

上海财经大学

硕士研究生教学教案



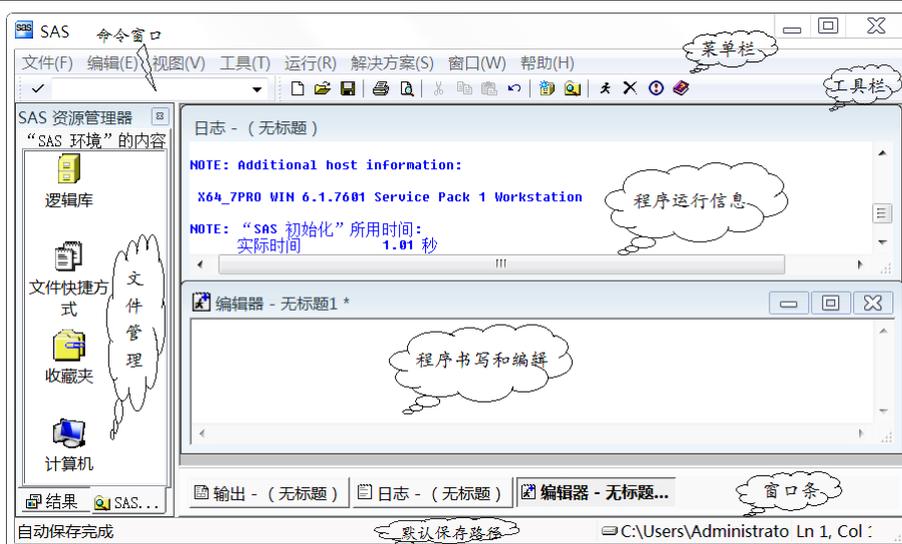
20××~20××学年第×学期

课程名称 计量软件在金融中的应用
授课班级 ××××××
主讲教师 沈根祥
职 称 教授
学 院 经济学院

课 程 类 别	专业必修	学分	2	学时	33	讲课+上机：33 学时
教材参考资料	<p>教 材</p> <p>1.沈根祥. 金融数据整理和分析—基于 SAS 软件. 上海财经大学出版社 2021.12。</p> <p>参考书</p> <p>1.蔡瑞胸 (Ruey S.Tsay), 王远林、王辉、潘家柱 译. 金融时间序列分析 (第3版). 人民邮电出版社, 2012年9月.</p> <p>2.朱世武. SAS 编程技术教程 (第二版). 清华大学出版社, 2013年10月.</p> <p>3.Burlew M. combining and modifying SAS data sets: Examples (second edition). 2009, SAS Institute Inc., Cary, NC, USA.</p>					

教案 1

授课方式	讲授+演示+上机	课次	1
		课时	3
教 学 内 容 (提 纲) ——第一章：SAS 简介+第二章：SAS 数据集			
一. SAS 软件简介 1. 简介 SAS 软件的产生和发展, SAS 的功能和局限性, 和其它软件的比较 2. SAS 界面 BASE/SAS 的视窗构成, 功能菜单和快捷键。SAS 文件快捷方式及其建立。			
二. SAS 数据集 1. 数据集结构 SAS 数据集的定义, 变量和观测, 文件名和变量名缩略列示方法 2. SAS 逻辑库 SAS 逻辑库的创建, SAS 两级文件名 3. SAS 数据集变量 SAS 数据集变量的属性, 数据集变量的输入输出格式。			
教学目的、要求、重点和难点			
目的: 了解和熟悉 SAS 软件的基本构成和操作, 理解 SAS 数据集的构成和管理			
要求: 1. 能够启动和关闭 SAS 软件; 2. 了解三个基本窗口的功能和操作; 3. 能够建立、合并和删除 SAS 逻辑库; 4. 理解并记住 SAS 数据的输入输出格式			
重点: 1. SAS 数据集的构成; 2. SAS 逻辑库的建立; 3. SAS 数据的输入输出格式			
难点: 1. SAS 逻辑库及其建立; 2. SAS 数据的输入输出格式			
教学过程			
第一章 SAS 简介→SAS 界面			



三个窗口及其功能菜单

→ SAS 文件快捷键建立

→ SAS 程序的强行终止

第二章

SAS 数据集

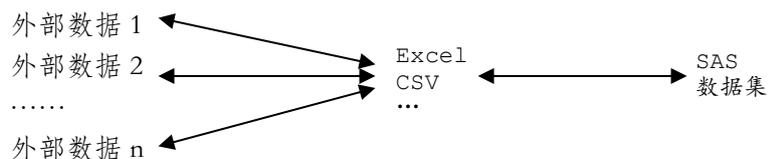
		变量				
		Name	Sex	Age	Height	Weight
观 测	—1	阿尔弗雷德	男	14	69	112.5
	—2	爱丽丝	女	13	56.5	84
	—3	芭芭拉	女	13	65.3	98
	—4	凯露	女	14	62.8	102.5
	—5	亨利	男	14	63.5	102.5
	—6	詹姆斯	男	12	57.3	83
	—7	简	女	12	59.8	84.5
	—8	雅妮特	女	15	62.5	112.5
	—9	杰弗瑞	男	13	62.5	84
	—10	约翰	男	12	59	99.5

