly) → OK

(2)Data → Bank → Open → Files → 磁盘驱动器(看看数据库放在那一槽)

————→            niaq.bnk  
      分别叫取        fsm.bnk → 确定 → Open → Cancel 跳回  
主画面

(3) Data → Bank → Copy → 键入 `gnp,jqind,m`(在 `Names or Wildcard` 处) → OK

如此可从已 open 的 banks 中找出 `gnp.q` 与 `jqind.m` 这两个 series 并拷到 `primary bank`。在此 `gnp.q` 与所设定频率相同,故不需打出 `versionname` 「.q」。 `jqind.m` 之 `versionname` 则不能省,因为频率不同。如要谨慎些,可打出数据的「完整名称」。

如:

      Data → Bank → Copy → 键入 `niaq:gnp.q, ind:jqind.m`(在 `Name or Wildcard` 处) → OK

注意: document 无法 copy, 因牵涉到 structure 的问题。

### (五) Series Editor 编辑方法

Data → Series → Edit → 在 Series Editor and Display 画面上输入 series 的 name → OK → 选 New or Update 去编辑 → Mark all → 按绘图工具(详细请参考前(范例二 第(19)式) 说明)。

## 6.8 AREMOS Functions

AREMOS 的 functions 可分五大类:

1. Main functions: log, exp, sqrt, max, min, mean, abs, round, variance, sum, std, median 等。
2. Difference functions: pch, pchya, diff, diffya, dlog 等。
3. Shift functions: switch, step, spike, missing 等。
4. Date functions: dateday, datemonth, dateyear, dateyday, after, before, calendar, period 等。
5. Financial functions: npv, irr, cirr, future, payback 等。

其执行均以 functionname() 来使用。这些 functions 的详细说明, 参考原文手册「Reference Manual p.262~279」。下面以简单范例说明其使用方法:

(1) 键入指令 set freq q; set per 1980:1 1992:4;

(2) 键入指令 series lgdp=log(gdp);

(3) 键入指令 equ eq1 log(cp96)=log(gdp96);

(4) 键入指令 print

pch(gdp96),pchya(gdp96),diff(gdp96),diffya(gdp96);

说明:

- (2): 可产生一个新的 series 叫 lgdp, 其数值即是对 gdp 取 log。

- (3): 此指令对 **cp96** 与 **gdp96** 做 **loglinear** 回归。亦即对 **cp96** 与 **gdp96** 取 **log** 后再进行回归。请参考范例(三)第(11)式之说明。
- (4): 此指令印出 **gdp96** 「与上期比之成长率(**pch**)」, 「与去年同期比之成长率(**pchya**)」, 「与上期比之数值差额(**diff**)」, 「与去年同期比之数值差额(**diffya**)」。 **pchya** 代表「**Percentage Change from a Year Ago**」。

## 6.9 「说明」(Document)

在 **AREMOS/Windows** 数据库系统里, 国内自己所建立的那些数据库, 每一笔数据均有三至五行的「说明」以供查询。这些说明通常包括该数据之全名、单位、来源等等。要看这些说明, 在 **open banks** 后, 以 **write** 指令即可:

键入指令 **set freq q;**

键入指令 **open niaq;**

键入指令 **write gdp96,cp96;**

或用鼠标操作:

(1)**Options** → **Set** → **Frequency (Quarterly)** → **OK**

(2)**Data** → **Bank** → **Open** → **Files** → 磁盘驱动器(看看数据库放在那一槽) → **niaq.bnk** → **确定** → **Open** → **Cancel** 跳回主画面

(3)**Report** → **Write** → 键入指令 **gdp96,cp96**(在 **Name** 处) → **OK**

※注明: 再 **Write** 一笔 **Object** (如 **gdp96** 或 **cp96**)之前, 必须先 **set** 这笔的频率即 **Open bank**。或是在 **Open bank** 后, 直接 **Write gdp96.q,cp96.q** 即可。

## 6.10 如何建立 Databank

读者亦可建立自己的 databank 放在 C 碟里。首先：

键入指令 `open<pri>bankname;`

此指令可指定开启某一数据库(新的或已存在的皆可)为 **primary bank**。接着以 6.7 节所介绍之方法「产生」新的 **series**。所产生的 **objects** 都会自动存放在此 **primary bank**。结束后以「`stop;`」离开 AREMOS。此 **primary bank** 便会储存在 C 碟下。

## 6.11 index, delete 与 clear

在第三章已介绍 **index** 指令可用来查询 **primary bank** 与 C 碟上各个数据库里存什么 **objects**。**Delete** 指令则可除掉 **primary bank** 或 C 碟上数据库里的 **objects**。

假定目前运作的 **work bank** 里有 **x1, x2, x3, x4, x5** 等 5 个 **objects**，而 C 碟上已建有「`mybank.bnk`」内含 **y1, y2, y3, y4, y5** 等 5 个 **objects**。则：

键入指令 `delete x2,x3;`

可除掉 **work bank** 里 **x2** 与 **x3** 这两个 **objects**。而在键入「`open mybank;`」后：

键入指令 `delete mybank:y3,mybank:y5 ;`

可除掉 **mybank** 里 **y3** 与 **y5** 这两个 **objects**。

此外 **index** 指令里用的 **wildcard characters** 「\*」与「?」也是用在 **delete** 指令。例如：「`delete g*;`」, 「`delete h??;`」, 等等。

若你要 **delete list** 的东西(如 **n1**)，必须键入指令 `del list n1;`

任何时候，如果：

键入指令 `clear;`

可除掉 **primary bank** 里所存 **objects**，而有一个干净的 **work bank**

重新开始。如果：

键入指令 `clear mybank;`

则会把 C 碟尚 `mybank.bnk` 里所存 `objects` 除掉，同时 `mybank.bnk` 也从 C 碟被 `erase` 掉。

`index` 与 `delete` 当然亦可透过鼠标执行。

(1) `Data` → `Bank` → `Index`

(2) `Data` → `Bank` → `Delete`

## 6.12 `rename.... as....`

`rename` 指令可把现有 `series` 的 `name` 改成令个 `name`。例如：

键入指令 `rename a,b,c as x,y,z;`

可把 `primary bank` 里面 `a`, `b`, `c` 3 个 `series` 改名为 `x`, `y`, `z`。又：

键入指令 `rename mybank:a,mybank:b,mybank:c as x,y,z;`

可把 `mybank` 里数据库里 `a`, `b`, `c` 3 个 `series` 改名为 `x`, `y`, `z`。又：

键入指令 `rename list a as x;`

可把 `primary bank` 里面的 `list a` 改名为 `x`。又：

键入指令 `rename procedure a as x;`

可把 `primary bank` 里面的 `procedure a` 改名为 `x`。

## 6.13 小数点与 `report columns` 的设定

我们 `print` 数据时，小数点以下几位数会依当时 `globally` 所设定的 `report decimals` 而印出几位。如要改变设定位数，可 `globally` 改变，例如：

(1) `Options` → `Set` → `Report(Decimal Places:2)` → `OK`

(2) 键入指令 `print jqind;`

或 `locally` 改变，例如：

(3)键入指令 **print<dec 2>jqind;**

此外, **print** 数据时, 每笔数据所占 **report columns** 数字, 计算机一开始(**default**)设定为 10 位。我们亦可 **globally** 改变其设定, 例如:

(4)**Option → Set → Report (Filed Width:8) → OK**

(5)键入只领 **print gdp96;**

或 **locally** 改变, 例如:

(6)键入指令 **print<col 8>gdp96;**

一般说来, 把 **report columns (Field Width)** 设定改少些, 可使我们一页里容纳更多笔数据, 但如果数据本身位数超过 **report columns** 数字的, 会改用「E 符号」表示, 看你来就很不方便。读者可自行斟酌。

任何时候, 键入指令 **q report;** 可查看当时有关 **report** 方面 **decimals, columns** 及其它方面的设定。**q** 是 **query** 之简称, 请参考下节。

## 6.14 help 与 query

(1)在 **AREMOS/Windows** 里直接按 **Help** 即可找寻你要的详细说明。

(2)**query**(简打为 **q**)可查询某个 **option** 的 **setting**, 或某个 **command** 之相关 **options** 的 **setting**, 例如:

键入指令 **q freq;**

可显示目前 **freq** 的设定。而:

键入指令 **q per;**

可显示目前 **per** 的设定。如果:

键入指令 **q print;**

则显示有关 **print** 指令之 **options** 的设定。又上节已提过, 如果:

键入指令 **q report;**

可显示所有有关 **report options** 的设定。

## 6.15 Option 设定的储存：store 与 restore

一进入 AREMOS/Windows 里，计算机自动对每一个 options 有预先的设定，此种设定称为 **default setting**。例如 **freq** 的设定是 **a**，**per** 的设定是 **80 85** 等等。在您操作过程中，如果改了 option 设定，而希望把这些设定储存起来，可用 **store** 与 **restore** 指令。例如，在任何时候：

键入指令 **store my**;

则在 **<WAREM32>** 子目录下会出现一个 file，叫 **my.opt**。**my** 是随便取的 filename。**my.opt** 会储存当时的 option 设定。将来如果下：

键入指令 **restore my**;

则会把 **my** 所储存的 option 设定变成是目前的 option 设定。当然，你可用鼠标操作：

(1) Option → Store → 键入指令 **my** (在 Store In File Name) → OK

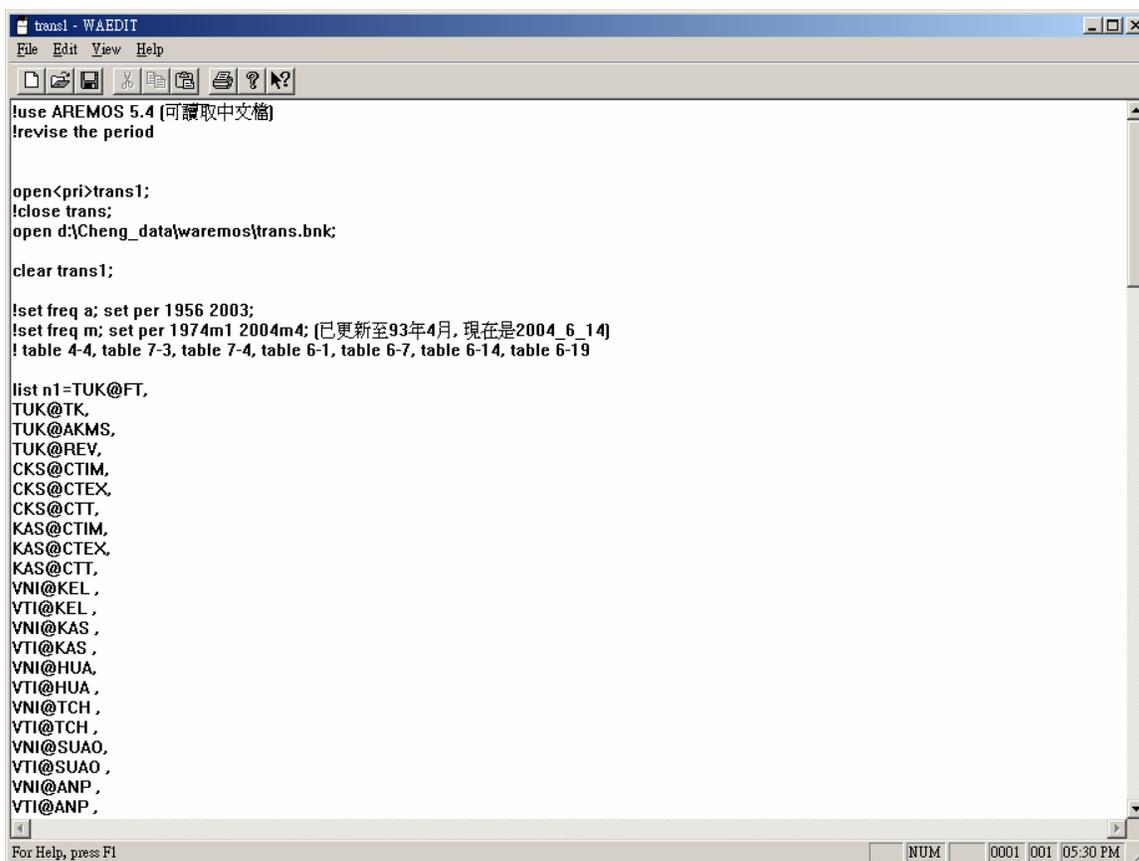
(2) Option → Restore → 键入指令 **my** (在 File Name) → OK

## 6.16 在 AREMOS/Windows 里进行编辑

在 AREMOS/Windows 里有「**WAEDIT**」的编辑程序，可直接对 C 碟的 file 做编辑，操作方法是鼠标点出：

Tools → AREMOS Editor → File → Open → 磁盘驱动器 (看看你的 file 放在那一槽) → 档案类型(选 All files) → **abc.prn** → 确定

如此在 **WAEDIT** 编辑系统下，可修修改改您要的东西，然后可存盘，也可印出(**print**)，(如图 6.16.1)。



```
trans1 - WAEDIT
File Edit View Help
!use AREMOS 5.4 (可讀取中文檔)
!revise the period

open<pri>trans1;
!close trans;
open d:\Cheng_data\waremos\trans.bnk;

clear trans1;

!set freq a; set per 1956 2003;
!set freq m; set per 1974m1 2004m4; [已更新至93年4月, 現在是2004_6_14]
!table 4-4, table 7-3, table 7-4, table 6-1, table 6-7, table 6-14, table 6-19

list n1=TUK@FT,
TUK@TK,
TUK@AKMS,
TUK@REV,
CKS@CTIM,
CKS@CTEX,
CKS@CTT,
KAS@CTIM,
KAS@CTEX,
KAS@CTT,
VNI@KEL,
VTI@KEL,
VNI@KAS,
VTI@KAS,
VNI@HUA,
VTI@HUA,
VNI@TCH,
VTI@TCH,
VNI@SUAO,
VTI@SUAO,
VNI@ANP,
VTI@ANP,
For Help, press F1
NUM 0001 001 05:30 PM
```

图 6.16.1

## 6.17 「递延」(lag) 与「领先」(lead)

AREMOS/Windows 以 [ ] 符号附于时间序列后面来代表「递延」或「领先」。例如： $gdp[-1]$ 代表  $gdp$  的递延(lag)一期， $gdp[-2]$ 代表递延二期，以此类推。而  $gdp[+1]$ 代表领先(lead)一期， $gdp[+2]$ 代表领先二期，等等。

## 第七章 LIST

List 是 AREMOS/Windows 语言里很方便而强有力的工具。简单说，一个 list 是「a collection of names」，代表一群 objects。如果要反复对某一群 objects 做操作，我们可先产生一个 list 来代表这群 objects。只要对这个 list 做操作，就相当于对那群 objects 做操作。同时 list 本身也是一种(type) object，可存在 bank 里；或也可以 C 碟上的一个 file 来保存，随时取用。

本章介绍三种不同的方法产生 list。

### 7.1 使用 List 指令来产生 list

最简单产生 list 的方法是使用典型的 object command，即以 list 指令来产生一个 list。看下面范例说明：

- (1) 键入指令 `list n1=gnp,cp,gnp96;`
- (2) 键入指令 `list listf n2=gdp,gdp96;`
- (3) 键入指令 `set freq q; set per 1985:1 1986:4; open niaq;`
- (4) 键入指令 `print #n1;`
- (5) 键入指令 `print #(listf n2);`
- (6) 键入指令 `display n1;`
- (7) 键入指令 `view n2.lst;`
- (8) Tools → AREMOS Editor → Files → Open → 磁盘驱动器(C 槽) → All files(在档案类型处) → n2.lst → 确定
- (9) File → Exit
- (10) 键入指令 `open<pri>mybank;`
- (11) 键入指令 `copy list n1;`

此时可键入指令「**stop;**」出来，或 **File** → **disconnect** 出来，稍后再进入 **AREMOS**：

(12) 键入指令 **open mybank;**

(13) 键入指令 **copy list n1;**

说明：

(1)：在第六章 6.1 节曾介绍产生 **object** 的 **object command** 一般型式是：

**objecttype objectname=definition;**

在此乃是以 **list** 指令(**objecttype**)产生一个 **list**,

名叫 **n1(objectname)**，而 **n1** 的内容即为 **gnp**、**cp**、**gnp96**，3 个 **names**。同时 **n1** 本身是一个 **object**，被产生后即存放在 **work bank** 里。

注意：您可直接键入指令，也可用鼠标操作，**Data** → **List** → 键入 **n1**(在 **Name** 处) → 键入 **gnp,cp,gnp96**(在 **list Item Text** 处) → **OK**(如图 7.1.1)。

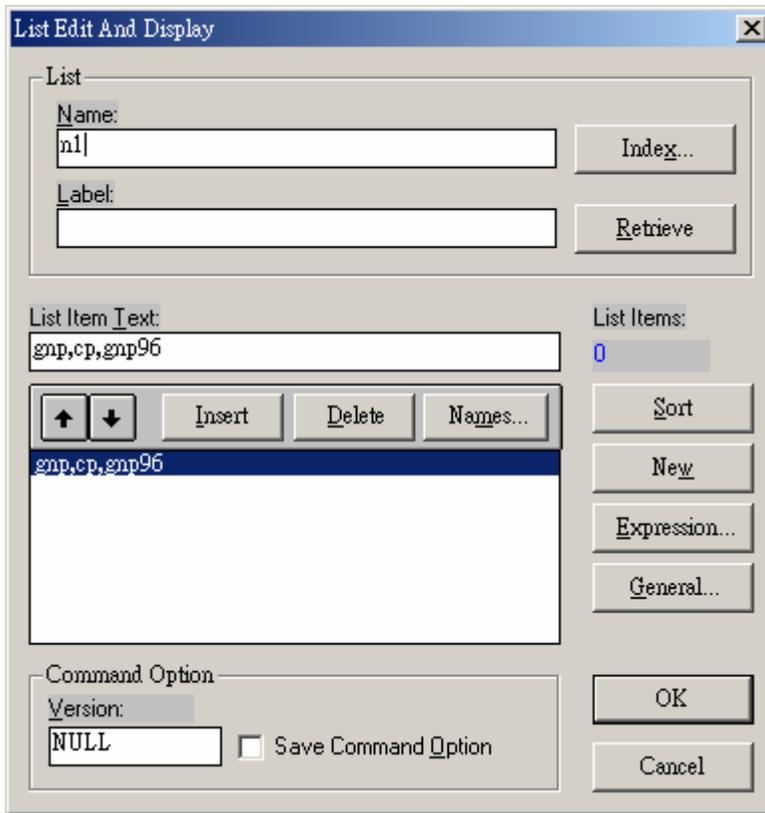


图 7.1.1

- (2): 此指令可产生另一个 list, 名叫 **n2**, 其内容为 **gdp, gdp96** 二个 **names**。但在 **list** 指令后面如果加上 **listf**(即 **listfile** 之简称), 则所产生的 **n2** 并不是存在 **work bank** 里的一个 **list**, 而是存在 C 碟上 <WAREM32> 的一个 **file** (就叫做 **list file**), 其 **filename** 为 **n2.lst**。若您用鼠标操作, 在 **List Edit And Display** 的画面上, 必须键入 **listfn2** (在 **Name** 处)。
- (4): 一旦产生 **n1**, 我们可用 **n1** 来替代它所代表的 3 个 **objects**。使用方法乃在前面加「#」符号。在此, **print #n1** 相当于 **print gnp, cp, gnp96**。因 **n1** 只含 3 个 **objects**, 尚感觉不出 **list** 的威力。如果 **n1** 含上百个或上千个 **objects**, 便可了解 **list** 可节省我们不少 **key in** 的人力时间。

- (5): 此指令与(4)式相同, 只是 C 碟上的 **list** 后面总要加 **listf**。此指令相当于 **print gdp,gdp96**。
- (6): 此指令可印出 **n1** 所代表的 3 个 **names**。
- (7): 范例(二)已介绍过, **view** 指令可查看 C 碟上的 **file**。
- (8): 使用 **AREMOS Editor** 去查看 **n2.lst file**, 如有必须, 可用此编辑程序来增加或减少 **n2** 里的 **names**。
- (9): 跳出 **AREMOS Editor** 系统。
- (10): **open** 一个新的 **bank** 叫 **mybank**, 同时指定其为 **primary bank**。原来的 **work bank** 已不再是 **primary bank**。
- (11): 此指令把 **work bank** 里的 **n1 copy** 到 **primary bank**(在此为 **mybank**)以便能够把 **n1** 这个 **object** 放在 **mybank** 里保存起来。注意: **copy** 可把任何 **object** 从已 **open** 的数据库中拷到 **primary bank**。

其一般句型是:

**copy objecttype objectname;**

但如何所要 **copy** 的是 **series**, 则 **objecttype**(即 **series**)不需打出来。否则必须打 **objecttype**(在此为 **list**)。当然, 如果您用鼠标操作, 则是:

