# EViews 编程基础

中国人民大学 统计学院

陈堰平

# 1、EViews 命令基础

## 1.1 对象类型指定

新建或调用一个对象时，最一般的命令格式是

**object_type** *object_name*

其中，对象名称（object_name）自定义，对象类型必须采用 EViews 规定的标志符，它们有：coef（系数）、equation、graph、group、matrix、model、pool、rowvector（行向量）、sample、scalar、series、sspace（状态空间）、sym（对称矩阵）、system、table、text、var（向量自回归）和 vector 等.
例如命令

*equation* *eq*1

产生一个名为 eq1 的方程对象.

命令

$$matrix(3,4)\quad y$$

生成名为 y 的一个 3×4 矩阵.

还可以利用赋值语句产生新对象．例如生成一个名义变量序列，当样本期在 25 和 50 之间时，取值为 1，样本期在 60 和 85 之间时，取值为 2，其余时期为缺失值，使用命令

$$smpl\quad 25\quad 50$$
$$series\quad dummy=1$$
$$smpl\quad 60\quad 85$$
$$series\quad dummy=2$$

又如利用 y 的自然对数值与 x 回归产生方程对象 eq2

$$equation\quad eq2.ls\quad log(y)=c(1)+c(2)*x$$

EViews 将回归方程的参数估计与相关检验结果存储在名为 eq2 的方程对象中.

## 1.2 对象命令

对象命令用于指定对象的某种观测方式或进行某项过程操作，一般格式为

**action** *object*.**view_proc** **arg_list**

其中，action 包括以下四个选项：

- do 执行程序而不新打开窗口
- freeze 从当前对象观测状态（如电子表格 SpreadSheet、估计结果 Estimation Output、柱图 Bar Graph 等）生成表或图像
- print 打印当前对象观测状态
- show 打开指定对象窗口进行观测

object 指对象名称，view_proc 代表待执行的对象观测方式或操作过程，arg_list 是对观测方式和操作过程的进一步说明，其中的关键字都用空格分开.

实际应用时 action 所代表的部分常常可以省略，例如

*gdp.line*　等价于　*show gdp.line*

即绘制序列对象 gdp 的折线图．又如对人均收入（inc）和 gdp 的方程执行最小二乘过程，下面两个命令是等价的

*eq1.ls inc c gdp*　和　*do eq1.ls inc c gdp*

## 1.3 工作文件命令

（1）创建时间序列工作文件的命令是

**workfile** *name frequency start end*

或者

**workfile**(*wf* = *name*) *frequency start end*

**workfile** 命令，也可以用 **wfcreate**、**create** 替代。其中，频率（frequency）有如下选项：a（年）、s（半年）、q（季度）、m（月度）、w（周）、d（天，按一周 5 天计算）、7（天，按一周 7 天计算）和 u（非日期形式）等八种.

例如命令

*workfile wf*1 *q* 1980:1 1999:4

生成名为 wf1 的季度工作文件，时间从 1980 年 1 季度到 1999 年 4 季度.

（2） 创建截面数据工作文件的命令

**workfile** *name u 观测个数*

或者 **workfile(wf=** *name)* *u 观测个数*

（3） 创建面板数据工作文件的命令

**workfile** *name frequency start end 截面个数*

或者 **workfile(wf=***name***)** *frequency start end 截面个数*

保存工作文件使用命令 **save**，后面可以跟相关路径，如

*save a:\eviews\workfile*

注意：上面的语句中省略了文件后缀，实际上保存的是 *workfile.wf1*

类似地，**close** 命令用于关闭一个工作文件，**load** 用于打开以前保存的工作文件.

## 1.4 数据导入导出

（1）数据导入

命令格式： read (选项) 路径\文件名 序列 1 序列 2 …

例 1

```
read(t=xls, s=data,b3,name) “d:\data.xls” gdp invest cons
```

例 2

```
read(t=xls, s=data,b3,name) “d:\data.xls” 3
```

（2）数据导出

```
write(t=xls,a2,nonames,nodates) “d:\data.xls” x1 x2 x3
```

# 2、EViews 程序基础

有过编程经历的人将会发现，EViews 软件语言跟一些常用计算机高级语言有很多类似之处，比较容易学习.