→ SAS 逻辑库



→ SAS 变量

→ SAS 数据集创建



讨论题、思考题和作业

讨论题：与 Python 软件相比，SAS 软件有什么优缺点？

思考题：如何用 SAS 将“02/03/12”读取为 2012 年 2 月 3 日？

作业：第一章习题 2、4、5；第二章习题 2、3、6

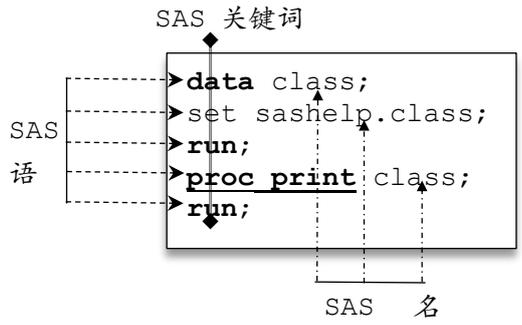
课后小结

1. 学生对 SAS 软件是否有用存在疑虑，认为 Python 和 R 比 SAS 好。需要讲清楚 SAS 的优点和应用场景。
2. 学生对输入输出格式较为陌生，短时间内难以理解。通过例子和实际演示加强对输入输出格式的理解。
3. 导入标准的外部数据，生成 SAS 数据集，并将其放入自建的 SAS 逻辑库应作为最基本的技能要求。

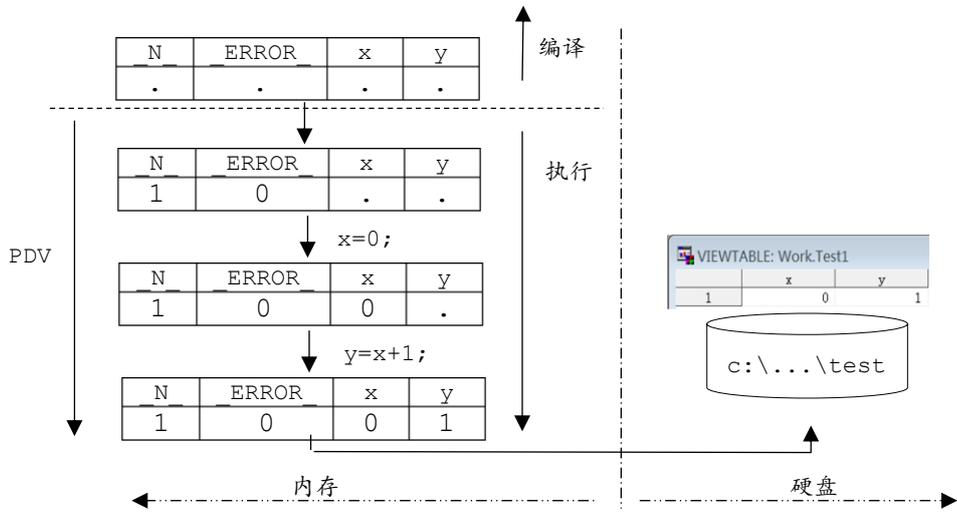
其它

教案 2

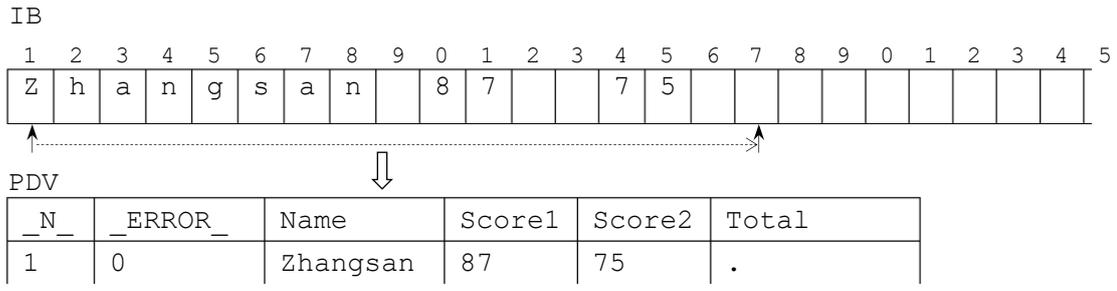
授课方式	讲授+演示+上机	课次	2
		课时	3
教 学 内 容 (提 纲) —— 第三章：DATA 步编程和数据处理：基础 (I)			
<p>一. SAS 程序</p> <p>1. SAS 语句 SAS 语句构成：关键词、语句结束标志、语法</p> <p>2. SAS 表达式 SAS 变量和常量、SAS 函数、SAS 运算符。</p> <p>二. DATA 步原理</p> <p>1. 程序的编译 编译阶段的认为、程序数据向量 PDV、输入输出数据集</p> <p>2. DATA 步内置循环 DATA 的开始和结束边界、变量初始化和循环赋值、变量观测自动输出</p> <p>3. DATA 步流程 定义变量生成数据集、输入数据集为基础生产数据集、读取外部文件生成数据集。</p>			
教学目的、要求、重点和难点			
<p>目的： 掌握 SAS 语句和 SAS 程序的构成，理解 DATA 步的运行机理。</p> <p>要求：</p> <ol style="list-style-type: none"> 1. 能够正确编写简单的 DATA 步程序； 2. 熟悉 SAS 表达式，掌握 SAS 运算 2. 对 DATA 步执行发生的错误能够给合理解释； 3. 正确理解 DATA 步中的 PDV 和 IB（输入缓冲区）的概 <p>重点：</p> <ol style="list-style-type: none"> 1. SAS 语句和 SAS 程序构成； 2. DATA 的运行机理。 <p>难点：</p> <ol style="list-style-type: none"> 1. DATA 步运行机理，尤其是内置循环中 PDV 变量取值的变化 2. Where 表达式中的算符及其功能 			
教学过程			
<p>SAS 语句和程序→</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">开始边界-----></div> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> data 数据集名; 语句 </div> </div> <div style="display: flex; align-items: center; justify-content: center; margin-top: 5px;"> <div style="margin-right: 10px;">结束边界-----></div> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> run; </div> </div>			