**Data** → **Bank** → **Copy** → 键入 **list n1** (在 **Name or Wildcard** 处) → **OK**

- (12)(13): 将来再进入 **AREMOS**, 可把存放在 **mybank** 里的 **n1** 随时 **copy** 出来使用。不过, 如以 **listfile** 型式保留在 C 碟, 例如 **n2.lst**, 则随时可使用, 不需如此拷来拷去。

## 7.2 使用 **Index** 指令来产生 **list**

上节所介绍以 **list** 指令产生 **list** 方法, 乃是土法炼钢, 把 **names**

逐一输入。现在介绍第二种产生 list 方法则是利用 **Index Bank Objects** 的方法。第三章已介绍过，**index** 方法可在线(online)查询已 open 的 banks 中之 object，尤其可利用 wildcard characters 「\*」与「?」。Index Bank Objects 后，在「Save Names In List」处，键入 n3(为设定 list 的 name)，则计算机会把 index 所查询到的所有 object names 自动形成一个 list，而以该指定的 list name(即 n3)为此新产生的 list 的 name。以下面范例说明(用鼠标操作)：

- (1) **Data** → **Bank** → **Open** → **Files** → 磁盘驱动器(看看自己的数据库放在那一槽) → **niaq.bnk** → 确定 → **Open** → **Cancel** 跳回主画面
- (2) **Data** → **Bank** → **Index** → 键入指令 **niaq:gnp\*.q**(在 **Name(s) or Wildcard** 处) → 键入指令 **n3** (在 **Save Names In List** 的 **Name** 处) → **OK**
- (3) **Data** → **Bank** → **Index** → 键入指令 **niaq:cp\*96.q** (在 **Name(s) or Wildcard** 处) → 键入指令 **listf n4**(在 **Save Names In List** 的 **Name** 处) → **OK**
- (4) **Data** → **Bank** → **Display** → 键入指令 **n3**(在 **Name** 处) → **OK**
- (5) 键入指令 **view n4.lst;**
- (6) 键入指令 **set freq q;**
- (7) 键入指令 **copy #n3;**
- (8) **Data** → **Bank** → **Index** → 键入指令 **work:\*,\***(在 **Name(s) or Wildcard** 处) → 键入 **n5**(在 **Save Names In List** 的 **Name** 处) → **OK**
- (9) **Data** → **Bank** → **Index** → 选 **series**(在 **Type** 处) → 键入 **work:\*. \*** (在 **Name(s) or Wildcard** 处) → 键入指令 **n6**(在 **Save Names In List** 的 **Name** 处) → **OK**

说明：

- (2)：此操作一方面做 **index** 查询，同时把所找到的 **object names** 形成一个 **list** 叫 **n3**，内含 3 个 **names**。
- (3)：此操作一方面做 **index** 查询，同时把所找到的 **object names** 形成 C 碟上<WAREM32>子目录下的一个 **listfile**，叫 **n4.lst**，内含 13 个 **names**。(操作方法，如图 7.2.3)

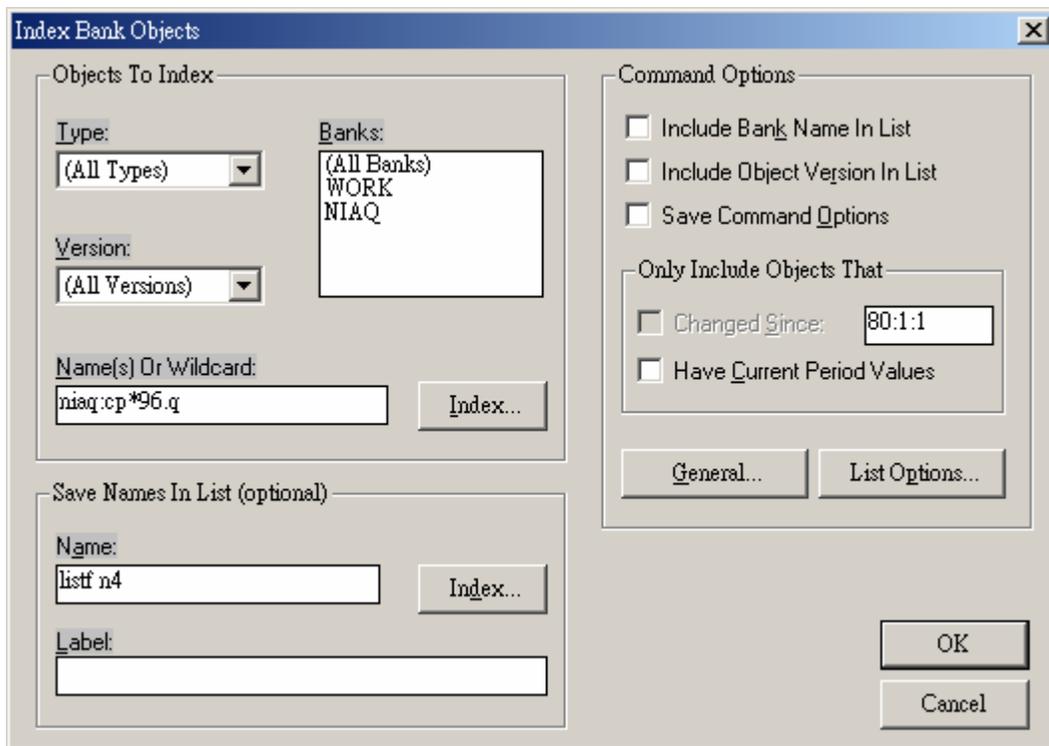


图 7.2.3

- (4)(5)：见 7.1 节之说明。
- (6)(7)：第 (7) 式把 **n3** 所代表的 3 个 **objects** 从 **niaq** 数据库拷到 **primary bank**。但从执行第 (4) 式 **display** 指令可看出一个

- list 所含 object names 并无 versionname。因此要先 set freq q, 与所要 copy 的数据频率相同, 否则会得到 error message。换句话说, copy #n3 相当于 copy gnp, gnp96, gnpexp96。而在 set freq q 下, 则相当于 copy gnp.q, gnp96.q, gnpexp96.q。
- (8): 此操作 index work bank 上所有 objects, 含 3 个 copy 过来的 series 以及执行 (2) 式所形成的 list n3, 并把这些 object names 形成一个 list 叫 n5。
- (9): 此操作 index work bank 所有属于 series 的 objects, 在此即是 copy 过来的 3 个 series, 而把这些 series names 形成一个 list 叫 n6。Index objecttype 乃限定某类 object 做 index, 否则会 index 所有 objects。(操作如图 7.2.9)

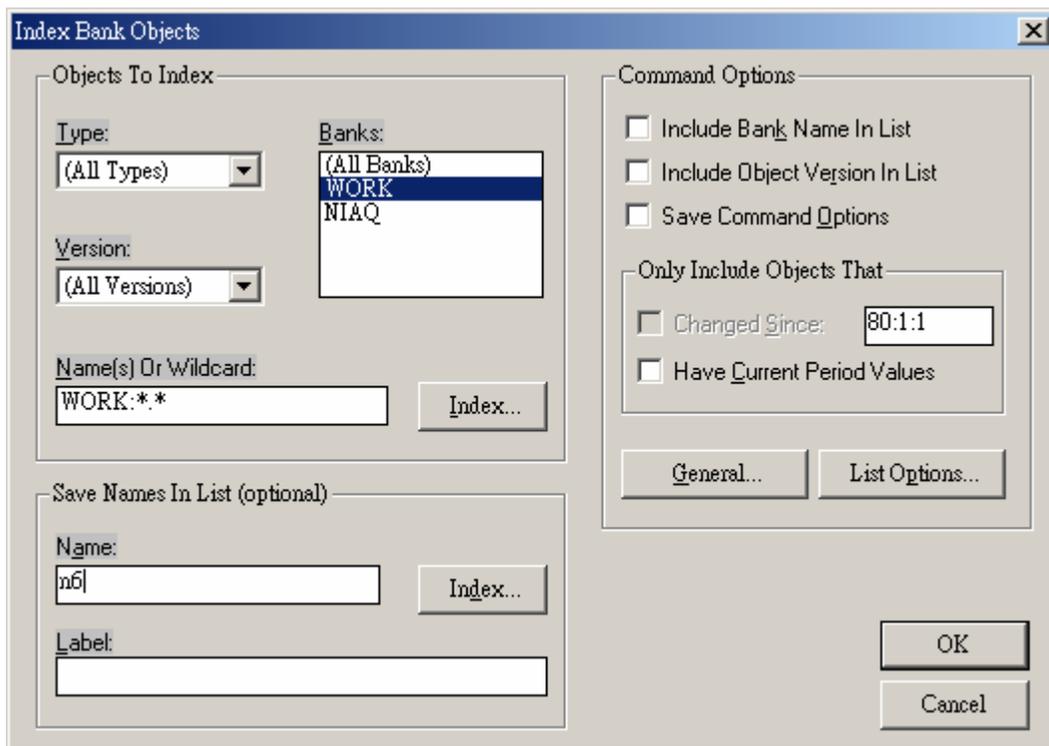


图 7.2.9

注：用 **Index Bank Objects** 方式做成的 **List** 是有限制 **objects** 数目，若数据库里的笔数很多皆想做成 1 个 **List**，需要修改 **AREMOS** 系统里的设定：**Options** → **Set** → **Databanks...** → 键入 **4000** (在 **Limit Number of items Found to**) → **OK** 。

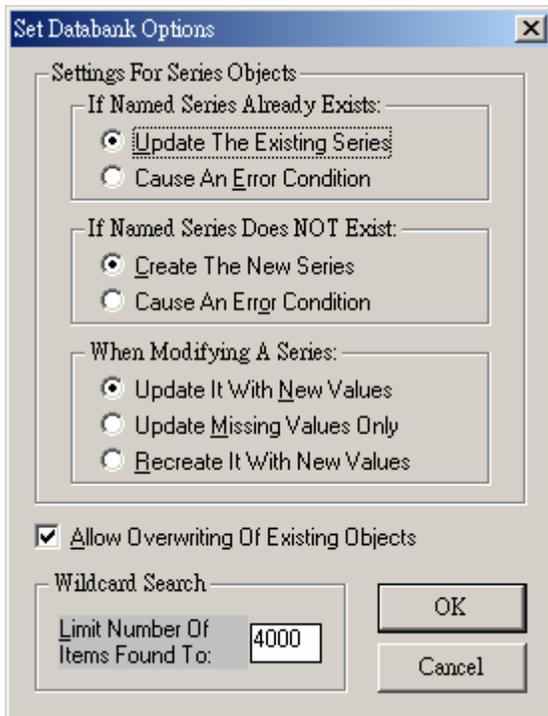


图 7.2.10

### 7.3 使用 Set Operator 来产生 list

如果 **primary bank** 里已建有一些 **lists**，或 **C** 碟上已建有一些 **listfiles**，我们可利用 **union(+)**，**difference(-)**，**intersection(\*)**，**concatenation(, )**等 **set operators** 来产生新的 **list** 或 **listfile**。假定 **primary bank** 里已有 **n1**，**n2** 两个 **lists**，而 **C** 碟上有 **m1.lst**，**m2.lst** 两个 **listfiles**。则：

(1)键入指令 `list n3=#n1+#n2;`

- (2) 键入指令 `list n4=#n1-#n2;`
- (3) 键入指令 `list n5=#n1*#n2;`
- (4) 键入指令 `list n6=#n1,#n2;`

说明：

- (1)：此指令展生新的 `list n3`，是 `n1` 与 `n2` 的 `union`。在其它操作方法上，也可使用鼠标，`Data` → `List` → `OK` 进去修改如图 7.3.1。

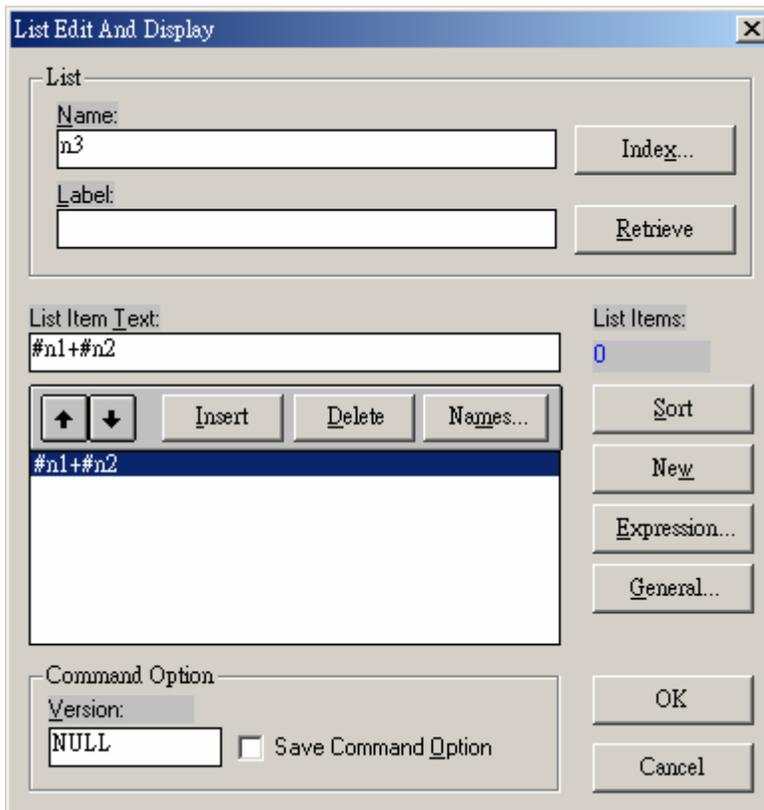


图 7.3.1

- (2)：`n4` 是 `n1` 去掉与 `n2` 有相同 `names`。
- (3)：`n5` 是 `n1` 与 `n2` 的 `intersection` (交集)。

(4): **n6** 与 **n3** 的差别在于, 如果 **n1** 与 **n2** 有重复, 则 **n3** 只取一个, 而 **n6** 会重复放进去。

此外, 我们亦可:

(5): 键入指令 **list m3=#(listf m1)+#(listf m2);**

(6): 键入指令 **list listf m4=#(listf m1)+#(listf m2);**

(7): 键入指令 **list k1=#n1[1-200];**

(8): 键入指令 **list k2=#n1[201-266];**

说明:

(5): 在 **primary bank** 里产生一个 **list** 叫 **m3**, 其内容是 **listf m1** 与 **listf m2** 的 **union**。

(6): 在 C 碟 <warem32> 子目录上产生一个 **listf** 叫 **m4.lst**, 其内容也是 **m1** 与 **m2** 的 **union**。

(7)(8): 假若 **list n1** 里的笔数(**objects**)超过 200 笔以上, 则必须分割, 在此分割成 **k1, k2** 两个 **list**。

## 第八章 PROCEDURE 与 OBEY

**Procedure** 是 AREMOS/Windows 语言里另一个强有力的工具。简言之，一个 **procedure** 是「a sequence of AREMOS commands」，代表一群 AREMOS/Windows 指令语句。如果要反复执行一群 AREMOS/Windows 指令语句。可先产生一个 **procedure** 来代表这群指令语句。任何时候，只要键入这个 **procedure** 的名字，便会执行该群指令。同时 **procedure** 本身也是 **object** 的一种(**type**)，一旦建好了 **procedure**，可存放在 **bank** 里。

**Procedure** 的一般形式如下（键入指令）：

```
procedure name;  
command statement;  
command statement;  
.....  
.....  
.....  
end;
```

第一行必须是「**procedure name;**」，其中 **name** 是我们为此 **procedure** 所取的名字。最后一行必须是「**end;**」。其间则是此 **procedure** 所代表的指令群。

有两种方法可产生 **procedure**：(1)使用 **Procedure Edit and display** 编辑程序，(2)使用 **Procedure** 指令。分别介绍如下。

### 8.1 使用 **Procedure Edit and Display** 编辑程序产生

## Procedure

假定我们要建一个 **procedure**，名叫 **pr1**，内容如下：

**Tools** → **Procedure Edit** → 键入指令 **pr1**(在 **Procedure Name** 处)

→ 在下面的框框键入 **procedure** 内容。

```
procedure pr1;
set freq q ; set per 1989:1 2004:3 ;
set report dec 2;
open e:\WAREMOS\niaq;
print gnp,pchya(gnp);
end;
```

→ **Save**

→ **Close** 跳回主画面

→ 键入指令 **pr1** ;

说明：这是建立一个 **procedure** 的第一种方法，名字叫 **pr1**，(如图 8.1)，若要执行 **pr1**，只要在主画面上，键入指令「**pr1;**」(在 **AREMOS Command** 上)，若您想修改 **pr1** 里的内容，只要再进入 **Procedure Edit...**，**Index...**叫出 **pr1**，或键入 **pr1** (在 **procedure name** 处)，按 **Retrieve**，然后光标移到 **Procedure** 内容去修改即可，然后按 **Save**，再按 **Close**，回到主画面，在 **AREMOS Command** 下再键入「**pr1;**」执行。

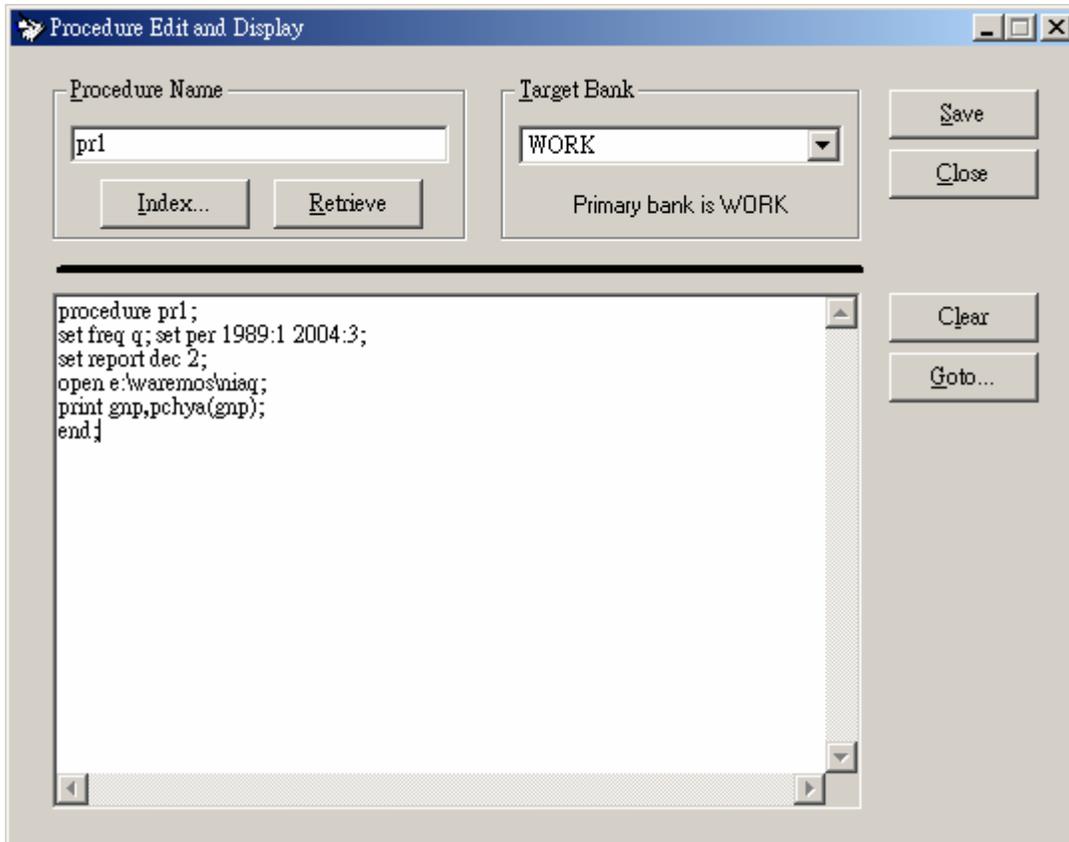


图 8.1

任何时候只要打「**pr1;**」, 即会执行那群 **command statements**。同时 **pr1** 也是一个 **object**, **object type** 是 **procedure**, 它可以像一个 **series** 一样存放在 **bank** 里面。如果在主画面键入:

**display pr1;**

即会印出 **pr1** 的内容。

※在此特别一提, 在 **procedure** 里的 **command statements**, 有一列是 **open e:\WAREMOS\niaq**; 这是到 E 磁盘驱动器里的 **WAREMOS** 子目录下, **open niaq** 数据库的意思。若您的数据库 **niaq** 是放在 G 槽的 **WAREMOS** 子目录下, 您就必须打入「**open g:\WAREMOS\niaq**」。

## 8.2 使用 Procedure 指令产生 Procedure

第二种产生 procedure 的方法是以 procedure 指令。在 AREMOS 里，如果在主画面上键入指令：

```
procedure pr2;
```

此时计算机进入 sequential mode，会出现

```
Sequence of commands ending with;
```

```
Enter END;as final command:
```

这是 sequential mode 的 prompt，接着依序打入：

```
set freq m; set per 91:1 92:12;
```

```
open e:\WAREMO\ind;
```

```
print jqind,j22;
```

```
end;
```

每行按 ENTER 时，都会出现「sequence of commands ending with ; Enter END; as final command:」。最后打「end;」即会跳出 sequential mode 回到正常 AREMOS/Windows 环境。此时也建好一个 procedure，名叫 pr2。要修改 pr2，可用 8.1 节所介绍之 Procedure Edit and Display 编辑程序去修改。

## 8.3 CMD file 与 obey

AREMOS/Windows 基本上是交谈式(interactive)语言。但也可用 batch job 执行。有时读者要反复使用同一群 AREMOS/Windows command statements，或有时一个 job 可能执行很久，则采 batch 作业比较方便。AREMOS/Windows 的 batch job 可透过 2 个方式进行。

(一)利用本章所介绍建立 procedure 的方法。一旦建好

**procedure**，可存于 **bank** 里随时可叫出来执行，或略加修改再执行。**procedure** 代表一群 **command statements**，执行某一 **procedure** 即相当于做一个 **batch job**。

(二) 也可用 **CMD file** 来做。假定我们 C 碟上已建好一个 **file**，**filename** 随便取，扩展名则规定为 **CMD**，而此 **file** 内容即是一堆 **AREMOS/Windows command statements**。如此，进入 **AREMOS/Windows** 后，找 **Tools** 里的「**obey**」去执行 **filename(.cmd)** 即可。

※ 至于如何再 C 碟上产生一个 **CMD file**?