## 2.1 有关程序的基本操作

新建一个程序是在主菜单中选择 File/New/Program，屏幕会显示一个文本编辑窗口，如图 1．也可以键入命令 program 打开程序编辑窗口，后面还可以输入程序名称，如 prg1

*program prg*1


图 1. EViews 程序编辑窗口

保存程序时，点击窗口上方工具栏中的 Save 或 Save As 即可，程序文件的扩展名是.prg. 打开原有程序只需在主菜单选择 File/Open/Program，再输入需打开或调用的程序名称即可，与工作文件操作类似.

运行一个程序是选择程序编辑窗口工具栏中的 Run 按钮，屏幕会弹出运行程序定义对话框，见图 2.

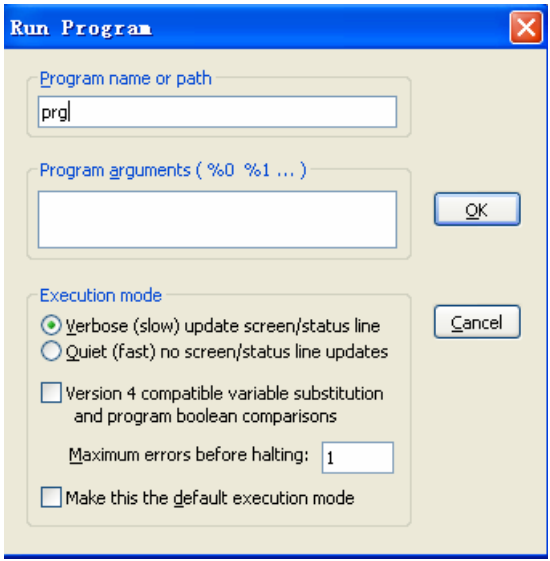


图 2.运行程序定义对话框

用户可在对话框最上面的空行输入程序的名称，如 prg2．在 Program arguments 下面指定命令行参数值．

Execution mode 下面的选项是关于程序的运行模式，点选 Verbose update screen 表示运行程序同时刷新计算机屏幕，Quiet no screen 则不更新．对话框右下角的空格中的数值表示计算机终止程序运行条件即程序执行中错误数的最大值．点选对话框最下面的选项表示将以上定义作为默认运行模式．用户按 F1 键可中断程序运行．

## 2.2 控制变量

控制变量（Control Variable）是在 EViews 程序中用于代替具体数值的变量，一旦对其赋值，就可以在程序任何地方进行调用．与 EViews 的其他变量不同的是，定义了控制变量以后不会再工作文件中保存，只会保存在内存中。定义一个控制变量的格式是在变量名前标“!”，且变量名不能多于 15 个字符，例如

$$!x \qquad !1234 \qquad !pi$$

对控制变量赋值使用赋值号“=”，如

$$!x = 2.5 \qquad !1234 = 0 \qquad !pi = 3.14159$$

经过赋值的控制变量可以在表达式中使用，如

$$!counter = !counter + 1$$

$$smpl \quad 1978:1+!i \quad 1994:12+!i$$

## 2.3 字符串变量

字符串变量（String Variable）的取值是一段文本，并在头尾使用双引号．变量名前的标记为“%”，赋值号仍然是“=”，例如

$$\%gdp = "gross\ domestic\ product"$$

$$\%pi = "3.14159"$$

$$\%sample1 = "1990:1\ \ 2000:4"$$

$$\%armas = "ar(1)\ \ ar(2)\ \ ma(1)"$$

赋值后的字符串变量可用于相关命令，如

*smpl* %*sample*1 等价于 *smpl* 1990:1 2000:4

$$equation \quad eq1.ls \quad y \quad y(-1) \quad \%armas$$

等价于

$$equation \quad eq1.ls \quad y \quad y(-1) \ ar(1) \ ar(2) \ ma(1)$$

下面是几个常用的字符串函数．*@left* 返回字符串左端开始给定个数的字符；*@right* 返回字符串右端开始给定个数的字符；*@mid* 返回字符串指定起点开始向右给定个数的字符．例如

$\%st1 =$"*I feel happy to go there*"

$\%st2 =$"*They believe EViews is a wonderful software*"

$\%st3 =$"  *learn*"（字母 *l* 前面有 3 个空格）

$$\%qq = @left(\%st1,15) + @right(\%st3,6) + @mid(\%st2,13,7)$$

函数 $@left(\%st1,15)$ 表示从变量 $\%st1$ 左端开始截取 15 个字符，即 "$I\ \ feel\ happy\ to$"；函数 $@right(\%st3,6)$ 表示从变量 $\%st3$ 右端开始截取 6 个字符，即 "$learn$"（字母 $l$ 前有 1 个空格）；函数 $@mid(\%st2,13,7)$ 表示从变量 $\%st2$ 左起第 13 个字符（字母 $e$ 和 $E$ 之间的空格）开始，向右截取 7 个字符，即 "$EViews$"（字母 $E$ 前有 1 个空格）. 因此，$\%qq$ 的结果为 "$I\ \ feel\ happy\ to\ learn\ Eviews$". 另外，函数 $@str$ 将控制变量取值转化为字符，若 $!x = 200$，则 $@str(!x)$ 返回字符 200.