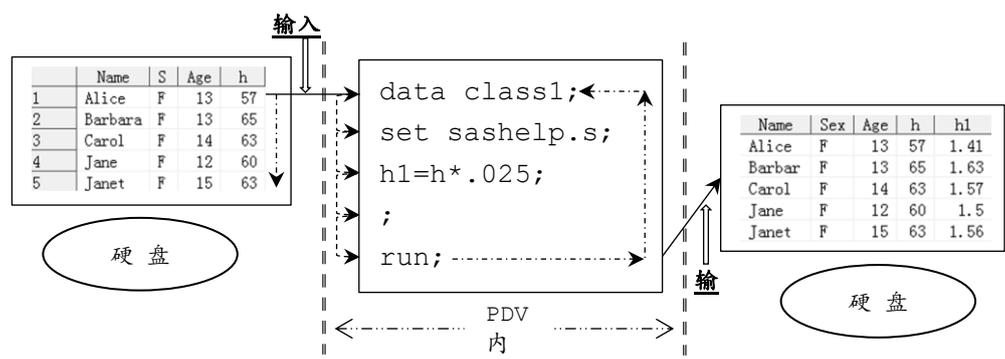
- SAS 表达式
- DATA 步中的 PDV



→ DATA 步中的 IB



→ DATA 步中的内置循环和自动输出



讨论题、思考题和作业

讨论题: DATA 步数据处理原理和其它软件比较。

思考题: 为什么需要输入缓冲区? 以 SAS 数据集为输入数据集时为什么不需要输入缓冲区? 如何用让 SAS 将“02/03/12”读取为 2012 年 2 月 3 日?

作业: 第三章习题 1、3、5、6

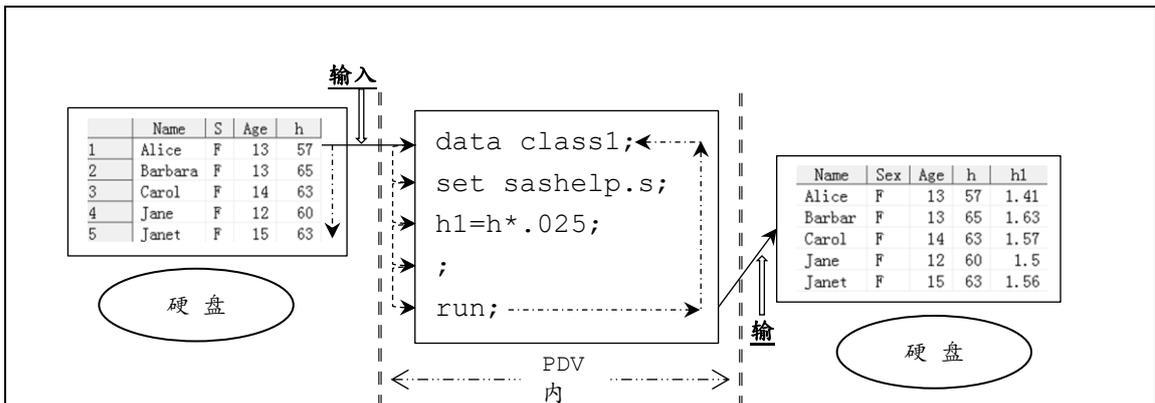
课后小结

1. 首次学习软件的学生, 尤其是文科和财经类学生, 由于逻辑思维训练不够, 对 DATA 步的处理原理和流程理解起来很吃力。用直观的示意图来解释有较好效果。
2. SAS 表达式中的比较运算中, 学生对一些特殊运算较为陌生, 例如=:、in 以及 where 表达式中的通配符使用等。需要较多的不同例子进行说明和演示。