可用 **AREMOS Editor** 编辑程序来编辑，同时设定扩展名为「**.cmd**」。下面，我们即将使用一个范例来解释。操作鼠标如下：

(1)**Tools** → **AREMOS Editor** → 然后在 **NEW-WAEDIT** 的画面下键入 **command statement**：

```
set freq m;set per 1989m1 2004m11;  
set report dec 2;  
open e:\WAREMOS\fsm.bnk;  
print rmt1 @ 1st@f,pch(rmt1 @ 1st@f);
```

→ **File** → **Save** → 键入 **ob1.cmd**(在文件名称处) → 确定

(2)**Tools** → **Obey** → 键入 **ob1.cmd**(在 **File Name:**) → **OK**

(3)**Tools** → **AREMOS Editor** → **File** → **Open** → 选 **ob1.cmd**  
→ 确定

说明：

(1)这一群 **command statements** 的前面不须写「**procedure name;**」及最后一行不必写「**end;**」，这和 **procedure** 的编辑方式有点不同，但相类似。而 **open e:\WAREMOS\fsm.bnk**，是 **open** 磁盘

驱动器 E 槽子目录 WAREMOS 里的 fsm 数据库(如图 8.3.1)。  
读者要特别留意自己的数据库是放在那一槽或在那一个子目录下。

(2)用 obey 去执行 ob1.cmd 档案，如图 8.3.2。

(3)回到 AREMOS Editor 编辑程序下，去修正 ob1.cmd 档案。

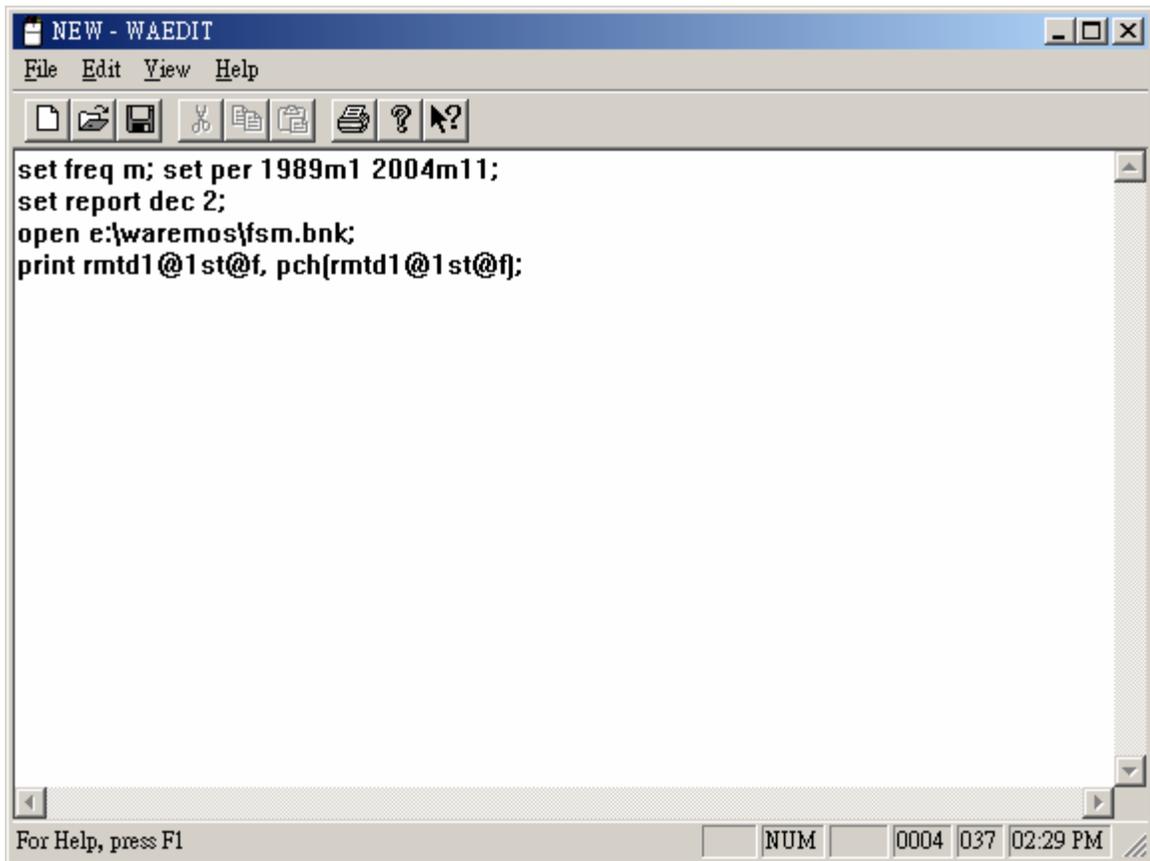


图 8.3.1

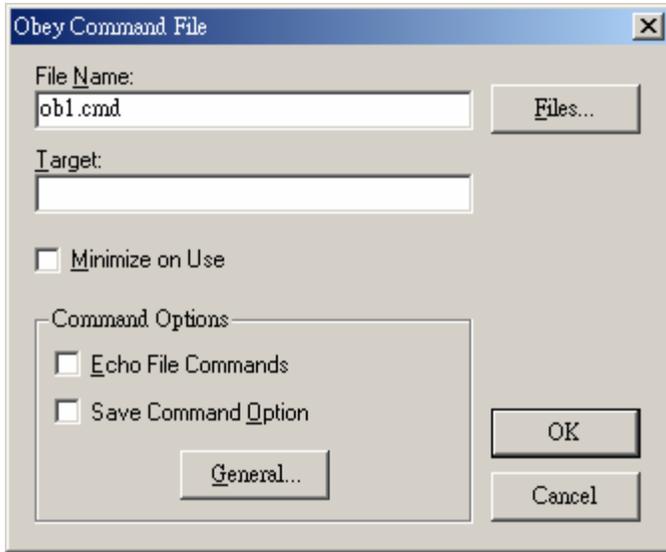


图 8.3.2

## 第九章 stocklist、complist、compflist、 compreport、compfreport 之使用

在 AREMOS/Windows 数据库系统里，「股票市场数据库」(stock70, stock80, stock90, stock00)与「上市公司财务报表数据库」(comp, compf)是比较大的数据库。有时使用者常会需要从这些数据库叫取大量数据，例如：

「所有上市公司的收盘价」

「台泥公司财务报表里所有项目」

「所有公司的税前纯益」

类似这样的需求，如果以范例(一)方法，势必输入上百个检索代号。为此，我们设计了五个 procedure 来简化这方面的操作：stocklist, complist, compflist, compleport, compfreport。这些 procedure 放在一个 bank 里叫 setup，每次进入 AREMOS 计算机必须先 **open** **setup.bnk**。因此随时都可用这些 procedure。其用法与功能分别说明如下。

### 9.1 Stocklist

stocklist 可就「股票市场数据库」(stock70, stock80, stock90, stock00)之检索代号，依使用者之指定来产生一个 list。例如：可指定「所有水泥股之收盘价与成交量」或「所有上市公司之收盘价」等等。一旦产生 list，即可利用此 list 来叫取数据，或作其它用途。由于 stock70, stock80, stock90, stock00 三个计算机档案内之上市公司家数并不相同，因此使用者必须先指定由那一个计算机档案内之上市公司来形成 list。

**stocklist** 是交谈式 **procedure**，只要在主画面下：

键入指令 **stocklist**;

接着随问题回答即可。

## 9.2 **Complis**、**Compflis**

**Complis** 可就「上市公司财务报表数据库」(**comp**)之检索代号，依使用者之指定来产生一个 **list**，但限于「一般产业股」。例如：可指定「某一家」或「某些」公司之「某些」或「所有」资产负债表项目。这也是交谈式 **procedure**，请在主画面 **AREMOS Command** 下：

键入指令 **complis**;

再随着问题回答即可。

另一 **compflis**，是针对「金融产业股」来指定产生一个 **list**，操作与前相似。

## 9.3 **Compreport**、**Compfreport**

**compreport** 可从「上市公司财务报表数据库」叫取某一家公司之数据而自动形成「资产负债表(**Balance Sheet**)」，或「收支损益表(**Income Statement**)」，但只针对「一般产业股」。而且由于屏幕宽度限制，一次最多只能有 3 年数据之报表。同时，所形成之报表在屏幕上出现后，使用者可指定是否要再 **printer** 打印，以及是否要形成一个 **C** 碟上的 **file**。使用方法：先开启 **comp** 数据库，然后在主画面下：

键入指令 **compreport**;

再随着问题回答即可。

另一 **compfreport**，是针对「金融产业股」来产生报表，操作与前相似。

## 第二部分

分布图，制表，绘图

## 第十章 分布图(Scatter & Plot)

有关分布图之绘制，以下以两个范例加以说明。范例一是利用 `scatter` 指令；范例二是利用 `plot` 指令。

### 10.1 范例

`scatter` 的标准指令如下：

Scatter ‘標題’ 數列一 ‘數列一之圖例’ 數列二 ‘數列二之圖例’；

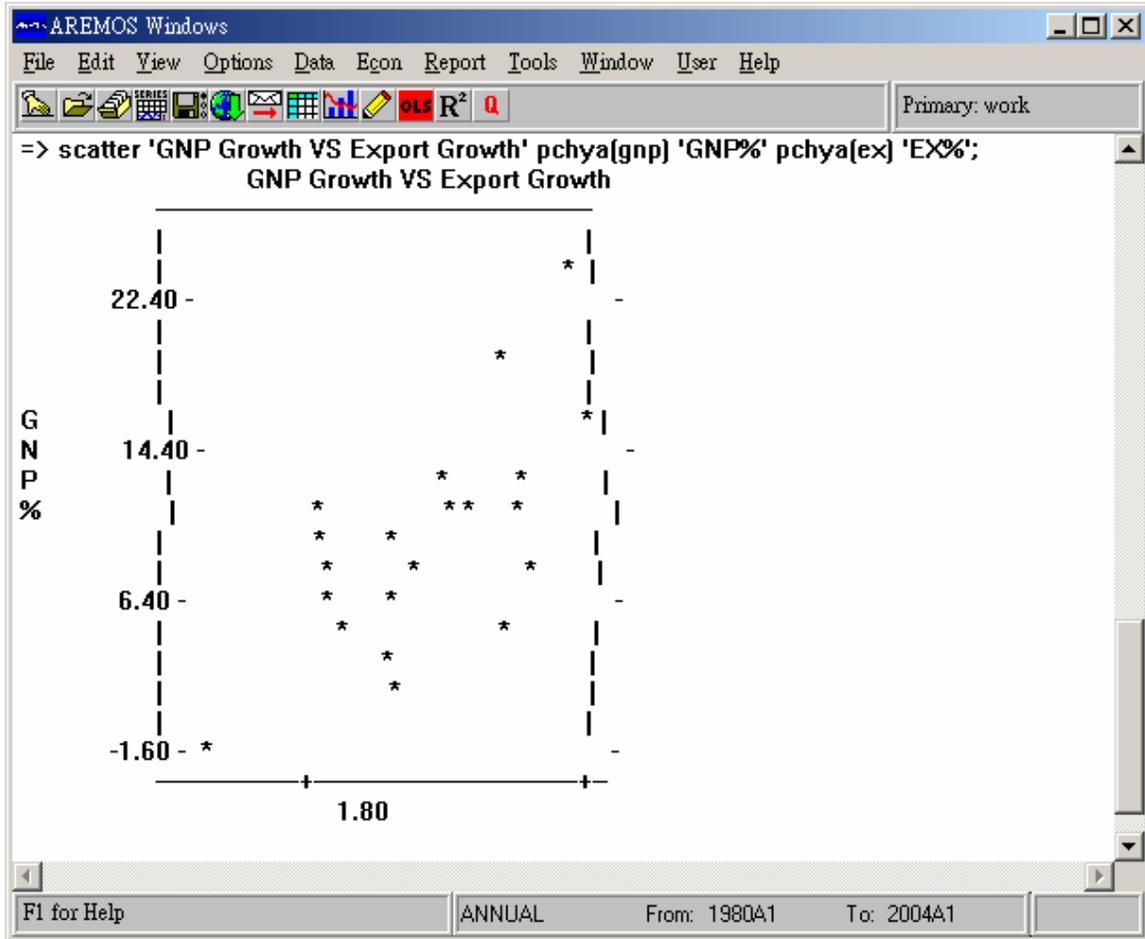
假设我们想要知道经济成长率与输出成长率间的关系，`scatter` 指令可以绘制二者的分布图。鼠标操作如下：

(1) Options → Set → Period (Frequency:Annual, From 1980 to 2004) → OK

(2) Data → Bank → Open → Files → 磁盘驱动器(看看数据放在那一槽) → niaa.bnk → 确定 → Open → Cancel 跳回主画面

(3) 键入指令 `scatter ‘GNP Growth VS Export Growth’ pchya(gnp) ‘GNP%’ pchya(ex) ‘EX%’;`

可得到以下图形：



## 10.2 范例

**plot** 的标准指令如下：

plot ‘标题’ 数列一 ‘数列一之图例’， 数列二 ‘数列二之图例’；

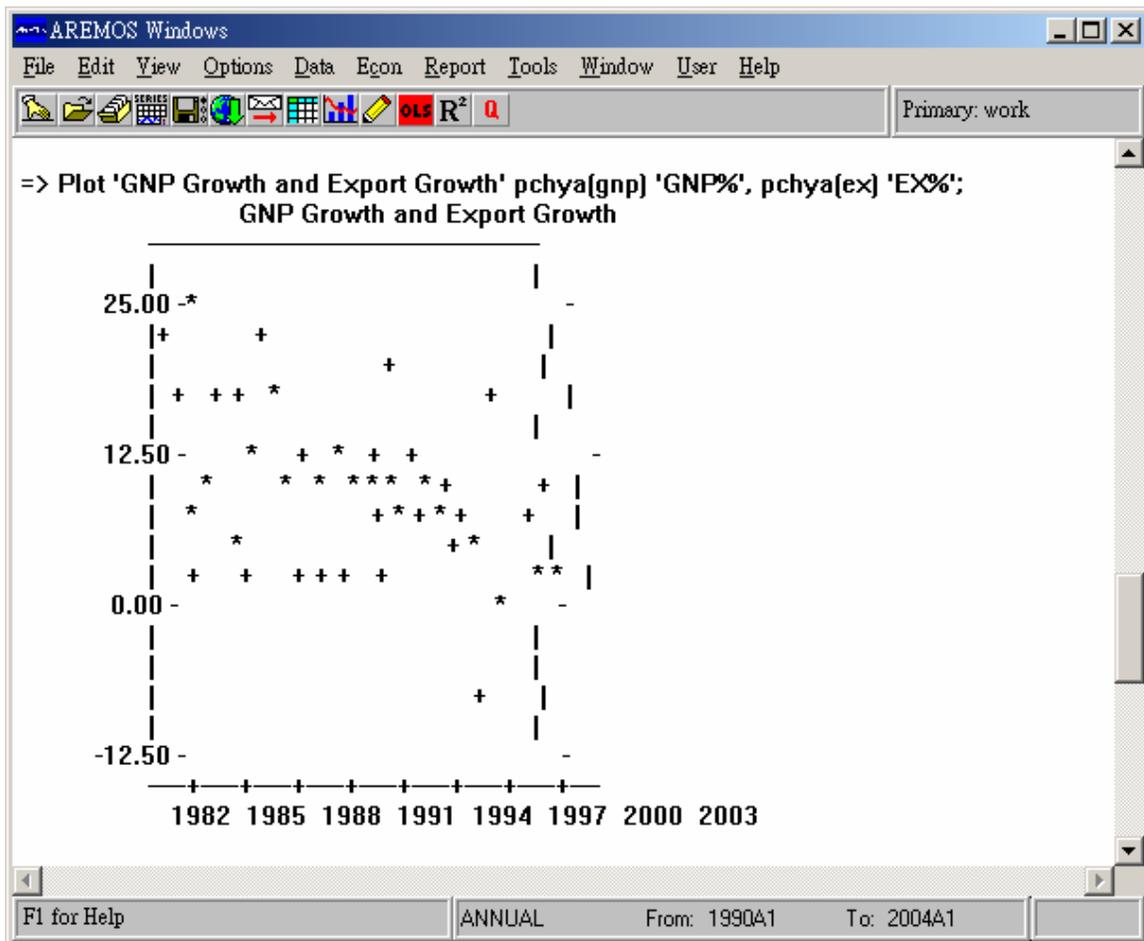
假设我们想要知道经济增长率及输出成长率趋势，**plot** 指令可以绘制二者对时间的分布图。相关指令如下：

**(1)Options → Set → Period (Frequency:Annual, From 1980 to 2004) → OK**

(2) **Data** → **Bank** → **Open** → **Files** → 磁盘驱动器(看看数据库放在那一槽) → **niaa.bnk** → 确定 → **Open** → **Cancel** 跳回主画面

(3) **Plot 'GNP Growth and Export Growth' pchya(gnp) 'GNP%', pchya(ex) 'EX%';**

可得到如下图形：



※记住 **scatter** 和 **plot** 的指令最大不同在于前者不须「,」, 而后者需要加上「,」。

# 第十一章 AREMOS/Windows Table 制表的简介

AREMOS/Windows 表格制作(Interactive Table), 提供下列功能:

- 可直接从数据库叫出数据到表格。
- 可修改表格上的数据, 并可存回数据库中。
- 可在表格上加入说明文字。
- 可以用工作底稿方式进行横列(rows)或纵行(columns)运算。
- 可以储存表格, 供往后修改或打印用。

## 11.1 表格步骤

表格之制作, 首先须进入表格画面, 其步骤为:

操作鼠标

Tools → Table → File → New (或按工具列中的 Table 制表工具)

即可进入新的表格画面。选 File 栏可点出您的所需的档案或新编个档。

## 11.2 表格画面

进入表格画面后, 可以从屏幕上看到由「时间(Data)轴」及「变量(Label)轴」所为成的表格区(Table section), 其中时间轴, 位于屏幕上方之反白空格(编号由左至右增大), 系用来定义纵行数据之时间; 变量轴, 位于屏幕左边之反白空格(编号由上至下增大), 系用来定义横列数据之变量名称; 表格区是表格的本身, 为输入数字及文字的地方。表格打印时, 只会印出表格区内的数据, 时间轴及变量轴内的设定是不会随同印出。

时间轴与变量轴之位置亦可互调。惟屏幕上方反白空格至多容纳 240 个字母，而屏幕左边反白空格则至多，允许 120 列(rows)，而上面反白空格则至多允许 50 行(columns)，如图 11.2。