## 2.4 替换变量

先看一个例子

$$\%x = "gdp"$$

$$ls \;\; \%x \;\; c \;\; \%x(-1)$$

$$\%x = "cons"$$

$$ls \;\; \%x \;\; c \;\; \%x(-1)$$

这里，通过改变字符串变量$\%x$的取值来代替它在命令中的实际内容，因此称该变量为替换变量（Replacement variable）.

当替换变量与其他字母、数字或变量合用时，应用大括号加以界定，如

$$!a = 2$$

$$\%b = "temp"$$

$$series \;\; x\{!a\}$$

$$vector(5) \;\; x\_\{\%b\}1$$

这组命令相当于定义了一个名为 x2 的序列和名为 x_temp1 的向量.

## 2.5 命令行参数

命令行参数是一些特殊的字符串变量．在运行程序时，计算机会出现对话框让用户给出参数的具体值．若程序中用到了这些参数，则每次运行程序都可以改变参数值，使程序有较强的通用性．在 EViews 中，命令行参数名默认为%1，%2，%3 等等．

来看这样一段程序

$$equation\ \ eq1$$

$$smpl\ \ 1980:1\ \ 1990:12$$

$$eq1.ls\ \ \%0\ \ c\ \%1\ \ \%1(-1)\ \ inc$$

如果在图 2 所示的运行程序对话框中定义 $\%0 = gdp$ 和 $\%1 = m2$，则程序最后一行命令相当于

$$eq1.ls\ \ gdp\ \ c\ m2\ \ m2(-1)\ \ inc$$

# 3、程序控制

与很多计算机语言类似，EViews 也提供了控制程序执行的一些方法．其作用是对于不同的条件，用户可以有选择地或者反复地执行一些命令．

## 3.1 IF 条件句

IF 条件句由关键词 if 开始，紧跟着是条件表达式，然后是关键词 then．条件表达式中常使用逻辑运算符 and 和 or，如果需要还可以加上圆括号．若表达式为真，则直到 endif 的所有命令都被执行，否则命令被跳过，例如

```
if  !i > 2  and  !i < 8  then
    smpl  1965:1 1985:1+!i
endif
```

表示当控制变量$!i$取值在 3 到 7 之间时，产生新样本期$1965:1\ 1985:1+!i$.

IF 语句还可以包括 else 从句，当且仅当 if 条件表达式不成立时执行．因此，若表达式为真，则运行 if 和 else 之间的命令，否则执行 else 到 endif 之间的命令，例如

$$
\begin{aligned}
&if \quad !scale > 0 \quad then \\
&\qquad series \quad age1 = age/!scale \\
&else \\
&\qquad series \quad age1 = age \\
&endif
\end{aligned}
$$

表示当控制变量$!scale$取值为正时，新序列$age1$用原序列$age$除以该控制变量值产生；当控制变量取值为负或 0 时，新序列$age1$等于原序列$age$.

IF 语句也能用于字符串变量并可以嵌套，如

```
if  %0 ="girl"  or  %0 ="boy"  then
    series  id =1

else
    if  %0 ="woman"  then
        series  id = 2

    else
        series  id = 3
    endif
endif
```

在条件句中，有时候会用到字符串的大小比较问题．EViews 中，字符串的大小比较是对应字符在 ASCII 码中的序号为基础的，越靠前则越小．两个字符串相等的条件是字符个数相同并且对应字符一样．由于阿拉伯数字在 ASCII 码中的序号要小于字母，所以由字母组成的字符串一定大于由阿拉伯数字组成的字符串．下面的不等式都是成立的：

"*a*"<"*abc*"　　"*b*2"<"*bc*"　　"*a*">156　　" *b*"<"*a*"

## 3.2 FOR 循环

FOR 循环可实现将控制或字符串变量的不同取值代入相同一组命令反复执行．但对不同的变量，语法有区别．

**1．对控制变量使用 FOR 循环**

先看一个例子

$$\begin{aligned} &for \quad !i = 1 \quad to \quad 10 \\ &\qquad series \quad decile\{!i\} = (inc <= rank\{!i\}) \\ &next \end{aligned}$$