其它

教案 3

授课方式	讲授+演示+上机	课次	3
		课时	3
教 学 内 容 (提 纲) —— 第三章：DATA 步编程和数据处理：基础 (II)			
一. DATA 步操作 1. set 语句读取 SAS 数据集 set 语句选项、set 数据集选项 2. 观测操作 If 语句和 where 语句过滤观测、改变观测值、输出节点控制。 3. 变量操作 keep 语句和 drop 语句实现变量筛选、keep=和 drop=数据集选项实现变量筛选，变量属性编辑。			
二. DATA 步执行流程控制 1 1. 分支控制 If-then 语句、if-then-else 语句，Do-End 语句组 2. 循环控制—Do-End 循环 Do-End 循环与 DATA 步内置循环的关系、指标变量循环。			
教学目的、要求、重点和难点			
目的： 掌握用 DATA 步实现 SAS 数据集的编辑，理解并掌握分支控制和循环控制语句的使用。 要求： 1. 正确理解 set 语句的各种选项的用途，并采用选项完成特定的任务。 2. 能够实现数据集观测和变量的筛选，实现对数据集变量的编辑。 3. 理解并实现分支流程控制和指标变量循环控制。 重点： 1. SAS 数据集编辑； 2. Do-End 循环。 难点： 1. DATA 步中多个 set 语句的执行、一个 set 的多次执行中的数据指针。 2. Do-End 循环和 DATA 步内置循环的关系			
教学过程			
set 语句读取 SAS 数据集→			

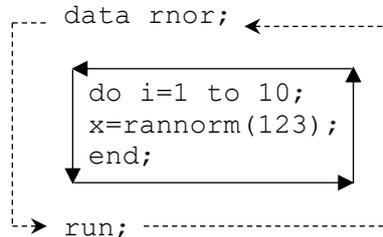


- 用 if 语句和 if-then-delete 语句筛选观测
- 用 where 语句筛选观测
- 用数据集选项 where=选择观测
- 用 drop 和 keep 语句筛选变量
- keep=和 drop=数据集选项筛选变量
- 用 rename 语句和 rename=数据集选项为变量改名、informat 和 format 语句改变输入输出格式
- 用函数 input () 和 put () 改变变量的数据类型、用 retain 语句实现新定义变量的取值保留

DATA 步流程控制→

If-then 分支控制→

End-Do 循环控制→



→output 语句的采用：及时输出 PDV 中变量值

讨论题、思考题和作业

讨论题：DATA 步中的 if 语句和 if-then 语句的区别。

思考题：DATA 步中的 Do-End 循环会改变 DATA 中自动变量 `_n_` 的值吗？

作业：第三章习题 8、10、11、12、14、17、19

课后小结

1. 这部分内容是 DATA 步较深入的内容，学生在首次学习时往往理解不够深刻，其主要原因是对其运行机理的理解不够深入和扎实。可通过更多的例子进行说明。
2. 加深理解的最好方法，是首先给出一段程序，在运行前让学生解读并预期运行结果。然后运行程序，让学生判断运行结果是否和预期一致，为什么不一致？然后再深入讲解学生解读程序中的错误。

其它

教案 4

授课方式	讲授+演示+上机	课次	4
		课时	3
教 学 内 容 (提 纲) —— 第 三 章 : DATA 步编程和数据处理: 基础 (III)			
<p>一. DATA 步流程控制 2</p> <p>1. 条件循环 Do-While 和 Do-Untile 语句;</p> <p>2. 指标变量循环和条件循环的联合使用 Do i=m to n while (cond1); Do i=m to n Untile (cond2);</p> <p>二. 数据集排序</p> <p>1. 数据集排序 排序过程 Proc sort, By 变量、By 组。</p> <p>2. By 组识别 DATA 步中 by 语句激活 set 语句打开的排序数据集、by 组标识变量 firs.vname 和 last.vname 的使用。</p> <p>三. 数据集合并</p> <p>1. 上下串接 Set 语句实现上下串接合并</p> <p>2. 左右并接 Merge 语句实现左右并接。</p>			
教学目的、要求、重点和难点			
<p>目的: 掌握用 DATA 步内条件循环控制, 掌握数据集的排序和合并操作。</p> <p>要求:</p> <ol style="list-style-type: none">1. 正确理解两种循环的联合使用及其必要性。2. 掌握数据集排序的编程, 理解并使用 by 组识别变量进行编程。3. 掌握 set 语句和 merge 语句及性能的数据集非匹配合并和匹配合并, 掌握采用多个 set 语句的匹配合并。 <p>重点:</p> <ol style="list-style-type: none">1. 条件循环, 条件循环和指标变量循环的联合使用;2. 数据集排序和合并。 <p>难点:</p> <ol style="list-style-type: none">1. 数据集排序后 by 组识别变量的使用。2. 多个 set 语句的匹配合并。			

教学过程

第一部分：Do-End 循环 2

Do-While 语句循环 → Do-Until 语句循环

→ Do i=n to m While (condt1)

→ Do i=n to m Until (condt1)

```
data rnor; ←-----  
do i=1to 10 while (x<=5);  
x=rannorm(123)+1;  
end;  
run; ----->
```