任一时间纵行与变量横列之交叉处，称为资料点(cell)。数据点式表格中用来键入数据数字或从数据库叫出数据数字的位置。至于说明文字，可在表格中之任意加入。

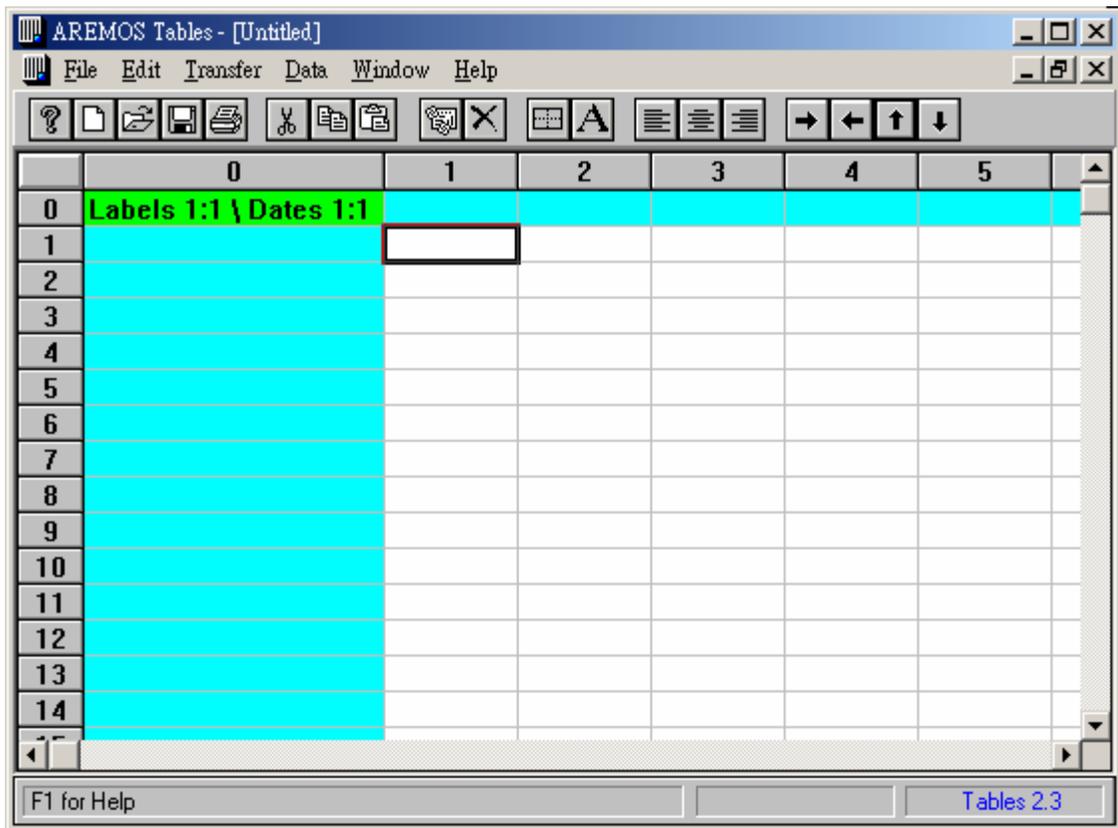


图 11.2

## 11.3 基本表格的设计

首先我们先练习基本表格的设计。先把鼠标移至您想要求东西的地方，按左键一下，即出现格子(及前所叙的数据点)。在此格子内，您即可输入您要的东西，当填完后，要到下一个格子，可将鼠标移至下一个格子，点一下，即可在输入你所要的东西。

主要的功能按键：

### **F7 : Edit Row(编辑横列)**

可清除资料，而此列还在 : **clear**

可删除数据，同时此列不见了 : **delete**

可插入资料，比以前多了一列 : **insert Before**

### **F8 : Edit Column(编辑纵行)**

可清除资料，而此行还在 : **clear**

可删除数据，同时此列不见了 : **delete**

可插入资料，比以前多了一列 : **insert Before**

### **F4 : Retrive 执行左边变量轴和右边时间轴所下的指令**

鼠标点取 **Edit → Cut** : 可清除此隔的所有东西，或按 **←** 键。

至于格子的大小，可用鼠标在 **Column1 & Column2** 中间，接着左拖曳左方(缩减格子)，或右方(增宽格子)，或在 **Row1 & Row2** 中间，接着左键拖曳上方(缩短格子)，或下方(增高格子)。其它的 **Column** 与 **Column** 之间，**Row** 与 **Row** 之间的放大缩小皆是如此。

若想要放大字体或要让字体置中排列，必须 **Mark** 您要的范围，(接着鼠标左键拖曳到您要的区域，然后放开左键)然后按 **AREMOS Tables** 里的工具列 **A**，进去改一改设定，即可放大。或按工具列的置中排列，及把整个 **Mark** 的地方，所有字体全部置中排列对齐。当然若您想整列(**row**)或整行(**column**)**Mark** 起来，只要用鼠标在整列的开

头(1,2,3 ...)处点一下，即整列都 Mark 起来。整行 Mark 也是如此。

## 11.4 数据取得和设定时间

数据可以由按键逐笔输入或从数据库中直接叫进来。若从数据库中叫进数据，须先开启数据库；亦即进入表格画面前，先 open 数据库。

AREMOS/Windows 允许的数学运算是可在表格变量轴中，用来定义变量的数据。位于屏幕上方(时间轴)之反白空格，系用来定义纵行数据时间。此反白时间空格共有 50 个，只作为定义变量之时间，不会随表格打印出。

(1) 设定时间的方式，是以英文前缀代表数据之频率，如下所示：

年	YYYY	或 YY	如：1995、95
季	YYYYQN	或 YYQN	如：1995Q1、95Q1
月	YYYYMN	或 YYMN	如：1995M1、95M1
周	YYYYWN	或 YYWN	如：1995W1、95W1
日	YYYY:MM:DD	或 YY:MM:DD	如：1994:12:20、 95:03:10。(2004:3:6、104:3:6)
	YYYYDN	或 YYDN	如：1995D10

注：2000 年以后可简用 100、101、102...(如 104Q3、104:3)

连续的时间可以简化用 + 或 - 号表示，如：

85Q1 + + + 产生 85Q1 85Q2 85Q3 85Q4  
— — — 86Q4 产生 86Q1 86Q2 86Q3 86Q4

不连续的时间可以简化用 +n 或 -n 表示，如：

83A1 +2 +4 产生 83A1 85A1 87A1  
-5 -3 82A1 产生 77A1 79A1 82A1

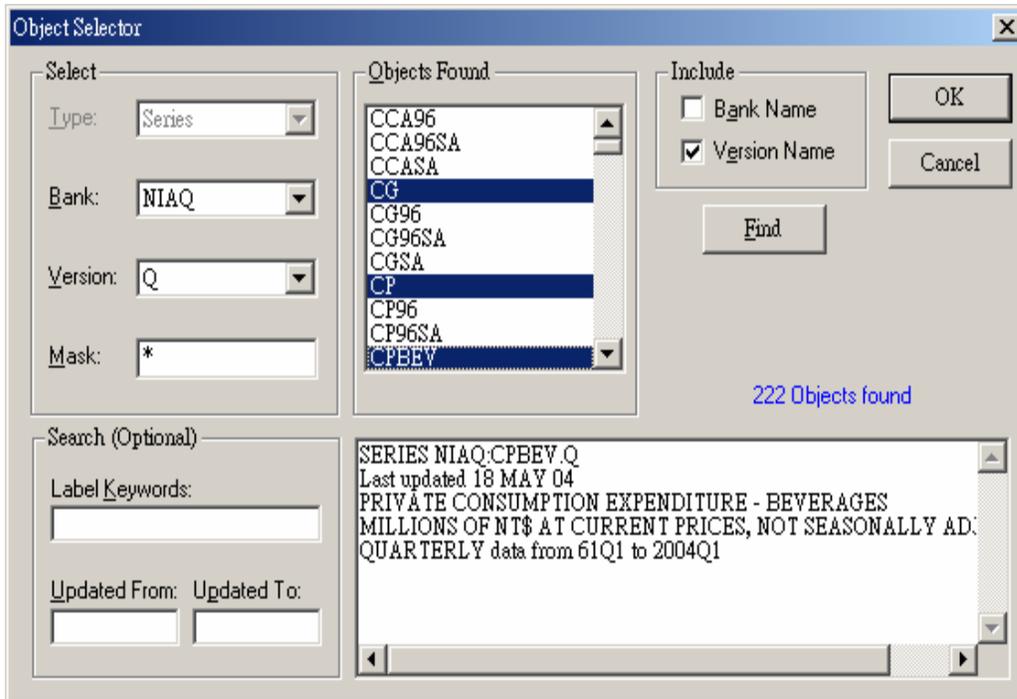
以上所提的时间设定，式屏幕正上方的时间轴设定。此时间轴 (Date 1:1) 设定不随表格印出，仅用来定义纵行数据之时间。如要使数据时间出现在表格区成为纵行数据之标题(headings)，只要用鼠标，在您想要设定的地方按下后，直接 key-in 标题即可，且可用鼠标在最上的 Column 1 和 Column 2 间，去拖曳格子的大小宽度(或 Column 2 和 Column 3 中间，以此类推，各取所需)。下一节，图 11.5 的编辑报表即是范例。

(2)因为在进入 Table(表格)画面前，已先在主画面 AREMOS Command 下 open 了数据库(niaq)，且之后在进入 Table 画面后，先在时间轴上(Dates 1:1)设定了时间(1988:4~1989:3)，然后下一步骤，就是叫取数据来制表啰！用鼠标操作如下：

Transfer → Get Series Name → 在 Bank: 处选 niaq(在 Select Series Objects 画面下) → 选 Q (在 Version 处)  
用 Ctrl-鼠标  
→ 选 cg,cp,cpbev → OK →按 F4 执行。(如图 11.4.2)

(3)用工具列上的 printer 印出 output 即可。但记住，此打印出的画面只有空白里所出现的文字或数字。而上方的时间轴及左边的变量轴皆不会印出。

※注：图 11.4.2 就是一个范例 (但别忘了之前要 open 您的数据库)，最后要按 F4 键(retrieve)执行。



AREMOS Tables - [Untitled]

File Edit Transfer Data Window Help

	0	1	2	3	4	5	6
0	Labels 1:1 \ Dates 1:1	-	1989q1	+	+		
1	CG.Q	131407	162051	144119	154235		
2	CP.Q	462484	528462	477384	558658		
3	CPBEV.Q	17559	16140	18970	22386		
4							
5							
6							
7							
8							
9							
10							
11							
12							

F1 for Help

Tables 2.3

图 11.4.2

## 11.5 编辑报表及数学计算公式

上一节的数据取得和设定时间，是最基本的拿数据制表的方法。但读者应之，当图 11.4.2 打印出 **output** 时，只有 **data** 显示，换句话说，只有白色空格里的东西才打印出来，而时间轴(**Date 1:1**)及变数轴(**Label 1:1**)里的符号皆不打印。所以您必须在此 **Table** 上，必须编辑文字，叙述变量轴上的每一笔项目(**objects**)即时间轴每一笔时间代表的意思。

在图 11.5，是个编辑报表，也是个范例，此 **AREMOS/Windows** 的制表(**Table**)功能很类似 **Microsoft** 里的 **Excel** 软件(电子表格)。不但可以调整格小长宽(用鼠标在 **column** 与 **column** 处,及 **row** 与 **row** 处),且可画格子(用工具列里的格子图样),不过记住,要先 **Mark** 范围。

至于特殊的数学计算公式，我们以下即将一一介绍。位于屏幕左边之变量轴，除可键入单个变量名字外，亦允许输入数学表达式。记住执行(**retrieve**)要按 **F4** 键。

**=Sn1..n2**      加总第 **n1** 列至第 **n2** 列资料。如 **gdp**、**cp**、**cpbev** 分别位于第 2、3、4 列，若在某列变数轴键入 **=S2..4**，可得该四列之加总。

**=Rn**      在某列变量轴键入 **=R4\*R6**，可以得到第 4 列与第 6 列相乘之结果。若 **=R4+R6**，可得到相加结果。若 **=R4/R6**，则可得相除结果。

**=Cn**      在某行时间轴键入 **=C5-C8**，可以得到第 5 行与第 8 行相减之结果。

**=n** 整列产生相同之 **n** 值。

**=p(Rn,Cm,x)** 等比(**projection**)公式。以第 **n** 列变量轴与第 **m** 行时间轴交叉之数据点为起始值，随时间轴之时间改变逐一乘以 **x** 值。

**=p(Rn,Cm,Rx)** 等比(**projection**)公式。以第 **n** 列变量轴与第 **m** 行时间轴交叉之数据点为起始值，随时间轴之时间改变逐一乘以第 **x** 列之数值。

**=i(Rn,Cm,x)** 等加(**increment**)公式。以第 **n** 列变量轴与第 **m** 行时间轴交叉之数据点为起始值，随时间轴之时间改变逐一等加 **x** 值。

**=i(Rn,Cm,Rx)** 等加(**increment**)公式。以第 **n** 列变量轴与第 **m** 行时间轴交叉之数据点为起始值，随时间轴之时间改变逐一等加第 **x** 列之数值。

	0	1	2	3	4	5
0	Labels 1:1 \ Dates 1:1		2003q1	+	+	+
1		項目/時間				
2	gdp	國內生產毛額	2479703	2275903	2518212	2573737
3	cp	私人消費	1640674	1402115	1618901	1524864
4	cpbev	私人消費, 飲料	38768	42494	51026	42671
5						
6						
7	=r2+r3+r4	列和[2]+[3]+[4]	4159145	3720512	4188139	4141272
8	=s2..7	列和from [2] to [7]	8318290	7441024	8376278	8282544
9	=3	列皆為3	3	3	3	3
10	=r4*r9*4/3	列[4]乘列[9]乘4除3	155072	169976	204104	170684
11						
12						

图 11.5

※注(1): 图 11.5 就是一个编辑报表的范例, 而且左边的变量轴也使用了数学计算公式(但别忘了之前要 **open** 您的数据库), 最后要按 **F4** 键(retrieve)执行。

注(2): 当 Table 的 **output** 已形成, 可自己储存此表格, 扩展名是 (.tbl), 以方便下次叫出来修改。

注(3): 在修改完表格上的时间序列数据, 要存回数据库(work bank)必须特别小心。因为会盖过原来数据库的数据。比如说, 你建了如图 11.5 的表格, 您储存的文件名叫 **TEST2.TBL**, 则在你的 **work bank** 里, 也就自动形成了 **gdp.q**、**cp.q**、**cpbev.q**、的 **series**, 时间由 **2003q1 to 2003q4**; 如果你在 **TEST2.TBL** 这个档案中修改了 **series(objects)**的 **data**, 然后 **save** 在同样的 **TEST2.TBL** 档, 则 **work bank** 里的 **sereis(objects)**的值就也跟着被修改过。所以尽可能用数据库里的数据来制表, 最好不要再修改数据内容。

## 第十二章 AREMOS/Windows Graphic 绘图简介

AREMOS/Windows 绘图制作，几乎完全是采屏幕上 Windows 画面交谈式来做出您所要的图形。非常简便，而且您可透过直接从数据库里叫取到的数据，直接拿来绘图。操作非常容易，几乎完全只用鼠标按键，点出您要的设定，迅速且快速绘图成功。AREMOS/Windows 只要用鼠标点取上面字段的选择项，即很容易马上了解此选择项的功能，这也是 Windows 窗口选想最受目前一般使用 PC 的人所欢迎。

### 12.1 绘图步骤

首先您还是必须在主画面上，去设定 frequency 频率即 period 期间，再 open 数据库。或许你的数据需要 collapse (conv) 转换低频率，在主画面上的 AREMOS Command 下指令，转换成功后，再进入 Graph 绘图的环境下去绘图。以下为绘图的基本操作，请用鼠标叫取。范例：

- (1) Options → Set → Period → (Frequency:Quarterly, From 1999:1 to 2003:4 → OK
- (2) Data → Bank → Open → Files → 磁盘驱动器(看看自己的叫取数据库放在那一槽) → niaq.bnk → 确定 → Open → Cancel 跳回主画面
- (3) Tools → Graph → Format → Data... → 键入指令 cp (在 Series Name or Expression 处) → insert → 键入指令 gnp → OK 跳到 Graph 画面
- (4) Fornat → Data... → Plot Style → 选 Bar (在 type 处) →

**OK**

**(5)Format → Axes → Bottom X Axes → Title → 键入指令  
Time (1999:1 to 2003:4)(在 Caption 处) → OK**

**(6)Format → Axes → Left Y Axes → Title → 键入指令  
UNIX : \$NT Million (在 Caption 处) → OK**

**(7)Format → Titles → Main Title → 键入指令 Gross  
National Product & Private Consumption(在 Caption 处) →  
OK**

**(8)File → Print → 确定**

**(9)File → Save as → 键入指令 test1.gra → 确定**

**(10)File → Exit**

说明：

**(1)(2)：**请参考前面范例。

**(3)：**叫取 **cp** 及 **gnp** 两笔 **time series** 资料，按 **ok**，马上就显现出最基本的直线形的画面，如图 12.1.3。(因为我们还没有开始设定图形的种类)

**(4)：**这是在设定图形的种类，然后画出长条形(**bar**)的二度空间图形，如图 12.1.4。

**(5)(6)(7)：**这是对图形 **X** 轴，**Y** 轴及 **Title** 加以注明写其代表的意思。一般来说，**X** 轴代表 **Time Period** 时间区间的意思，**Y** 代表数量单位，而 **Title** 是注明此两笔 **objects(time series)** 的名称全名。

**(8)：**从打印机打印出。

**(9)：**存档。通常扩展名取(**.gra**)，以方便下此叫取。

**(10)：**跳回主画面。

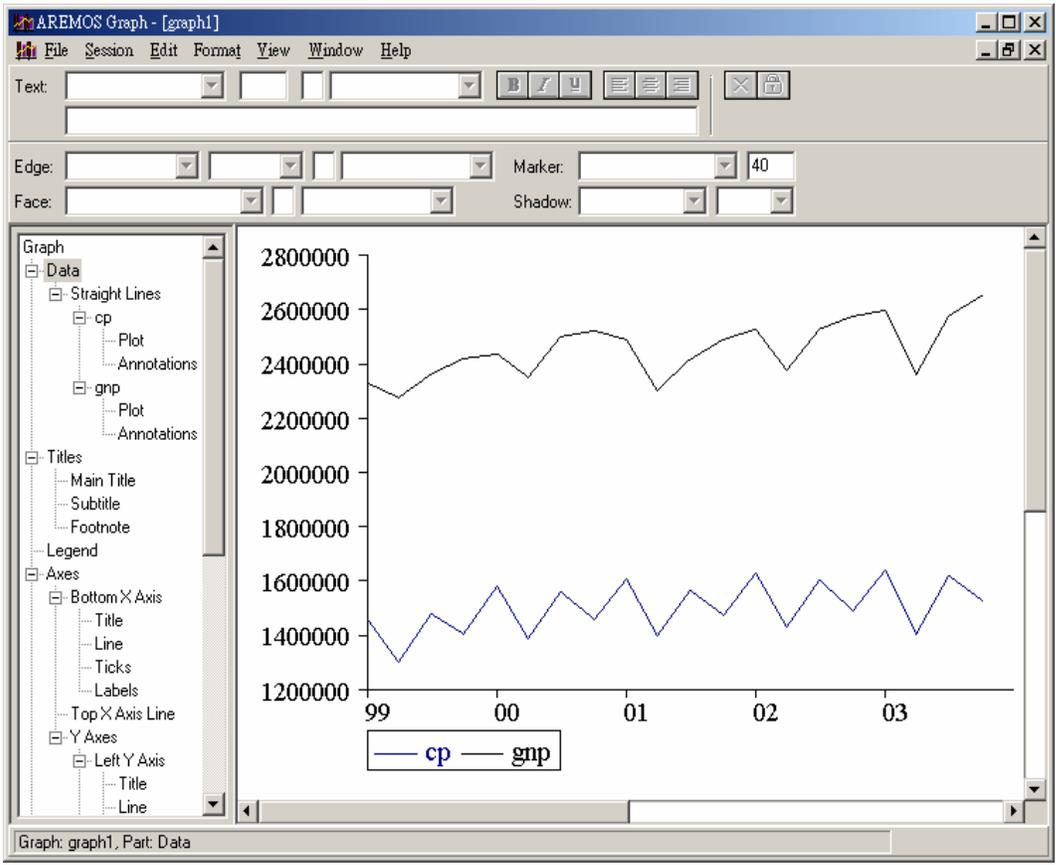
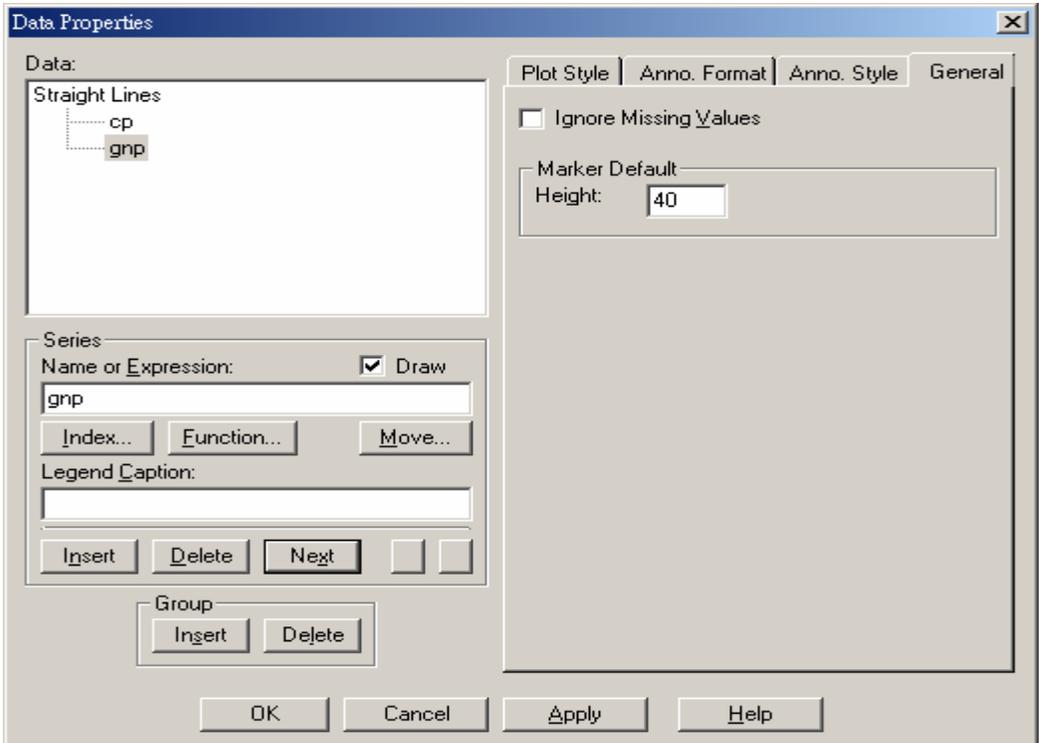