这里，变量$!i$两次被用作替换变量，分别产生名为 decile1 到 decile10 十个序列和调用 rank1 到 rank10 十个变量．

在 for 语句中一般还应该有关键词 step，后面跟循环变量每次递进步长数值，上面的例子省略了这一部分，系统就默认为步长是 1．语句

$$for \quad !i = 8 \quad to \quad -8 \quad step \quad -1$$

表示每循环一次，$!i$ 的取值减 1，循环总次数为 17 次．

对控制变量使用 FOR 循环的作用之一是改变样本期，下面一段程序可对逐渐扩大的样本，估计一系列的回归方程

$$for \ \ !month = 12 \ \ to \ \ 72$$
$$smpl \ \ 1994:1 \ \ 1994:1+!month$$
$$equation \ \ eq\{!month\}.ls \ \ sales \ \ c \ \ orders$$
$$next$$

与 IF 语句类似，FOR 循环也可以嵌套使用.

```
matrix(15,10) matr
for !i=1 to 15

for !j=1 to 10
        matr(!i,!j)=(!i-1)*10+!j
    next
next
```

这段程序用于产生一个 15 行 10 列的矩阵，并将其命名为 matr．第四行赋值语句的意义是对第$!i$行第$!j$列的元素进行赋值.

**2．对字符串变量使用 FOR 循环**

对字符串变量使用 FOR 循环时需要在变量名后列出它的取值清单，如

```
for %x gdp gnp ndp nnp
    equation {%x}1.ls %x c {%x}(-1) time
next
```

此程序实际估计了名为 gdp1、gnp1、ndp1 和 nnp1 的四个方程.

在 for 语句中可以包括多个字符串变量，但后面的取值一定要注意对应，例如

```
for %1 %2 %3 51:1 78:4 plan 79:1 99:4 market
    smpl %1 %2

    equation {%3}eq.ls income c product
next
```

仔细观察可以发现，该循环共进行了两次，变量%1 和%2 用于改变样本期，先后估计了名为 planeq 和 marketeq 的两个回归方程.

循环嵌套的构造方法跟嵌套的 IF 条件句很类似，如

```
!i = 1
for  %1  50  60

for  %2  80  90  99
        smpl  %1  %2
        equation  eq{!i}.ls  gdp  c  invest

        !i = !i + 1
    next
next
```

与上面例子不同，这里共循环了 6 次，样本期分别为 50 到 80、50 到 90、50 到 99、60 到 80、60 到 90 和 60 到 99，并且注意控制变量 $!i$ 起到了给诸方程命名的作用.

## 3.3 WHILE 循环

与 FOR 循环不同，实际应用中经常遇到的是只有满足特定条件时才执行循环命令的情况，这就需要使用 WHILE 循环语句. WHILE 循环以 while 语句起始，以 wend 语句结尾，中间可包括任意数目的命令，也允许嵌套使用.

while 语句由关键词 **while** 和包含控制变量的表达式构成. 该表达式具有逻辑真 / 假值或一个数字取值. 对于取值为一个数字的情况，0 表示假，其他非零数值则代表真.

若表达式为真，则循环执行到 wend 为止的命令，否则直接运行 wend 后的程序．例如

```
!unit = 1
!i = 0

while  !unit <= 1000000  and  !i < 12
    smpl  1990:1  1998:1+!i
    series  popul{!unit} = population/!unit

!unit = !unit*10
    !i = !i+1
wend
```

程序的第一部分作用是给 while 语句中的控制变量!*unit* 和!*i* 赋初值，并且一定注意应在循环命令中对控制变量取值更新，如语句!*unit* = !*unit* *10 和!*i* = !*i* +1起到了避免出现死循环的作用．

在循环语句中，有时希望当满足某个条件时退出当前循环，这时可以配合使用 IF 条件句和 exitloop 命令．另外，命令 stop 可随时中断程序的运行．

# 参考文献

[1] 易丹辉，数据分析与 EVIEWS 应用（第二版），人大出版社，2008 年
[2] 张晓峒，Eviews 使用指南与案例，机械工业出版社，2007 年
[3] Eviews Command Reference，Quantitative Micro Software, LLC,2007
[4] EViews 6 User’s Guide I&II, Quantitative Micro Software, LLC,2007