例子：银行存款

第二部分：数据集排序

Proc sort → By 变量、By 组

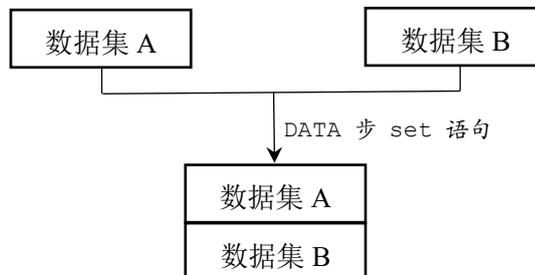
→ By 组标识和标识变量

→ DATA 步中 by 语句激活排序

→ 变量 first.vname 和 last.vname 的使用

第三部分：数据集合并

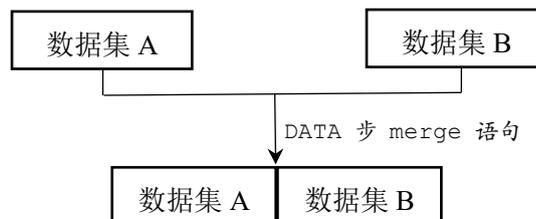
set 语句串接合并



→ 非匹配串接

→ 匹配串接 (by vnames)

merge 语句并接合并



→ 非匹配并接

→ 匹配并接 (by vnames)

→多个 set 语句匹配合并：采用 point=v 选项

讨论题、思考题和作业

讨论题：数据集合并中 DATA 步的处理流程。

思考题：为什么要进行匹配合并？什么情况下匹配合并和非匹配合并得到的数据集相同？DATA 步中有多个 set 语句时，内置循环何时结束？同一个 set 语句读取多个数据集时，内置循环合适结束？

作业：第三章习题 20、21、23、24

课后小结

1. 数据集合并是一种数据整理中常用的操作，难点在于匹配合并及其机理。通过实际例子能够更清楚地展示有关操作和响应的结果。
2. DATA 步中的多 set 语句讲解是一个难点，核心在于每个 set 语句都有其数据读取指针，且互不干扰。
3. 要通过实例展示排序数据集中 by 组标识变量的妙用。

其它

1. 第三章内容时 DATA 步的基础和核心，内容较多，分配的时间较多，分三次 9 个课时讲授。
2. 对例子和习题中用到的数据进行介绍，给出获取方式。

教案 5

授课方式	讲授+演示+上机	课次	5
		课时	3
教 学 内 容 (提 纲) ——第四章：DATA 步编程和数据处理：进阶 (I)			
<p>DATA 步读取外部数据</p> <p>1. 内嵌流式数据读取 内嵌流式数据的边界、input 语句的四种读取方式；</p> <p>2. 读取文件的引导—infile 语句 Infile 语句的格式，infile 语句的选项；</p> <p>3. 文件标识语句—filename 读取文件的引导—infile 语句 Infile 语句的格式，infile 语句的选项；</p>			
教学目的、要求、重点和难点			
<p>目的： 掌握用 input 语句读取外部数据，生成 SAS 数据集的编程。</p> <p>要求：</p> <ol style="list-style-type: none"> 1. 正确理解 input 的四种读取方式。 2. 掌握数据读取指针控制符的使用方法。 3. 掌握 infile 语句各种选项的使用方法。 <p>重点：</p> <ol style="list-style-type: none"> 1. input 的四种读取方式； 2. infile 的使用。 <p>难点：</p> <ol style="list-style-type: none"> 1. 列表读取方式中的修饰符使用。 2. 多次读取一行和多次读取一列一行指针控制和列指针控制。 3. infile 语句的各种选项及其功能 			
教学过程			
<p>内嵌数据→</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 20px;"> <p>开始</p> <p>内嵌数据</p> <p>结束</p> </div> <div style="border: 1px solid black; padding: 10px; text-align: left;"> <pre> data tem; input name \$ age; datalines; 张曼 13 李丽 14 王鹏 12 ; run; </pre> </div> </div>			

→input 语句

- 四种数据读取方式→变量列表(list)
 - 列方式 (column)
 - 格式化 (formatted)
 - 命名方式 (named)

→数据读取规则：指针控制

- 指针的列位置控制符
 - 绝对位控制和相对位置控制
- 指针的行位置控制符
 - 绝对位控制和相对位置控制
- 指针定位于指定字符之后

→infile 语句及其选项：更多控制

- 数据行读取范围设定
 - firstobs=和 obs=
- 数据行长度不足的指针控制
 - missover、trunccover 和 stopover
- 数据指针位置变量
 - line=lv 和 column=cv
- 数据指针位置变量
 - missover、trunccover 和 stopover
- 设定分隔符
 - length=v 和 DSD
- 规定每次读取的记录宽度 (字符数)
 - lrecl=n
- 以字符长度确定数据行
 - recfm=f
- 读取多个外部文件
 - 定义当前文件名变量：filevar=v
 - 定义当前目录中文件名变量：memvar=v
 - 读取一类外部文件：pipe 选项和通配符
- 读取网页—URL 选项

讨论题、思考题和作业

讨论题：灵活读取磁盘文件——infile 语句及其选项。

思考题：在 infile 语句中采用 reconf=f 选项后，input 读取记录时，数据行的结束标志还起作用吗？为什么？如果选项 reconf=f 和 lrecl=n 一同使用，适合读取什么数据？此时的 missover、trunccover 和 stopover 还起作用吗？为什么？