图 12.1.3

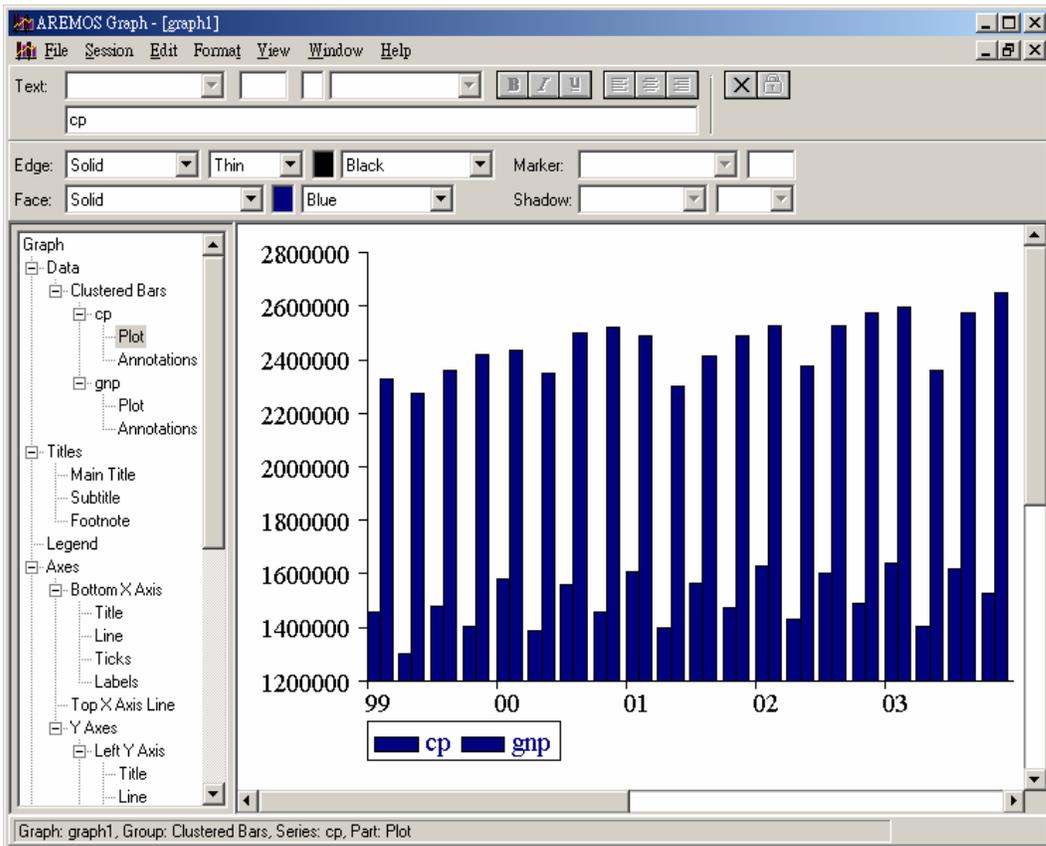
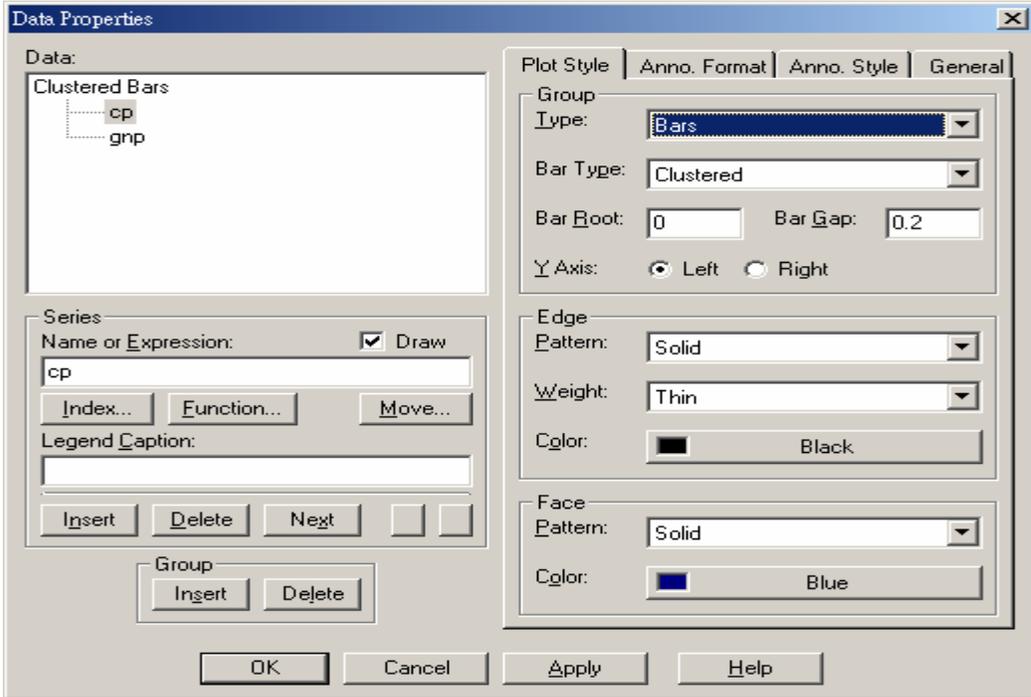


图 12.1.4

## 12.2 AREMOS Graph 「Help」

图形种类的设定，在 AREMOS Graph 的画面下，有多种选择，尤其是 **Format** 字段底下的东西，有 **Graph, Data Area, Axes, Titles, Fills, Legend, Bounding Box** 可进行设定你所想表示的图形，本入门讲义上一节 12.1 是最基本的绘图方法；假若你有兴趣设计你的独创式图形，却不知如何了解 **Format** 里的工具种类的意思，请参考 AREMOS Graph 「Help」里的操作方法。

例：(1) 请用鼠标点取 AREMOS Graph 画面上的「Help」地方。

(2) 多利用「F1」功能键。当鼠标停留在某画面，按「F1」键盘功能键，即出现原文的操作使用说明书。

## 第三部分

### 循环及计量模型

## 第十三章 循环的使用 ( for-end )

AREMOS/Windows 提供我们可对一个 list 里的每一个 item 进行同样的「运算操作」。由于 list 可含上百或最多至 10,000 个 items，因此若有循环操作的功能，可节省我们不少时间。有关循环的使用，可参考原文手册「AREMOS/Windows Reference Manual」p. 95~97 之说明。本章仅以三个范例来介绍其最简单之用法。以下皆是在主画面 AREMOS Command 下键入指令。

### 13.1 循环之使用例(甲)

- (1) 键入指令 `open e:\waremos\niaq ;`
- (2) 键入指令 `list n1=gnp ,cp,cg ;`
- (3) 键入指令 `set freq q ; set per 1990q1 2003q4`
- (4) 键入指令 `for aaa=#n1 ;`
- (5) 键入指令 `print pchya(#aaa) ;`
- (6) 键入指令 `analyze #aaa ;`
- (7) 键入指令 `end ;`

说明：

(4)：「for aaa=#n1」表示要对 list n 进行循环运作。aaa 是随便给代号。从此行起即开始循环的运作，直到第 (7)行 end 为止。for-end 之间即是循环的「运作指令群」。

(5)(6)：此二行在 for 与 end 之间，是循环的「运作指令群」。基本上，「aaa」是一代号 (是一虚拟代号，并不存在)。计算机会把 n1 里的每一个 item 当做「aaa」而依此「运作指令群」执行一次。

(7): 输入 **end** 以后, 计算机才会开始执行(5)(6)之指令群。

## 13.2 循环之使用例(乙)

假定我们要从数据库叫取大堆数据, 取出后并加以改名, 而且是定期(每月或每季)常常要进行此种取改动作。此时以 **for-end** 来做就简便多了。下面范例是把 **aa list** 所代表之检索代号从 **niaq** 数据库 **copy** 到 **primary bank**, 同时把名称改成 **bb list** 所代表之名字。

(1) 键入指令 **set freq q ; set per 1985:1 1994:4;**

(2) 键入指令 **open e:\waremos\niaq;**

(3) 键入指令 **for aa=gnp,cp,cg bb=x1,x2,x3;**

(4) 键入指令 **copy #aa as #bb;**

(5) 键入指令 **print #aa,#bb;**

(6) 键入指令 **end;**

说明 ;

(3): 前面我们采用预先设定 **list** 的内容为何。在此, 我们也可采用直接在 **for listname=**后面把 **items** 的名称一一输入。**for aa=.... bb=....;** 表示要对两个 **list aa** 与 **bb** 进行循环运作, 以空一格把 **aa** 与 **bb** 连接起来。**For** 是宣称开始循环运作, 直到第(6)**end** 出现为止。**for-end** 之间即是循环的「运作指令群」。

(4)(5): 此二指令为循环的「运作指令」。「**aa**」与「**bb**」为虚拟代号。计算机会把 **aa list** 里的每一个 **item** 当做 **aa**, 把 **bb list** 里的每一个 **item** 当做 **bb**, 而依此「运作指令」执行一次。

(6): 输入 **end** 以后, 计算机才会开始执行循环的运作指令。

## 13.2 循环之使用例(丙)

在第四章范例(五), 我们要叫取跨数据库之时间序列时, 必须另外建

立一个新的变量来将放在不同数据库的同一笔数据加以结合。

若所较取数据比数不多，则可土法炼钢式的如范例(五)来操作。但新叫取的数据有几十笔、甚至上百笔时，则可以循环来节省时间。此将范例(五)的指令改为如下：

- (1) 键入指令 `set freq d; set per 1989:12:1 1990:1:12;`
- (2) 键入指令 `open e:\waremos\stock80,e:\waremos\stock90;`
- (3) 键入指令 `list n1=s1206@cl,s2805@cl;`
- (4) 键入指令 `for aaa=#n1;`
- (5) 键入指令 `series<1989:12:1 1989:12:31>new|#aaa=stock80:#aaa;`
- (6) 键入指令 `series<1990:1:1 1990:1:20>new|#aaa=stock90:#aaa;`
- (7) 键入指令 `end;`
- (8) 键入指令 `list n2=#n1 prefix=new;`
- (9) 键入指令 `print #n2;`

说明：

(3)：建立一个 list 叫 n1，n1 代表着 s1206@cl、s2805@cl 两个变量名称。

(5)：此指令乃透过循环的执行形成一笔新的变量。新变量名称即在 aaa 所代表的变量名称前面加上 new 字眼。如当 aaa=s126@cl 时，此指令即在 primary bank 形成一笔数据叫 news1206@cl。

(6)：见(5)之说明。

(8)：建立一个 list 名叫 n2。此指令相当于 list n2=news1206@cl,news2805@cl; 即在 n1 所代表的各个变量名称前加上一个字符串 new。我们也可将此字母加于变量之后，指令则是将 prefix 改为 suffix 即可。

如：

```
list n2=#n1 suffix=new;
```

相当于：

```
list n2 =s1206@clnew,s2805@clnew;
```

※ 请注意：在 open 数据库时，「键入指令 **open niaq;**」，意思事 open c:\WAREM32 子目录里的 **niaq.bnk**；若您的数据库 **niaq.bnk** 是放在 e 槽的 WAREMOS 子目录时，则您必须「键入指令 **open e:\WAREMOS\niaq;**」。

※ 补充：转换频率的循环设定，在 AREMOS Command 下

- (1) 键入指令 **set freq w; set per 1990:1 2003:47;**
- (2) 键入指令 **open e:\waremos\tse.bnk;**
- (3) 键入指令 **list n1=jhk,jbangkok,js;**
- (4) 键入指令 **for x=#n1;**
- (5) 键入指令 **collapse<tolerate=99>#x|.w=#x|.d last;**
- (6) 键入指令 **end;**
- (7) 键入指令 **print #n1;**

## 第十四章 计量模型之建立与求解

本章介绍 AREMOS/Windows 计量模型之做法，已一简单实例，按下列小节依序说明：

- 基本概念。
- 模型建立。
- 样本外预测。
- 外生变数仿真。
- 加入 **Addfactors**。
- 产生 **Implicit Addfactors**。
- **Large Model** 解法。
- 结果陈示。
- 指令摘要。

### 14.1 基本概念

用 AREMOS/Windows 建立模型，从单一方程式到包罗上百条以上行为方程式的复杂联立模型均可处理。

AREMOS/Windows 求解联立模型时，会自动进行下列过程：

- 利用 **Tarjan ordering algorithm** 法，排列方程式。
- 利用 **Gauss—Seidel** 法求解模型。
- 产生模型解之数列。

值得注意的是，因求解过程采用 **Gauss—Seidel** 法，模型内生变量 (**endogenous variable**) 应置于方程式等号左边，而且同一变量不能在其他方程式左边重复出现。

AREMOS/Windows 对联立模型求解，分为「标准模型」

(standard)与「大模型」(large model)两种方式。大模型顾名思义应为方程式多、数据量大且复杂程度高之联立模型，但如何划分并无严格标准。二者求解过程略有差异，为对实际作业影响不大。

本章先介绍标准模型之解法，至于大模型解法，留待 14.7 节说明。

AREMOS/Windows 的求解指令只要为 model 及 solve 两个，此二指令均有许多设定条件(options)可供选择，可用 query 指令查看或 set 指令改变，在本章第 14.9 节说明。

为便于说明，以下设定简单模型，由四条方程式组成，其中行为方程式三条，定义式及恒等式一条。方程式设定如下：

$\begin{aligned} \text{cp96} &= f_1(\text{gdp96}, \text{cp96}_{.1}) \\ \text{ifix96} &= f_2((\text{cp96} + \text{cg96} + \text{ifix96} + \text{invch96}), \text{ex96}) \\ \text{m96} &= f_3(\text{gdp96}, \text{m96}_{.1}) \\ \text{gdp96} &= \text{cp96} + \text{cg96} + \text{ifix96} + \text{invch96} + \text{ex96} - \text{m96} \end{aligned}$
--

上列方程式中，其变数皆取实质(基期调整后)，cp96=民间消费，cg96=政府投资，ifix96=固定资本形成，invch96=存货增加，ex96=商品及劳务输出，m96=商品及劳务输入，gdp96=国内生产毛额，cp96.1、m96.1 为前期变数。显而易见，本模型之内生变量为 cp96、ifix96、m96、及 gdp96,而外生变数为 cg96、invch96 及 ex96。

## 14.2 模型建立

### 14.2.1 估计方程式

建立模型的第一步工作为估计方程式。指令如下：

- (1) 键入指令 `open<primary>rocmd`; 新开启模型库
- (2) 键入指令 `clear rocmd`; 清除模型库内对象
- (3) 键入指令 `set freq a`; 数据频率
- (4) 键入指令 `set per 1970 2004`; 资料期间
- (5) 键入指令 `open e:\waremos\niaa`;
- (6) 键入指令 `equ<autofit no>cp96`  
 $\log(\text{cp96})=\log(\text{gdp96}),\log(\text{cp96.1});$
- (7) 键入指令 `normalize cp96=exp(??)`;
- (8) 键入指令 `fit`;
- (9) 键入指令 `equ<auto no>ifix96`  
 $\log(\text{ifix96})=\log(\text{cp96}+\text{cg96}+\text{ifix96}+\text{invch96}),\log(\text{ex96});$
- (10) 键入指令 `normalize ifix96=exp(??)`;
- (11) 键入指令 `fit`;
- (12) 键入指令 `equ<autofit no>m96`  
 $\log(\text{m96})=\log(\text{gdp96}),\log(\text{m96.1});$
- (13) 键入指令 `normalize m96=exp(??)`;
- (14) 键入指令 `fit`;
- (15) 键入指令 `equ<autofit no constant`  
 $\text{no}>\text{gdp96}=\text{cp96}+\text{cg96}+\text{ifix96}+\text{invch96}+\text{ex96}-\text{m96};$
- (16) 键入指令 `fit`;

第一条至第三条的估计结果如下：

```

AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
Primary: rocmd1
=> NORMALIZE CP96=EXP{??}

=> FIT;
CP96
Ordinary Least Squares
ANNUAL data for 34 periods from 1970 to 2003
Date: 13 AUG 2004

log(cp96)

= 0.69338 * log(cp96)[-1] + 0.30338 * log(gdp96) - 0.07424
(12.1189) (5.03589) (0.69911)

Sum Sq 0.0119 Std Err 0.0196 LHS Mean 14.5503
R Sq 0.9994 R Bar Sq 0.9993 F 2, 31 24768.3
D.W.{ 1} 0.6690 D.W.{ 2} 1.2894
H 4.1057

CP96=EXP{??}

```

```

AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
Primary: rocmd1
=> FIT;
IFIX96
Ordinary Least Squares
ANNUAL data for 34 periods from 1970 to 2003
Date: 13 AUG 2004

log(ifix96)

= 1.22198 * log(cp96+cg96+ifix96+invch96) - 0.08956 * log(ex96)
(8.12932) (0.82668)

- 3.63160
(4.69871)

Sum Sq 0.2284 Std Err 0.0858 LHS Mean 13.5119
R Sq 0.9891 R Bar Sq 0.9884 F 2, 31 1403.42
D.W.{ 1} 0.4817 D.W.{ 2} 0.9311

IFIX96=EXP{??}

```

```

AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
Primary: rocmd1
=> NORMALIZE M96=EXP{??}

=> FIT;
M96
Ordinary Least Squares
ANNUAL data for 34 periods from 1970 to 2003
Date: 13 AUG 2004

log(m96)

= 0.44151 * log(m96)[-1] + 0.71222 * log(gdp96) - 2.86327
  (4.13558)          (4.90593)          (3.94470)

Sum Sq 0.1167 Std Err 0.0614 LHS Mean 14.0724
R Sq 0.9962 R Bar Sq 0.9960 F 2, 31 4082.97
D.W.(1) 1.5932 D.W.(2) 1.6641
H 1.5000

M96=EXP{??}

```

ANNUAL From: 1970A1 To: 2004A1

```

AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
Primary: rocmd1

M96=EXP{??}

=> EQUATION <CONSTANT=NO SEASONAL=NO AUTOFIT=NO VERSION=NULL> GDP96 " gdp96 = CP96+C

=> FIT;
GDP96
Ordinary Least Squares
ANNUAL data for 34 periods from 1970 to 2003
Date: 13 AUG 2004

gdp96

= 1.00000 * cp96+cg96+ifix96+invch96+ex96-m96
  ( 3E+09)

Sum Sq 0.0039 Std Err 0.0109 LHS Mean 4656864
R Sq 1.0000 R Bar Sq 1.0000 F 0, 33 NC
D.W.(1) 2.0111 D.W.(2) 2.0108

```

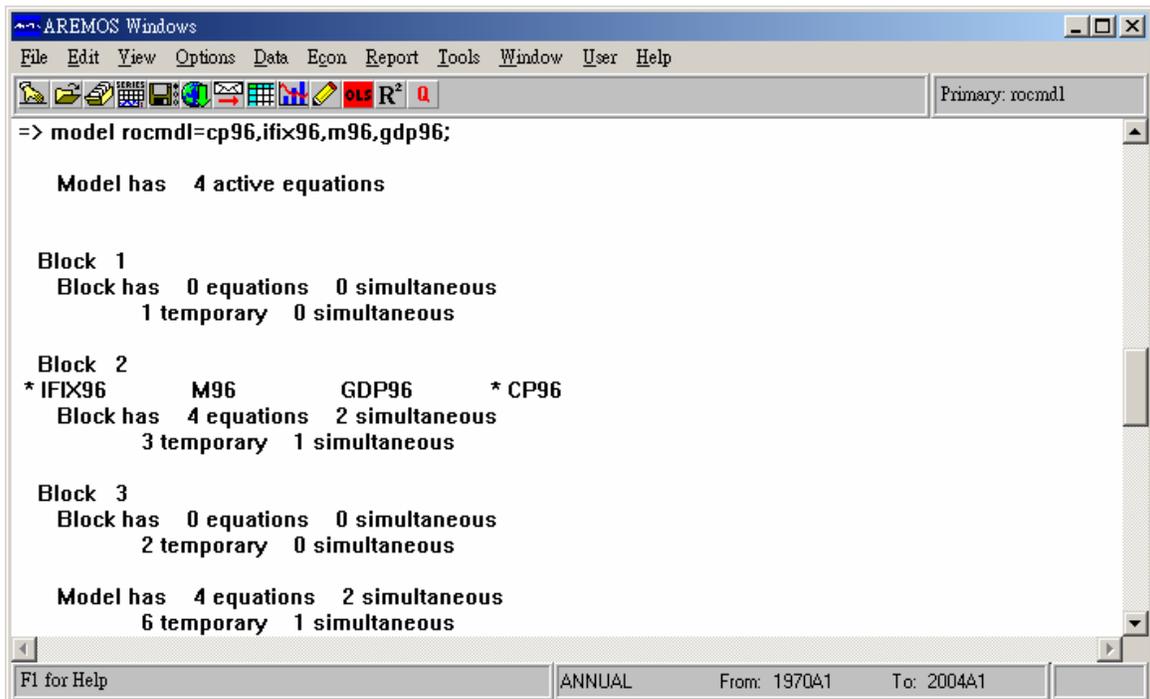
ANNUAL From: 1970A1 To: 2004A1

## 14.2.2 编译及求解

其次为编译(compile)及求解。指令如下：

- (1) 键入指令 `set compile order yes;`
- (2) 键入指令 `model rocmdl=cp96,ifix96,m96,gdp96;`
- (3) 键入指令 `set solve solution soll;`
- (4) 键入指令 `set per 1970 2004;`
- (5) 键入指令 `solve;`

上列指令中，`set compile order` 为开启 **Tarjan Ordering Algorithm** 排序方式(YES 可省略)。`Model` 指令后为对本模型命名，接着等号之后为所有方程式名称；如<primary>数据库中，除了本模型用到的方程式外，无其他方程序，亦可简写为”`model rocmdl=*.*`”。`set solve solution` 为预留位置，使所得之解以”变量.soll”名字陈现。上列指令之过程如下：



```
AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
=> model rocmdl=cp96,ifix96,m96,gdp96;

Model has 4 active equations

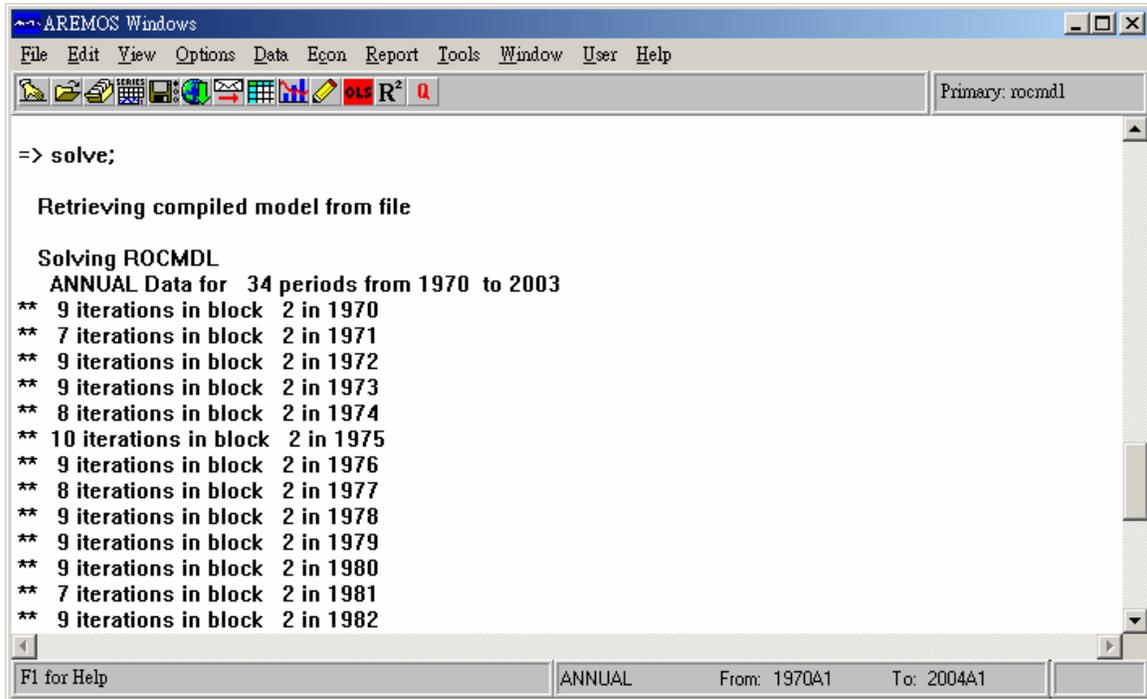
Block 1
Block has 0 equations 0 simultaneous
1 temporary 0 simultaneous

Block 2
* IFIX96      M96      GDP96      * CP96
Block has 4 equations 2 simultaneous
3 temporary 1 simultaneous

Block 3
Block has 0 equations 0 simultaneous
2 temporary 0 simultaneous

Model has 4 equations 2 simultaneous
6 temporary 1 simultaneous

F1 for Help ANNUAL From: 1970A1 To: 2004A1
```



如要查看求解结果，可用 **print** 指令叫出来。如：

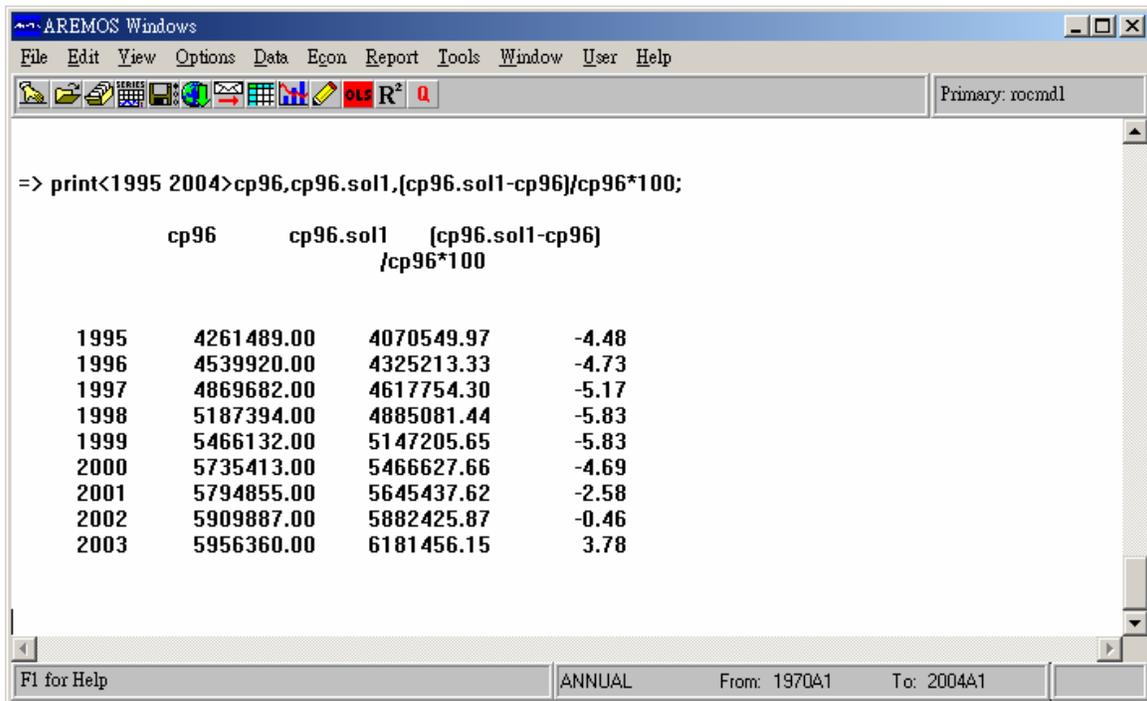
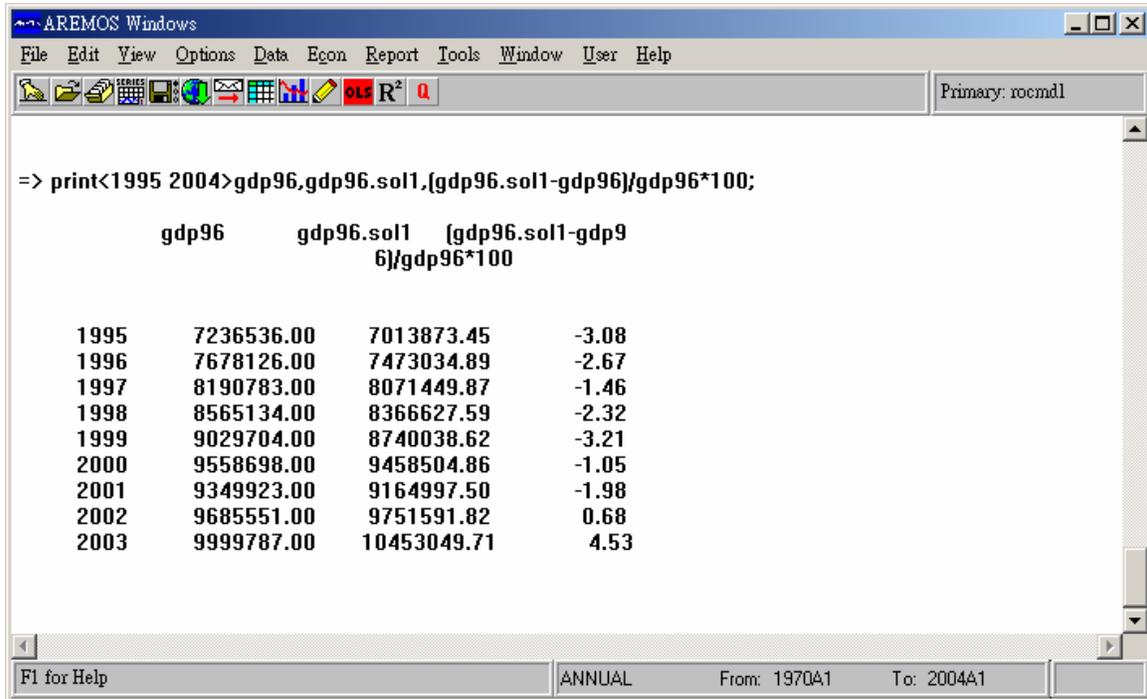
(1) 键入指令 **print<1995**

```
2004>gdp96,gdp96.sol1,(gdp96.sol1-gdp96)/gdp96*100;
```

(2) 键入指令 **print<1995**

```
2004>cp96,cp96.sol1,(cp96.sol1-cp96)/cp96*100;
```

结果如下：



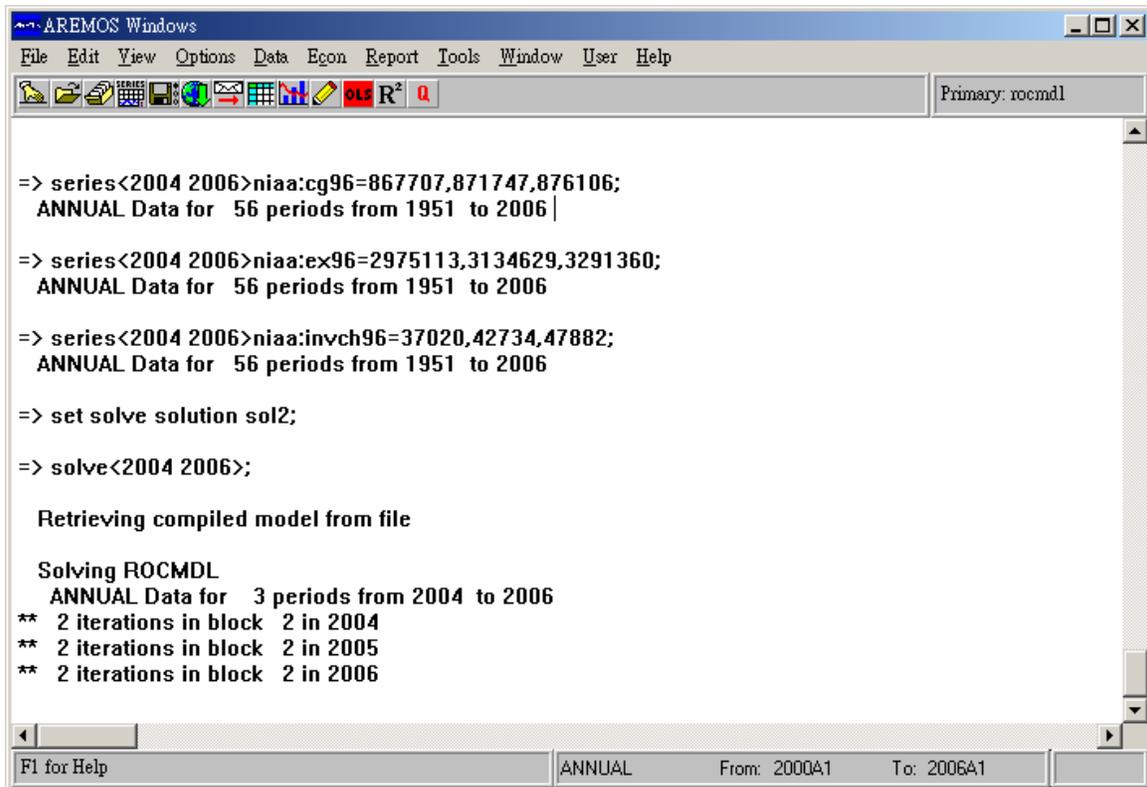
### 14.3 样本外预测

预测是假设样本期间外之外生变量，将其值代入模型，已求解内生变数之值。AREMOS 对样本外预测，作法与前小节几乎相同。只要外生变量数据时间够长，即可求得相同期间内生变量解。

如继续前小节模型，对 2004 至 2006 年做预测，首先输入该段时间外生变量:政府消费(**cg96**)、输出(**ex96**)及存货增加(**invch 96**)之值；其次改变 solution 的 versionname 为 **sol2**；最后为求解。求得之预测值会以”变量.sol2”名字出现。指令如下：

- (1) 把光盘的「niaa.bnk」copy 到 c:\warem32 子目录，且把「只读」取消，然后在 AREMOS 主画面下 open niaa;
- (2) 键入指令 **set per 2004 2006;**
- (3) 键入指令 **series<2004 2006>niaa:cg96=867707,871747,876106;**
- (4) 键入指令 **series<2004**  
**2006>niaa:ex96=2975113,3134629,3291360;**
- (5) 键入指令 **series<2004 2006>niaa:invch96=37020,42734,47882;**
- (6) 键入指令 **set solve solution sol2;**
- (7) 键入指令 **solve<2004 2006>;**
- (8) 键入指令 **print<2000 2006>gdp96,gdp96.sol1,gdp96.sol2;**
- (9) 键入指令 **print<2000 2006>cp96,cp96.sol1,cp96.sol2;**

详细过程如下：



	gdp96	gdp96.sol1	gdp96.sol2
2000	9558698.00	9458504.86	
2001	9349923.00	9164997.50	
2002	9685551.00	9751591.82	
2003	9999787.00	10453049.71	9999787.00
2004		7588321.00	
2005		7755296.58	
2006		7854081.59	

=> print<2000 2006>cp96,cp96.sol1,cp96.sol2;

	cp96	cp96.sol1	cp96.sol2
2000	5735413.00	5466627.66	
2001	5794855.00	5645437.62	
2002	5909887.00	5882425.87	
2003	5956360.00	6181456.15	5956360.00
2004		5657888.08	
2005		5495990.14	
2006		5407187.37	

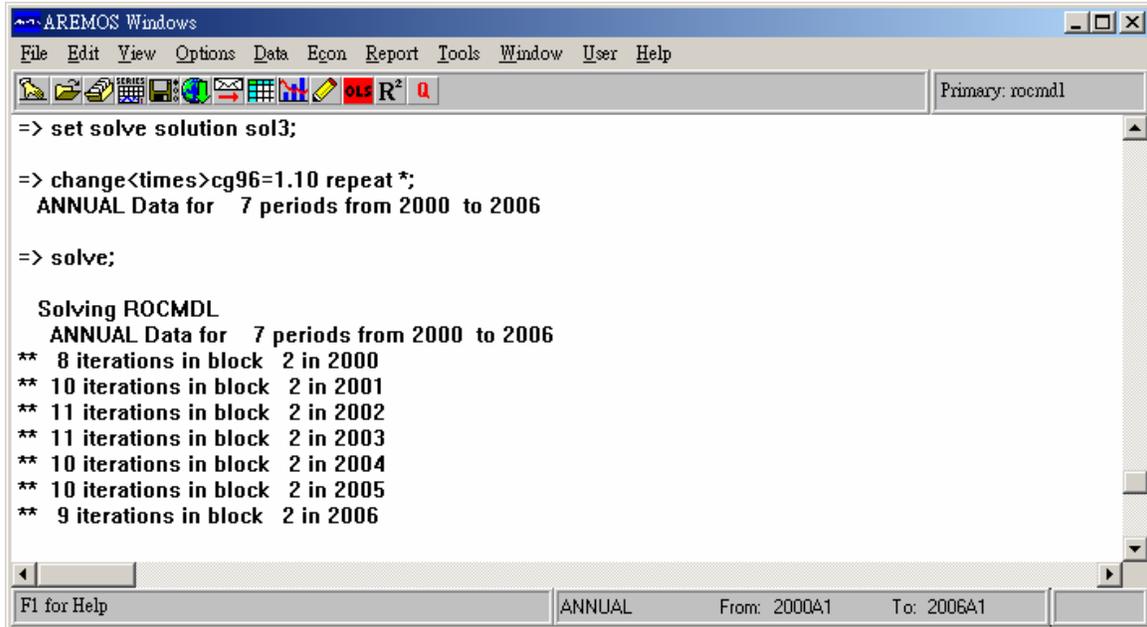
## 14.4 外生变数仿真

外生变数仿真是解答所谓“**What If**”问题。亦即及外生变量设定不同水平值下，内生变量产生多大变化。继续前例，如以预测期间的政府消费(**cg96**)增加 10%进行仿真，指令如下：

- (1) 键入指令 **set solve solution sol3;**
- (2) 键入指令 **change<times>cg96=1.10 repeat \*;**
- (3) 键入指令 **solve;**
- (4) 键入指令 **print<2000 2006>gdp96,gdp96.sol2,gdp96.sol3;**

(5) 键入指令 `print<2000 2006>cp96,cp96.sol2,cp96.sol3;`

详细过程如下：



```
AREMOS Windows
File Edit View Options Data Egon Report Tools Window User Help
Primary: rocmdl
=> set solve solution sol3;

=> change<times>cg96=1.10 repeat *;
  ANNUAL Data for 7 periods from 2000 to 2006

=> solve;

Solving ROCMDL
  ANNUAL Data for 7 periods from 2000 to 2006
** 8 iterations in block 2 in 2000
** 10 iterations in block 2 in 2001
** 11 iterations in block 2 in 2002
** 11 iterations in block 2 in 2003
** 10 iterations in block 2 in 2004
** 10 iterations in block 2 in 2005
** 9 iterations in block 2 in 2006

F1 for Help ANNUAL From: 2000A1 To: 2006A1
```

Primary: rocmdl

```
=> print<2000 2006>gdp96,gdp96.sol2,gdp96.sol3;
```

	gdp96	gdp96.sol2	gdp96.sol3
2000	9558698.00		9797695.53
2001	9349923.00		9486167.37
2002	9685551.00		10055653.55
2003	9999787.00	9999787.00	10743466.10
2004		7588490.46	7966258.18
2005		7755118.30	8108100.08
2006		7853930.32	8179546.24

```
=> print<2000 2006>cp96,cp96.sol2,cp96.sol3;
```

	cp96	cp96.sol2	cp96.sol3
2000	5735413.00		5760556.46
2001	5794855.00		5915604.34
2002	5909887.00		6133089.97
2003	5956360.00	5956360.00	6416055.06
2004		5657940.05	6045762.36
2005		5495958.13	5832775.65
2006		5407109.22	5704686.71

F1 for Help      ANNUAL      From: 2000A1      To: 2006A1

上述指令中，首先改变 solution 的 versionname。其次用 change 指令改变外生数值，使变量 cg96 的数值乘上 1.10(即增加 10%)，然后进行求解。所得之新预测值以”变量.sol3”名字呈现。

Change 指令有许多设定条件(options)可供选择，除 times 外，还有 replace、increase、decrease、%base、%change 等，内设为 replace，读者可参阅使用手册说明。

## 14.5 加入 Addfactors

预测时加入常数项调整(Constant Adjustment)，AREMOS 称为

**Addfactor** 。**Addfactor** 是针对行为方程式之被解释变数进行调整，亦即先外加(或减)某被解释变量一个数值，透过联立求解，使该被解释变量之预测值趋近目标值，并达到同时影响其它变量预测值的目的。

在运用 **Addfactor** 前，须对 **solve** 指令中 **options** 作设定，包括三项：

- (1) 键入指令 **set solve adjust yes;**
- (2) 键入指令 **set solve solution soladd;**
- (3) 键入指令 **set solve addfactor add;**

第一项为开启做 **Addfactor** 状态(yes 可省略)；第二项为预留位置，使求解之值以”变量.soladd”名字陈现；第三项为产生”被解释变量.add”之名字，以输入 **Addfactor** 之值。