作业：第四章习题 1、2、4、6

课后小结

1.input 语句读取外部数据，是 DATA 步编程较为高级的内容，十分灵活，同时又十分有效，功能十分强。学生在初次接触时往往难以把握，需要降低授课进度，多举例子进行演示，更为直观地让学生理解有关语句和选项的功能。

2. 可以采用读取网页的例子，讲解 infile 语句众多选项的作用，以网页爬取为例，能够极大提高学生积极性。
3. 让学生自己设计一个任务，用 input 语句读取外部数据，例如从网页上拷贝沪深 300 股票代码，然后读取为一个 SAS 数据集。

其它

读取外部数据，往往是一种高级编程技术，并非所有学生都需要掌握。讲解这部分内容时，要根据情况进行取舍。如果是金融计量学的学生，则要少讲，对于大数据专业的学生，则需要多讲，将深讲透。

教案 6

授课方式	讲授+演示+上机	课次	6
		课时	3
教 学 内 容 (提 纲) ——第四章：DATA 步编程和数据处理：进阶 (II)			
1.DATA 步数组 1.数组的定义和引用 2.数组的应用； 2.DATA 步数据查询 1.同一数据集跨行查询 2.数据集间查询； 3.DATA 步间数据传递 1.用数据集传递 2.DATA 步接口			
教学目的、要求、重点和难点			
目的： 让学生理解数组的概念和使用方法，掌握数据简单查询的编程技术，掌握 DATA 步间的数据传递方法。			
要求： 1.正确理解 DATA 数组的含义和必要性。 2.掌握 ARRAY 语句及其选项的使用。 3.掌握数组的三个主要应用场景。 4.用多个 set 语句实现跨行查询，用数据实现跨行查询。 5.用数组实现数据集间查询			
重点： 1.数组定义和使用；			
难点： 1.数组的定义方式。 2.DATA 步接口例程 <code>symputx()</code> 和函数 <code>symget()</code> 的使用			

教学过程

第一部分：数组及其应用

数组的概念→变量引用的便利

- 数组的定义：array 语句
- 数组要素：数组名、下标、类型
 - 一维数组和多维数组
 - 下标为*的数组
 - 没有变量列表的数组
 - 临时数组
- 数组的定义：array 语句
- 一维数组和多维数组
- 一维数组和多维数组
- 数组的引用
 - 通过下标引用
 - 引用所有元素—数组名[*]

数组的应用→变量操作中的数组应用

- 和 Do-End 结合实现变量的批量操作
- 和 lag 函数结合实现变量的跨行操作
- 把数据集读为数组，在 DATA 步实现数据集的矩阵操作。

第二部分：数据集查询

数据查询的概念和应用场景→数据查询类型和方法

- 同一数据集的跨行查询
 - 多个 set 语句多次读取同一数据集
 - 将数据集读入数组
- 不同数据集间查询
 - 多个 set 语句多次读取不同数据集
 - 将数据集读入数组

第三部分：DATA 步间数据传递

DATA 步间数据传递的必要性：DATA 步边界

- DATA 步间的数据传递
 - 用数据集过渡进行传递
 - 用 DATA 步接口进行传递
 - 将 DATA 步数据传给全局宏变量—例程 symputx()
 - 引用宏变量函数 sysget()

讨论题、思考题和作业

讨论题：(1) 数组的作用；(2) DATA 步数组和其它语言中数组的区别。

思考题：(1) 定义数组时，为什么变量不能作为数组下标？(2) DATA 步间的数据传递，还有其它方法吗？

作业：第四章习题 7、9、10、12

课后小结

1. 数组以及 DATA 步间数据传递，也是 DATA 步编程较为高级的内容，十分灵活，同时又十分有效，功能十分强。在选择性讲解的同时，适当降低授课进度，多举例子进行演示，便于降低理解和掌握的难度。

2. 具体应用能提高学生学习兴趣，在数据集查询和输出传递内容中，引入学生容易接触到的较为熟悉的数据处理例子，能取得较好的效果。

其它

由于不对 SAS 宏语言编程进行详细讲解，学生对 DATA 步间数据传递中用到的例程 `sysputx()` 和函数 `sysget()` 设计到的概念会比较陌生，可通过形象的数据容器比喻，对宏变量进行直观介绍。

教案 7

授课方式	讲授+演示+上机	课次	7
		课时	3
教 学 内 容 (提 纲) ——第六章：时间序列数据处理			
1.时间序列数据集 1.时间序列数据集要件 2.时间区间及其操作; 2.频率转换 3.变量变换 1.H-P 滤波 2.移动窗口内的计算 3.其它变换			
教学目的、要求、重点和难点			
目的: 熟悉时间序列数据集的特点, 掌握日期和时间的常用操作和处理; 要求: 1.正确理解时间序列数据集的构成。 2.掌握日期、时间的拆分和合并。 3.掌握用函数 <code>intnx()</code> 进行日期和时间间隔平移的方法, 掌握用函数 <code>intck()</code> 计算间隔的方法 3.掌握 <code>proc expand</code> 过程的使用方法。 重点: 1.日期、时间的拆分和合并; 2. <code>proc expand</code> 过程的使用 难点: 1.函数 <code>intnx()</code> 和函数 <code>intck()</code> 的使用 2.分数差分和求和 — <code>proc expand</code> 语句 <code>convert</code> 选项/ <code>transform=(FDIF,d)</code> — <code>proc expand</code> 语句 <code>convert</code> 选项/ <code>transform=(FSUM,d)</code>			