以下指令针对模型中民间消费(**cg96**)变量，在预测时间加入

**Addfactor:**

- (1) 键入指令 **set per 2004 2006;**
- (2) 键入指令 **set solve adjust solution soladd addfactor add;**
- (3) 键入指令 **series cp96.add=72500,68000,64000;**
- (4) 键入指令 **solve;**
- (5) 键入指令 **print<2000 2006>cp96,cp96.sol2,cp96.soladd;**
- (6) 键入指令 **print<2000 2006>gdp96,gdp96.sol2,gdp96.soladd;**

详细过程如下：

```

AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
Primary: rocmdl

=> Set Period 2004 2006A1;

=> Set Frequency Annual;

=> series cp96.add=72500,68000,64000;
    ANNUAL Data for 3 periods from 2004 to 2006

=> solve;

Retrieving compiled model from file

Solving ROCMDL
    ANNUAL Data for 3 periods from 2004 to 2006
** 9 iterations in block 2 in 2004
** 9 iterations in block 2 in 2005

F1 for Help ANNUAL From: 2004A1 To: 2006A1

```

```

AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
Primary: rocmdl

cp96      cp96.sol2      cp96.soladd

2000      5735413.00
2001      5794855.00
2002      5909887.00
2003      5956360.00      5956360.00      5956360.00
2004      5657940.05      5752240.74
2005      5495958.13      5662663.85
2006      5407109.22      5628676.13

=> print<2000 2006>gdp96,gdp96.sol2,gdp96.soladd;

gdp96      gdp96.sol2      gdp96.soladd

2000      9558698.00
2001      9349923.00
2002      9685551.00
2003      9999787.00      9999787.00      9999787.00
2004      7588490.46      7685471.38
2005      7755118.30      7918705.97
2006      7853930.32      8063828.21

F1 for Help ANNUAL From: 2004A1 To: 2006A1

```

## 14.6 产生 Implicit Addfactors

在样本期间内，AREMOS 具有计算隐含(Implicit) Addfactors 之功能。意即在使样本其得到之解与实际值相同情况下，对应求得 Addfactors 值。这种反向过程，AREMOS 称为 Residual Feedback。其步骤为：

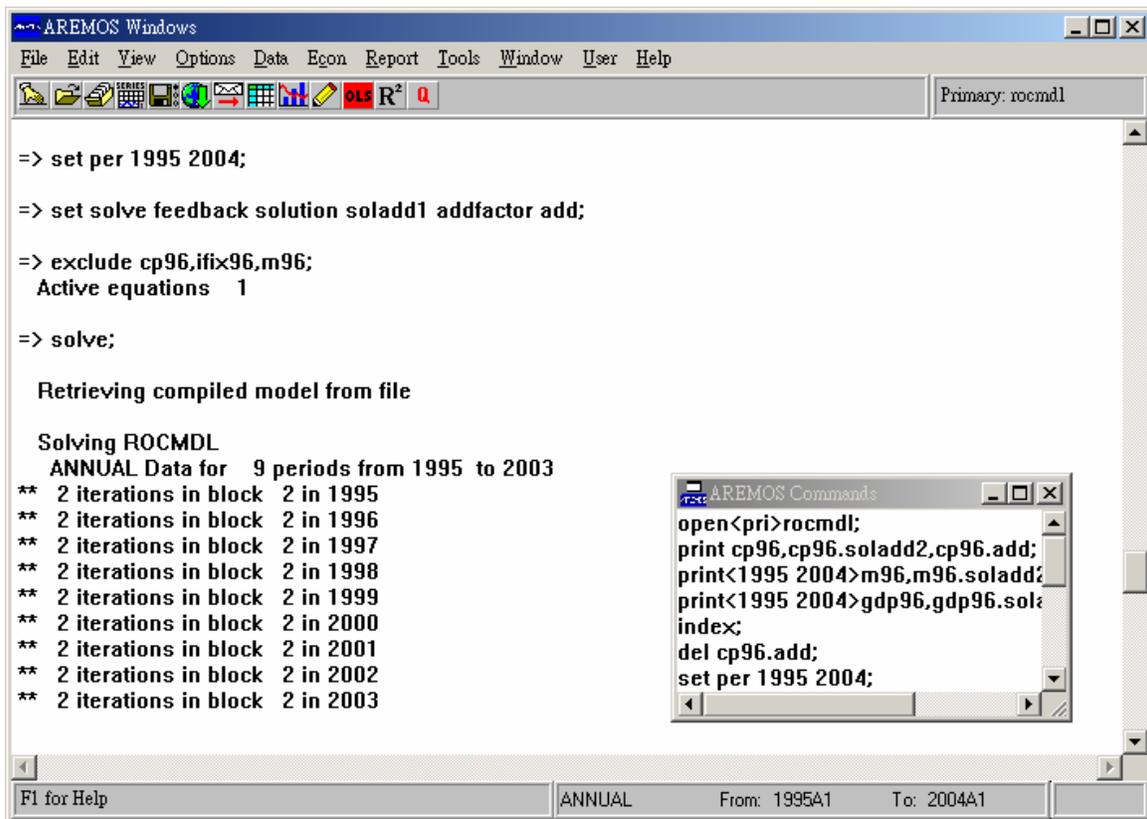
- (1) 键入指令 `set solve feedback yes;`
- (2) 键入指令 `set solve solution...;`
- (3) 键入指令 `set solve addfactor;`
- (4) 键入指令 `exclude` 行为方程式(意即被解释变量);
- (5) 键入指令 `solve;`
- (6) 键入指令 `set solve feedback no;`
- (7) 键入指令 `include` 行为方程式(意即解释变量);
- (8) 键入指令 `set solve solution...;`
- (9) 键入指令 重新 `solve;`

前五步骤产生隐含 Addfactors; 后四步骤为将隐含 Addfactors 代回模型，重新求解，以显示该隐含 Addfactors 却能使样本期得到之解与实际值一样。此处隐含 Addfactors 代回过程，亦可参采上小节纳入 Addfactors 之作法，结果完全一样。

以下接续上小节指令，以求取行为方程式 `cp96`、`ifix96`、`m96` 之隐含 Addfactors，指令如下：

- (1) 键入指令 `del cp96.add;`
- (2) 键入指令 `set per 1995 2004;`
- (3) 键入指令 `set solve feedback solution soladd1 addfactor add;`
- (4) 键入指令 `exclude cp96,ifix96,m96;`

- (5) 键入指令 `solve;`
- (6) 键入指令 `set solve feedback no;`
- (7) 键入指令 `include cp96,ifix96,m96;`
- (8) 键入指令 `set solve solution soladd2;`
- (9) 键入指令 `solve;`
- (10) 键入指令 `print cp96,cp96.soladd2,cp96.add;`
- (11) 键入指令 `print<1995 2004>m96,m96.soladd2,m96.add;`
- (12) 键入指令 `print<1995 2004>gdp96,gdp96.soladd2;`



```

AREMOS Windows
File Edit View Options Data Egon Report Tools Window User Help
Primary: rocmdl

=> set solve feedback no;

=> include cp96,ifix96,m96;
Active equations 4

=> set solve solution soladd2;

=> solve;

Solving ROCMDL
ANNUAL Data for 9 periods from 1995 to 2003
** 9 iterations in block 2 in 1995
** 9 iterations in block 2 in 1996
** 9 iterations in block 2 in 1997
** 10 iterations in block 2 in 1998
** 10 iterations in block 2 in 1999
** 10 iterations in block 2 in 2000
** 8 iterations in block 2 in 2001
** 10 iterations in block 2 in 2002
** 11 iterations in block 2 in 2003

=> print cp96,cp96.soladd2,cp96.add;

F1 for Help ANNUAL From: 1995A1 To: 2004A1

```

```

AREMOS Windows
File Edit View Options Data Egon Report Tools Window User Help
Primary: rocmdl

=> print cp96,cp96.soladd2,cp96.add;

      cp96      cp96.soladd2      cp96.add

1995      4261489.00      4261436.64      4710.27
1996      4539920.00      4539836.48      38226.22
1997      4869682.00      4869533.56      72893.70
1998      5187394.00      5187219.90      82888.16
1999      5466132.00      5465926.65      46806.52
2000      5735413.00      5735172.07      17896.99
2001      5794855.00      5794739.33      -77044.53
2002      5909887.00      5909841.79      -67757.94
2003      5956360.00      5956369.78      -162295.06
2004              5872133.69

=> print<1995 2004>m96,m96.soladd2,m96.add;

      m96      m96.soladd2      m96.add

1995      3188605.00      3188468.08      59074.34

F1 for Help ANNUAL From: 1995A1 To: 2004A1

```

## 14.7 Large Model 解法

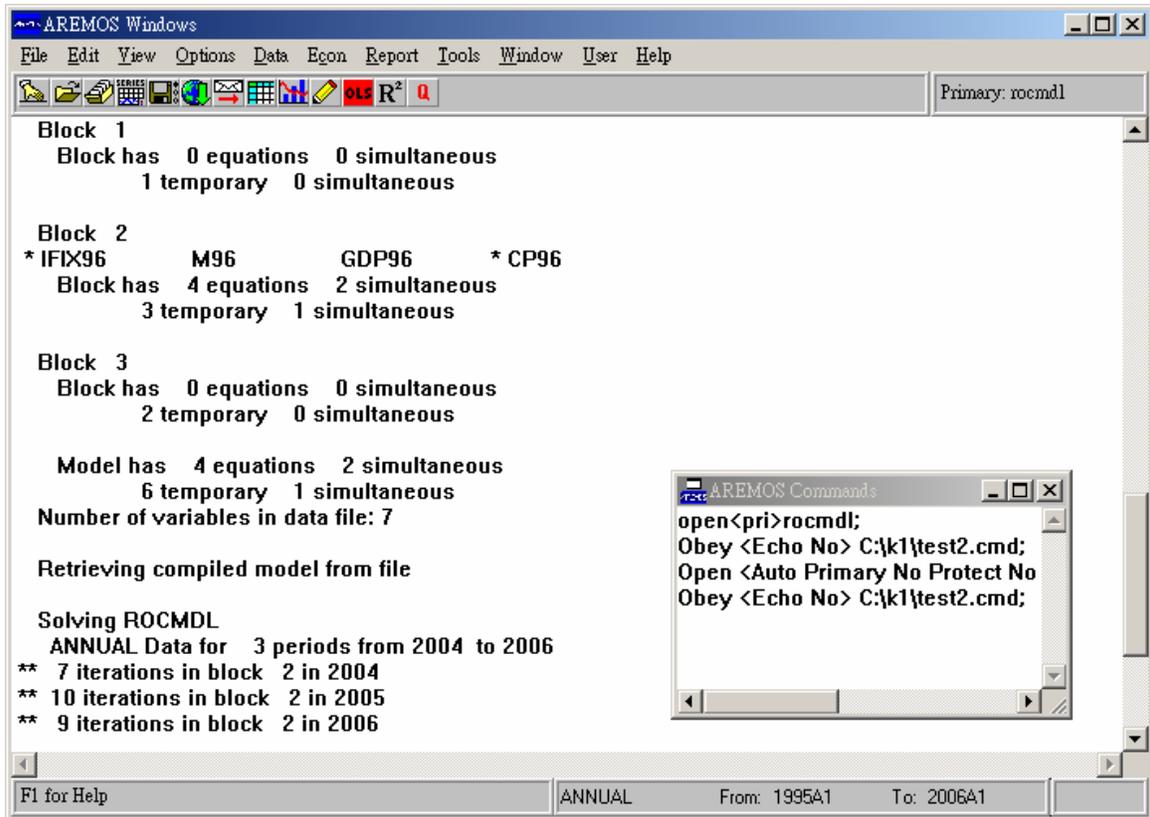
### 14.7.1 样本外预测

前面几节介绍的属标准模型，以下则介绍大模型解法。二者不同地方，是求解大模型时，数据从 SIM file 读取，而不从 AREMOS 数据读取。故指令与标准解法略有差异，惟对实际作业其实影响不大。仍延续用本章模型，指令如下：

- (1) 键入指令 `set per 1995 2006;`
- (2) 键入指令 `set compile order;`
- (3) 键入指令 `q compile;`
- (4) 键入指令 `model<autocompile>Rocmdl=cp96,ifix96,m96,gdp96;`
- (5) 键入指令  
`datafile<replace>rocmdl=#rocmdl.endogenous,#rocmdl.exogenous;`
- (6) 键入指令 `set solve solution rocmdl;`
- (7) 键入指令 `load * using niaa:*.a into rocmdl;`
- (8) 键入指令 `solve<2004 2006>;`
- (9) 键入指令 `unload * as *.sol4 from rocmdl;`

上列指令中，可以 `q compile` 查看看。其次为编译模型，用到 `model` 这个指令。`Datafile` 指令为创造一个数据文件(`rocmdl.sim`)，以便在求解过程中存入内生变量与外生变量之值。`solve` 之 `solution` 须设定与 `model` 指令后之名称(`rocmdl`)相同。`load` 指令将数据放入 `datafile` 所创造之数据文件(`rocmdl.sim`)中。求解完成后，再将所解之值从数据文件(`rocmdl.sim`)中 `unload` 下来，并以“变数.sol4”名字陈现。

详细过程如下：



任意打印几个变量，可以看到大模型求解结果(变量.sol4)与第 14.3 节标准模型

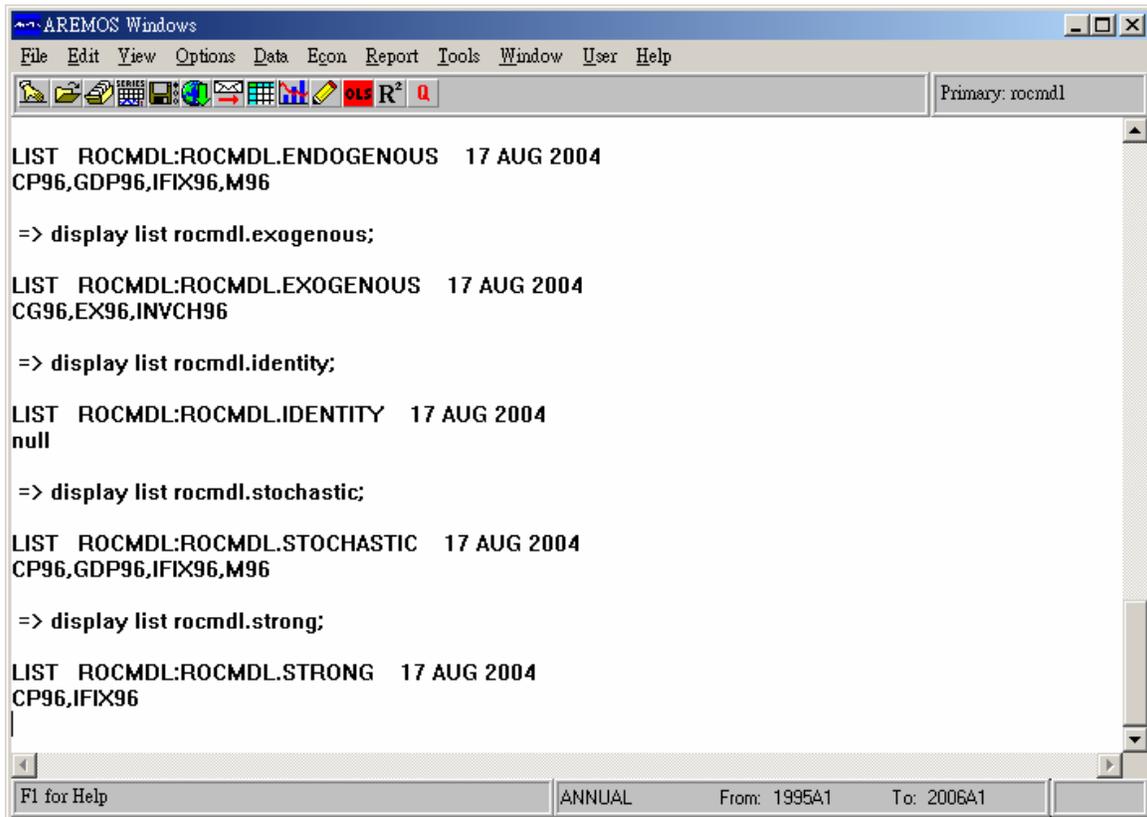
求解结果(变量.sol2)完全相同，如下所示：

- (1) 键入指令 `print<2004 2006>gdp96.sol2,gdp96.sol4;`
- (2) 键入指令 `print<2004 2006>m96.sol2,m96.sol4;`
- (3) 键入指令 `print<2004 2006>ifix96.sol2,ifix96.sol4;`

另外值得一提的是，`model` 指令在完成编译模型后，AREMOS 会产生五个 `list`，显示本模型相关变量关系，可以 `display` 出来，如下所示：

- (1) 键入指令 `display list rocmdl.endogenous;`
- (2) 键入指令 `display list rocmdl.exogenous;`
- (3) 键入指令 `display list rocmdl.identity;`
- (4) 键入指令 `display list rocmdl.stochastic;`

(5) 键入指令 **display list rocmdl.strong;**



The screenshot shows the AREMOS Windows interface. The title bar reads "AREMOS Windows". The menu bar includes "File", "Edit", "View", "Options", "Data", "Egon", "Report", "Tools", "Window", "User", and "Help". The toolbar contains various icons, including a red "R" icon. The main window displays the following text:

```
LIST ROCMDL:ROCMDL.ENDOGENOUS 17 AUG 2004
CP96,GDP96,IFIX96,M96

=> display list rocmdl.exogenous;

LIST ROCMDL:ROCMDL.EXOGENOUS 17 AUG 2004
CG96,EX96,INVCH96

=> display list rocmdl.identity;

LIST ROCMDL:ROCMDL.IDENTITY 17 AUG 2004
null

=> display list rocmdl.stochastic;

LIST ROCMDL:ROCMDL.STOCHASTIC 17 AUG 2004
CP96,GDP96,IFIX96,M96

=> display list rocmdl.strong;

LIST ROCMDL:ROCMDL.STRONG 17 AUG 2004
CP96,IFIX96
```

At the bottom of the window, there is a status bar with "FI for Help" on the left, "ANNUAL" in the center, and "From: 1995A1 To: 2006A1" on the right.

## 14.7.2 外生变数仿真

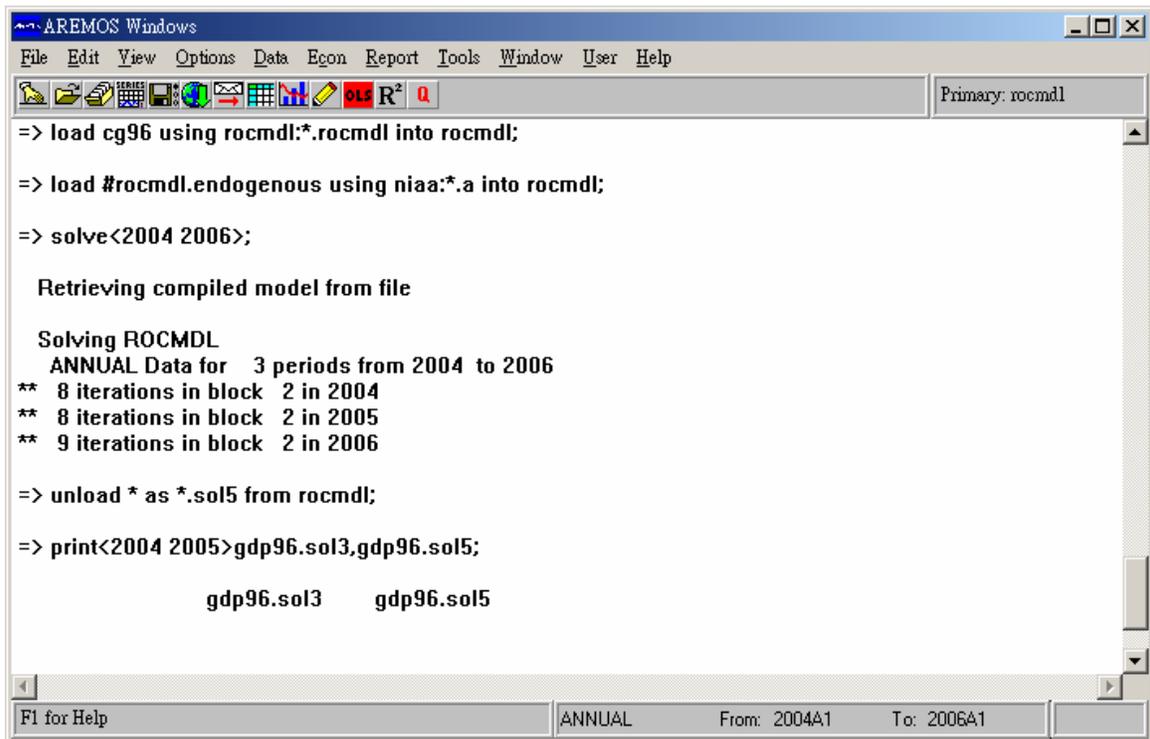
至于大模型做外生变量仿真，指令如下(与本章第 14.1 节比较):

- (1) 键入指令 **set per 2004 2006;**
- (2) 键入指令 **change<times>cg96=1.10 repeat \*;**
- (3) 键入指令 **load cg96 using rocmdl:\*.rocmdl into rocmdl;**
- (4) 键入指令 **load #rocmdl.endogenous using niaa:\*.a into rocmdl;**
- (5) 键入指令 **solve<2004 2006>;**
- (6) 键入指令 **unload \* as \*.sol5 from rocmdl;**

上列指令中，比较特殊为两个 load 指令。因 change 指令使变量 cg96 乘上 1.10(即增加 10%)，其乘积会存在一个新创的变量(cg96.rocmdl)

中,而前面一个 load 指令即是将 cg96.rocmdl 数列放进 rocmdl.sim 之数据文件(datafile 指令所创造);后面一个 load 指令为将内生变量再放进 rocmdl.sim 之资料文件;求解完成后,从数据文件(rocmdl.sim)中 unload 下来,并以”变数.sol5”名字陈现。

其完整过程如下:



```
=> load cg96 using rocmdl:*.rocmdl into rocmdl;
=> load #rocmdl.endogenous using niaa:*.a into rocmdl;
=> solve<2004 2006>;

Retrieving compiled model from file

Solving ROCMDL
ANNUAL Data for 3 periods from 2004 to 2006
** 8 iterations in block 2 in 2004
** 8 iterations in block 2 in 2005
** 9 iterations in block 2 in 2006

=> unload * as *.sol5 from rocmdl;
=> print<2004 2005>gdp96.sol3,gdp96.sol5;

          gdp96.sol3    gdp96.sol5
```

任意打印几个变量,与 14.4 节之结果(变量.sol3)比较,如下所示(由于求解 iteration 之起始值不同,尾数略有差异):

- (1) 键入指令 `print<2004 2005>gdp96.sol3,gdp96.sol5;`
- (2) 键入指令 `print<2004 2005>m96.sol3,m96.sol5;`
- (3) 键入指令 `print<2004 2005>ifix96.sol3,ifix96.sol5;`

### 14.7.3 加入 Addfactors

大模型做样本外预测，亦可加入 Addfactors，指令如下(与本章第 14.5 节比较)：

(1) 键入指令 **set per 2000 2006;**

(2) 键入指令 **compile;**

(3) 键入指令

```
datafile<replace>rocmdl=#rocmdl.endogenous,#rocmdl.exogenous;  
s;
```

(4) 键入指令 **load \* using niaa:\*.a into rocmdl;**

(5) 键入指令 **series<2004 2006>cp.add=72500,68000,64000;**

(6) 键入指令 **datafile<replace>rocmdl.a=cp96;**

(7) 键入指令 **load \* using rocmdl:\*.add into rocmdla;**

(8) 键入指令 **set solve adjust solution rocmdl addfactor rocmdla;**

(9) 键入指令 **solve<2004 2006>;**

(10) 键入指令 **unload \* as \*.so14a from rocmdl;**

上列指令说明如下：

(1) 设定数据期间。虽然预测时间始于 2004 年，但模型使用落后一期变量，故数据时间至少需提前一年。

(2) **compile<large>** 模型。

(3) 用 **datafile** 指令创建一个数据文件(**rocmdl.sim**)，以便放入数据。

(4) 用 **load** 指令将数据从数据库 (**niaa**) 转入 **datafile** 所创造之资料文件(**rocmdl.sim**)中。

(5) 输入 **addfactor** 值。内生变数之 **version** 须为 **add**。如有两个以上变量须加入 **addfactor**，依序以 **series** 输入。

- (6)用 **datafile** 指令创造另一个数据文件(**rocmdl.sim**), 以便放入 **addfactor** 值。在此资料文件(**rocmdl.sim**)等号右边为 **addfactor** 变数, 不须含 **version(.add)**, 如有两个以上变数, 以逗号隔开。
- (7)用 **load** 指令将 **addfactor** 值转入数据文件(**rocmdl.sim**)中。
- (8)开启 **solve** 指令中的 **adjust** 为 **yes**; 并设定 **solution** 为数据文件 **rocmdl** 及 **addfactor** 为数据文件 **rocmdla**。
- (9)设定预测期间, 并进行求解。
- (10)求解完成后, 将预测值从数据文件(**rocmdl.sim**)中 **unload** 下来, 并以”变数**.sol4a**”名字陈现。

其完整过程如下:

<插图>资料文件(**rocmdl.sim**)

任意打印几个变量, 与 14.5 节之结果(变量**.soladd**)比较, 如下所示(由于求解 **iteration** 之起始值不同, 尾数略有差异):

- (1)键入指令 **print<2004 2006>cp96.soladd,cp96.sol4a;**
- (2)键入指令 **print<2004 2006> ifix96.soladd,ifix96.sol4a;**
- (3)键入指令 **print<2004 2006> gdp96.soladd,gdp96.sol4a;**

## 14.8 结果陈示

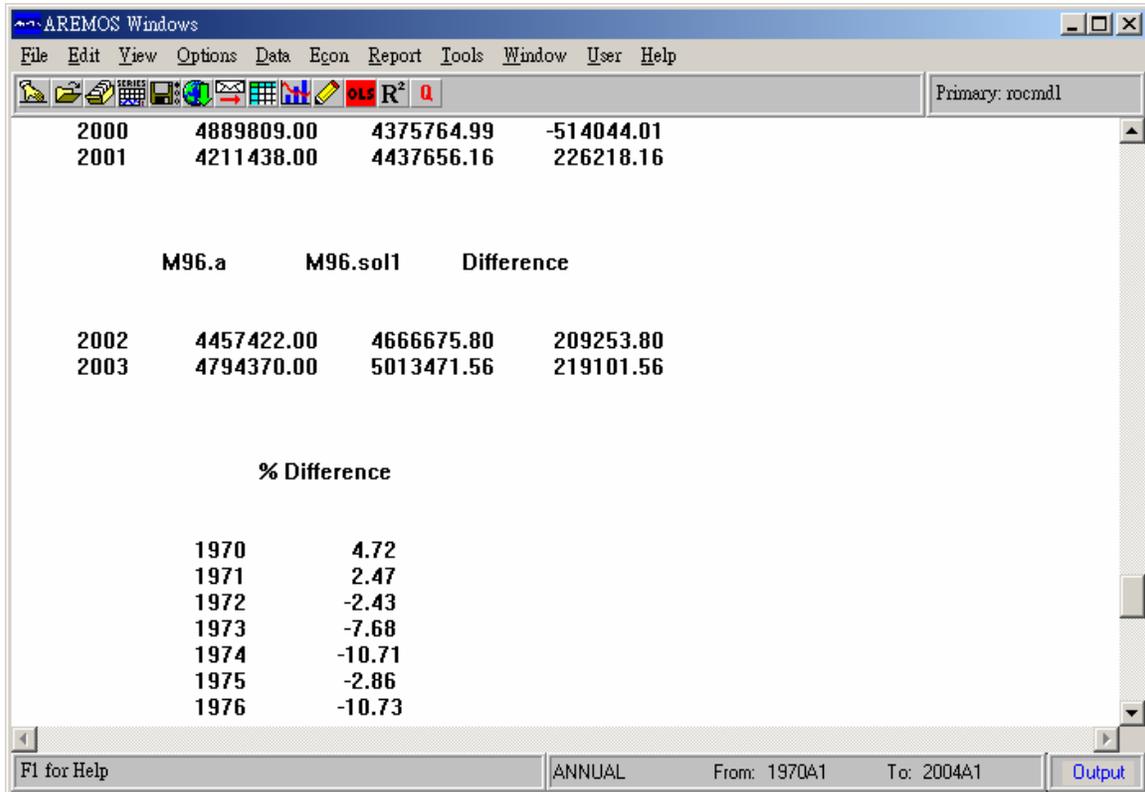
本章前面几节陈列求解结果, 用到 **print** 指令。另外, 亦可用 **m\_compare** 指令比较求解与实际值之差异, 用法如下:

<b>m_compare</b> 实际值 <b>version</b> 求解值 <b>version</b> 由 <b>list</b> 构成之层串变数
--

例如第 14.2.2 节求解结果，用 `m_compare` 指令作法如下：

- (1) 键入指令 `set per 1970 2004;`
- (2) 键入指令 `m_compare a sol1 #rocmdl.endogenous;`

并得到下列结果：



	M96.a	M96.sol1	Difference
2000	4889809.00	4375764.99	-514044.01
2001	4211438.00	4437656.16	226218.16
2002	4457422.00	4666675.80	209253.80
2003	4794370.00	5013471.56	219101.56

	% Difference
1970	4.72
1971	2.47
1972	-2.43
1973	-7.68
1974	-10.71
1975	-2.86
1976	-10.73

此外，模型求解之结果，如能迅速陈现在眼前，必可大幅提高模型作业效率。AREMOS Modeling 与 Graphic 及 Table 连结使用，恰可发挥“一贯作业”的功能(有关图、表制作，读者可参阅该章说明，此处不再赘述)。以下仅以 AREMOS Table 陈列第 14.3 或第 14.7 节样本外预测结果，以及用 AREMOS Graphic 陈示第 14.2.2 节求解结果。

## 14.9 指令摘要

### 14.9.1 check

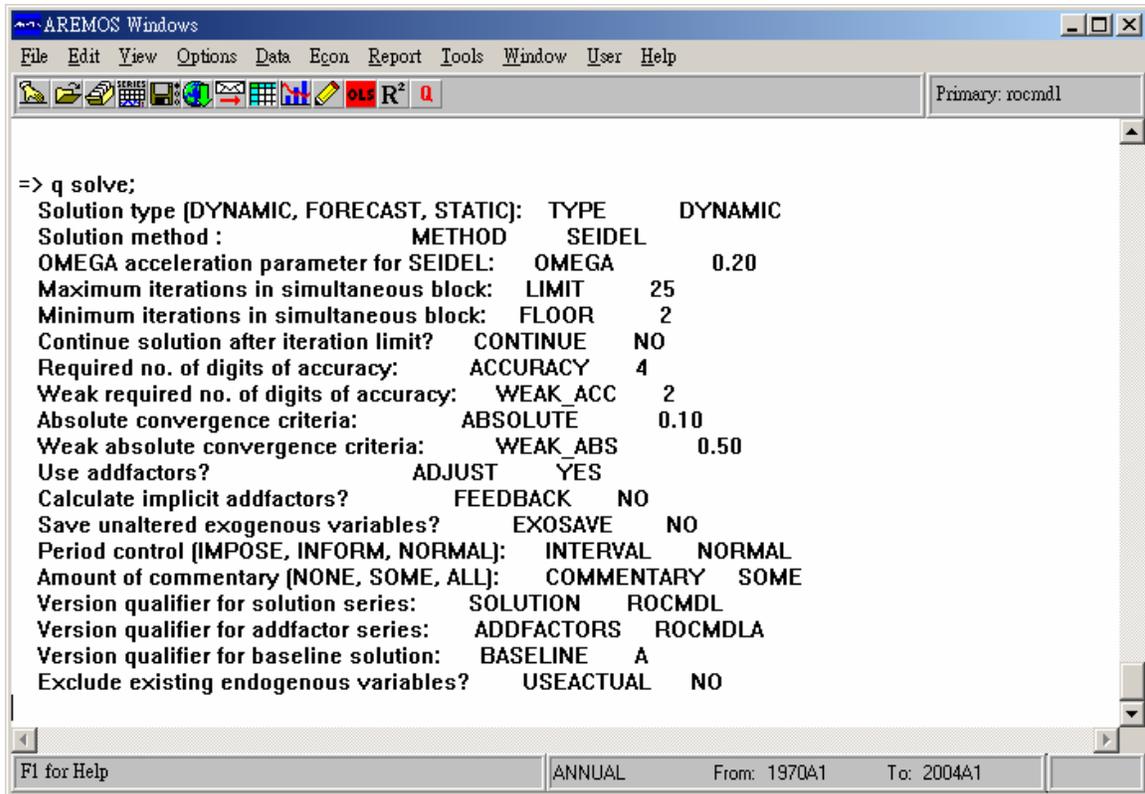
建立一个计量经济模型，当涉及众多的变量及大量数据。除模型之良窳外，数据正确与否，亦攸关求解过程与结果。为避免转录数据疏忽，在求解前先对数据作查证，乃重要步骤。以下指令可针对模型中定义式及恒等式等号两边变量数据进行勾稽核对，如有错误，相关统计量将显示异于零。

- (1) 键入指令 `open<pri>rocmdl;`
- (2) 键入指令 `open c:\warem32\niaa;`
- (3) 键入指令 `for x=#rocmdl.identity;`
- (4) 键入指令 `check<1995 2004>#x;`
- (5) 键入指令 `end;`

### 14.9.2 solve

`solve` 指令有许多 `options` 可供选择，可以 `query` 指令查阅，如下所示：

键入指令 `q solve;`



其中值得注意为：

**TYPE** 求解方式。分为动态(Dynamic)、预测(Forecast)及静态(Static)等三种求解方式可供选择。内设值为动态求解。

**METHOD SEIDEL**。求解为 Gauss-Seidel 法。

**LIMIT** 每期求解最大 iteration 次数。内设值为 25 次，不过可以放大。

**FLOOR** 每期求解达到收敛最小 iteration 次数。内设值为 2 次。

**COTINUE** 当预测不收敛时，是否继续求解下一期。内设值为 NO。

**ADJUST** 如为 Yes，开启 Addfactor(参见第 14.5 节)。内设值为 NO。

**FEEDBACK** 如为 Yes，开启隐含 Addfactor(参见第 14.6 节)。内设值为 NO。

**SOLUTION** 求解内生变数之 version。内设值 SOL。

**INTERNAL** 当数据不足时如何处置。可选择 **Impose**、**Inform** 或 **Normal**。内设值为 **Normal**。

**EXOSAVE** 如为 **Yes**，外生变量亦同时冠上 **SOLUTION** 之 **version**。

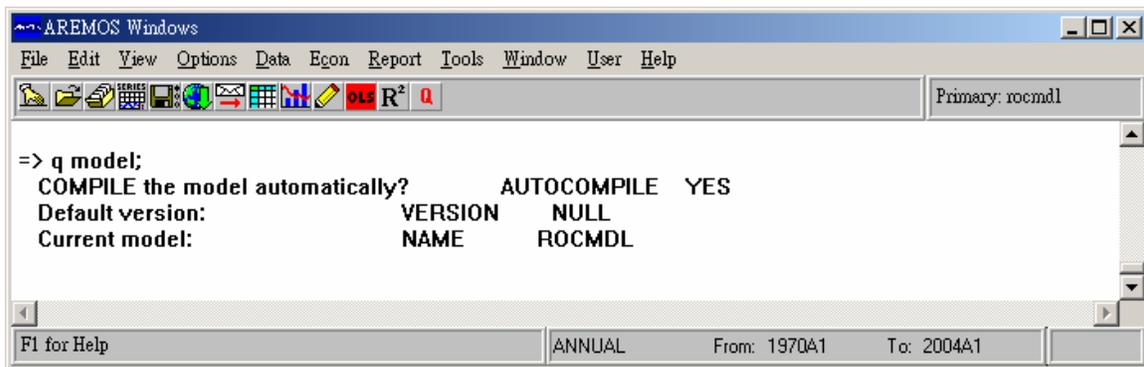
**ADDFACTORS ADJUST** 开启后，加入 **Addfactor** 内生变数之 **version**(参见第 14.5 节。内设值为 **ADD**。)

**BASELINE** 起始值(**starting value**)之 **version**。当输入特定 **version**，及拥有该 **version** 之变数数值为求解之起始值。否则以目前 **Frequency** 之变量数值为起始值。

### 14.9.3 model

**model** 指令是用来宣告模型所包含的所有方程式，并做自动编译 (**Autocompile**)。以 **query** 指令查阅 **model** 的 **option**:

**q model;**



```
AREMOS Windows
File Edit View Options Data Econ Report Tools Window User Help
Primary: rocmdl

=> q model;
COMPILE the model automatically? AUTOCOMPILE YES
Default version: VERSION NULL
Current model: NAME ROCMDL

F1 for Help ANNUAL From: 1970A1 To: 2004A1
```

**AUTO\_COMPILE** 如为 **yes**, 执行 **model** 后会自动编译, 并产生五个 **list**, 每个 **list** 包含特定变量。这五个 **list** 为:

- . **modelname .ENDOGENOUS**
- . **modelname .EXOGENOUS**
- . **modelname .IDENTITY**
- . **modelname .STOCHASTIC**
- . **modelname .STRONG**

因模型进行求解前, 一定都须经过编译(**Compile**), 因此, 如 **model autocompile** 不设为 **yes**, 于执行 **model** 指令后, 须再用 **compile** 指令以编译模型。

本章以上所介绍的指令均可写成 **cmd** 文件, 然后用 **obey** 指令执行。

附录(一):

## 财团法人经济资讯推广中心

# 「AREMOS/Windows 经济统计数据库系统」

安装使用收费标准

单位：新台币元

1.	单一份用户： 每增一份：	11 万元/每年 增收 5 万 5 仟元/每年
2.	五份用户： 每增一份：	33 万元/每年 增收 4 万 4 仟元/每年
3.	二十份用户： 每增一份：	66 万元/每年 增收 3 万 3 仟元/每年

注:

(1)学校系所、老师及学生等照上述收费标准 70%收费。

(2)系统包括 26 个国内数据库(一般性及金融性统计数据库)

(3)以上收费含每年 12 次(每个月一次)更新数据库。更新方法以光盘片邮寄。

(4)以上收费标准由本中心董事会议决定,适用于所有用户,绝无例外。

附录二: ※「键入指令法」(在 AREMOS Command 画面下)

范例(一)

- (1) 鼠标点取, 选择 AREMOS Windows 软件
- (2) 鼠标点取, 进入 AREMOS Windows 环境
- (3) 键入指令 **set freq q;set per 1982:1 2002:4;**
- (4) 键入指令 **set report dec 0;**
- (5) 键入指令 **open e:\WAREMOS\niaq;**
- (6) 键入指令 **print gdp,cp,gdp96;**
- (7) 键入指令 **set report dec 2;**
- (8) 键入指令 **print pchya(gdp), pchya (cp), pchya (gdp96);**
- (9) 键入指令 **analyze gdp, pchya (gdp96);**
- (10) 键入指令 **write gdp,cp,gdp96;**
- (11) 键入指令 **index gdp,cp,gdp96;**
- (12) 用滚动轴看上、下、左、右的东西。

- (13) 键入指令 `close niaq;`
- (14) 键入指令 `stop;`
- (15) 跳出 Windows

#### 范例(二)

- (1) 鼠标点取 选择 AREMOS Windows 软件
- (2) 鼠标点取 进入 AREMOS Windows 环境
- (3) 键入指令 `set freq m;set per 1991:1 1992:12;`
- (4) 键入指令 `set report dec 2;`
- (5) 键入指令 `open e:\WAREMOS\ind, e:\WAREMOS\trade, e:\WAREMOS\man;`
- (6) 键入指令 `print jqind,ex,n;`
- (7) 键入指令 `list n1= jqind,ex,n;`
- (8) 键入指令 `export<prn>#n1 aaa;`
- (9) 键入指令 `view aaa.prn;`
- (10) 键入指令 `export<prn nodate>#n1 bbb;`
- (11) 键入指令 `view bbb.prn;`
- (12) 键入指令 `export<prn nodate>#n1 ccc;`
- (13) 键入指令 `view ccc.prn;`
- (14) 鼠标点取 AREMOS Editor 软件，去编辑 `aaa.prn;`
- (15) 键入指令 `set freq q;set per 1982:1 1992:41`
- (16) 键入指令 `set report dec 0;`
- (17) 键入指令 `open e:\WAREMOS\niaq;`
- (18) 鼠标点取 Send Series Data，然后键入 `gdp,cp,gdp96。`
- (19) 鼠标点取 Series Edit And Display，然后叫取数据且绘图。
- (20) 键入指令 `close ind,man,niaq,trade;`
- (21) 键入指令 `stop;`

### 范例(三)

- (1) 鼠标点取 选择 AREMOS Windows 软件
- (2) 鼠标点取 进入 AREMOS Windows 环境
- (3) 键入指令 `set freq q;set per 1993:1 2003:4;`
- (4) 键入指令 `open e:\WAREMOS\niaq, e:\WAREMOS\fsm;`
- (5) 键入指令 `collapse m1a=m1a.m ave;`
- (6) 键入指令 `collapse rmtd1@1s@f=rmtd1@1st@ f.m ave;`
- (7) 键入指令 `equ equ1 m1a=gdp, rmtd1@1st@f;`
- (8) 键入指令 `equ equ2 m1a=gdp, rmtd1@1st@f ar=1;`
- (9) 键入指令 `equ equ3 m1a=gdp, m1a (-1) , rmtd1@1st@f;`
- (10) 键入指令 `equ equ4 log (m1a)=log(gdp), log(rmtd1@1st@f);`
- (11) 键入指令 `stop;`