教学过程

第一部分：日期时间的操作

时间序列数据集：具有日期时间戳的有序序列

- 日期和时间变量操作
- 不带月份和日的日期
- 日期的拆分和合并
- 小于秒的时间表示
- 时间的拆分和合并

时间间隔操作：

- 间隔（interval）的概念和间隔名构成
- 间隔平移操作——函数 `intnx()` 的应用
- 间隔（interval）数计算——函数 `intck()` 的应用。

第二部分：抽样频率转换

抽样频率转换的必要性：频率匹配

- 高频转低频
- 抽样汇总
- 插值
- 低频转高频
- 插值

第三部分：变量变换

变量变换：过程 `proc expand`

- H-P 滤波
- 移动窗口内的计算
 - 中心化统计量计算
 - 后移统计量计算
 - 前移统计量计算
 - 累积统计量计算
- 分数差分和求和
 - 分数差分
 - 语句 `convert 选项/transform=(FDIF,d)`
 - 分数求和
 - 语句 `convert 选项/transform=(FSUM,d)`

讨论题、思考题和作业

讨论题：时间区间或者间隔（interval）有什么作用？

思考题：为什么 year2.25 是无效的间隔名？在应用函数 intnx() 平移时间时，按间隔名规定的间隔移动到的终点时间超出间隔规定的范围时，intnx() 是如何确定终点日期和时间的？

作业：第六章习题 2、3、4、6、9、11、12、13

课后小结

1. 本章内容为时间序列数据处理数组方法，由于金融数据绝大多数为时间序列数据，因此本章内容十分重要。
2. 日期和时间间隔的概念和相关操作（函数 intnx()）在日期时间的处理上具有十分独特的功能，正确使用能够处理一些复杂的问题，但同时也是一个难点。应该分配更多的时间讲解。

其它

H-P 滤波是宏观经济分析中常用的数据处理方法，而分数差分是针对金融时间序列中的长记忆模型建模前的数据预处理，差分的阶数要用第 8 章中的 proc varmax 过程来估计。

教案 8

授课方式	讲授+演示+上机	课次	8
		课时	3
教 学 内 容 (提 纲) ——第七章：一元时间序列分析			
1.过程 proc arima 1.单位根非平稳性检验 2.模型定阶和估计; 3.预测 2.过程 proc autoreg 1.时间序列回归模型 2.自回归模型; 3.ARCH 和 GARCH 模型 3.结构突变检验 1.Chow 方法 2.BP 方法			
教学目的、要求、重点和难点			
目的: 通过两个过程的学习,掌握一元时间序列模型的建模方法,包括均值建模和方差建模; 要求: 1.会使用两个过程进行单位根检验。 2.会建立 ARIMA 模型,包括阶数确定和模型参数估计。 3.会建立 ARCH 模型和各种 GARCH 模型 4.会通过模型检测评价建模效果。 5.会进行结构变换检验。 重点: 1.模型建模; 难点: 1.变点检验的 BP 方法			

教学过程

第一部分: `proc arima` 过程

一元时间序列的均值模型 ARMA (p, q)

$$y_t = \phi_0 + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} - \theta_1 a_{t-1} + \dots - \theta_q a_{t-q} + a_t, a_t \sim i.i.N(0, \sigma^2)$$

滞后算子表示为

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) y_t = \phi_0 + (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) a_t$$

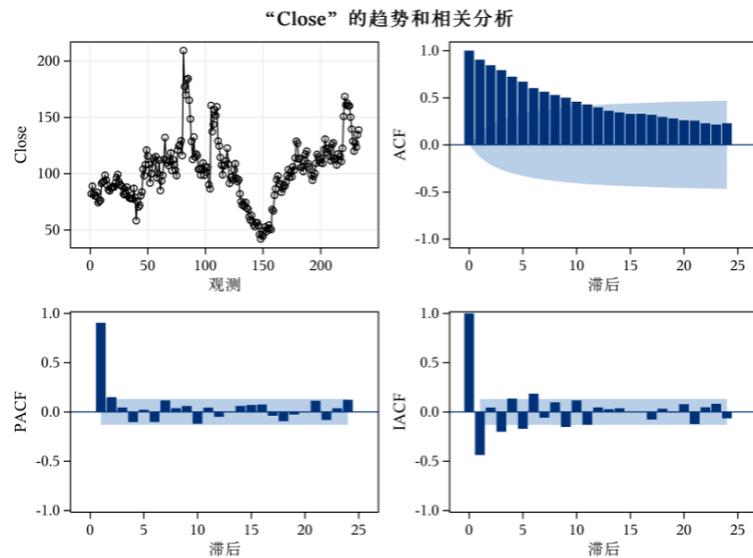
建模过程:

→ 是否平稳: 单位根检验

IDENTIFY VAR=variable STATIONARITY=(options);

→ 确定阶数: ACF、EACF、AIC

IDENTIFY VAR=variable Option



→ 参数估计:

ESTIMATE options;

→ 设定阶数: $p=n | (n_1, n_2, \dots, n_k)$
 $q=m | (m_1, m_2, \dots, m_k)$

→ 设定阶数: 选择估计方法

Method=CLS | ML | ULS

→ 预测: `forecast` options;

→ 设定预测步数: `lead=n`

→ 输出预测结果: `out=SASdt`

第二部分: `proc autoreg` 过程

时间序列回归: 具有自回归的误差项

$$y_t = c + \beta_1 x_{1t} + \beta_2 x_{2t} + \dots + \beta_k x_{kt} + u_t$$

$$u_t = \epsilon_t - \phi_1 u_{t-1} - \phi_2 u_{t-2} - \dots - \phi_m u_{t-m}$$

建模过程:

→ 是否平稳: 单位根检验

```

model close=/stationarity=(option);
→自回归模型估计:
model y=/nlag=n;
→时间序列回归:
→估计方法: OLS+N-W——covest=neweywest
model y=x1 x2 ... xk/covest=neweywest;

```

参数估计					
变量	自由度	估计	标准误差	t 值	近似 Pr > t
Intercept	1	-0.000392	0.000393	-1.00	0.3189
r_m	1	0.8383	0.0771	10.87	<.0001

WLS+Yule-Walker

```

model y=x1 x2 ... xk/nlag=p;

```

自回归参数的估计			
滞后	系数	标准误差	t 值
1	0.051999	0.028788	1.81
2	0.051014	0.028810	1.77
3	-0.052863	0.028807	-1.84
4	0.034282	0.028810	1.19
5	0.089349	0.028788	3.10

```

→ARCH 和 GARCH 模型:
→ARCH 效应检验: ARCHTEST=(options)
→GARCH 模型估计
model y=x1 x2 ... xk/nlag=n GARCH=(options);

```

第三部分: 结构突变检验

结构变点的概念及其检验:

→Chow 检验

$$\text{Model I: } y_t = c^{(1)} + \beta_1^{(1)} x_{1t} + \beta_2^{(1)} x_{2t} + \dots + \beta_k^{(1)} x_{kt} + u_t^{(1)}, \quad 0 \leq t \leq T_1$$

$$\text{Model II: } y_t = c^{(2)} + \beta_1^{(2)} x_{1t} + \beta_2^{(2)} x_{2t} + \dots + \beta_k^{(2)} x_{kt} + u_t^{(2)}, \quad T_1 \leq t \leq T$$

$$\text{Model III: } y_t = c + \beta_1^{(2)} x_{1t} + \beta_2^{(2)} x_{2t} + \dots + \beta_k^{(2)} x_{kt} + u_t^{(2)}, \quad 0 \leq t \leq T$$

→BP 检验

$$\begin{aligned}
y_t &= \mathbf{x}'_t \boldsymbol{\beta} + \mathbf{z}'_t \boldsymbol{\delta}_1 + u_{1t}, & t \in (1, T_1] \\
y_t &= \mathbf{x}'_t \boldsymbol{\beta} + \mathbf{z}'_t \boldsymbol{\delta}_2 + u_{2t}, & t \in (T_1, T_2] \\
&\dots & \dots \\
y_t &= \mathbf{x}'_t \boldsymbol{\beta} + \mathbf{z}'_t \boldsymbol{\delta}_m + u_{mt}, & t \in (T_m, T]
\end{aligned}$$

→第一种: SupF

$$H_0: \delta_1 = \dots = \delta_k = 0; \quad H_1: \exists i \in \{1, 2, \dots, k\}, \delta_i \neq 0$$

$$\text{sup}F(k) = \frac{1}{T} \left(\frac{T - (k+1)q - p}{kq} \right) (R\hat{\boldsymbol{\delta}})' (\text{Var}(R\hat{\boldsymbol{\delta}}))^{-1} (R\hat{\boldsymbol{\delta}})$$

→第二种: UDmaxF 和 WDmaxF

原假设 H_0 : 不存在断点; 备择假设 H_1 : 断点个数不超过k

$$\text{WDmax}F(M, \alpha) = \max_{k \leq M} \frac{c_\alpha(1)}{c_\alpha(k)} \text{sup}F(k)$$

→第三种: $\text{Sup}F(l+1|l)$

$$\text{sup}F(l+1|l) =$$

$$(T - (l + 1)q - p) \frac{\text{SSR}(\hat{\beta}, \hat{\delta} | \hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_l) - \text{SSR}(\hat{\beta}, \hat{\delta} | \hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_l; \hat{\tau}_{l+1})}{\text{SSR}(\hat{\beta}, \hat{\delta} | \hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_l)}$$

讨论题、思考题和作业

讨论题: proc arima 和 proc autoreg 的结合使用?

思考题: (1) 单位根检验种的 kpss 方法和其它方法的区别是什么? (2) 如何检验股票波动受交易量的影响?

作业: 第七章习题 1、3、4、6、7

课后小结

1. 本章内容为一元时间序列数据的基本分析方法, 要求学生掌握基本的模型和方法的 SAS 实现。
2. 如果在前导课程(如金融计量经济学、时间序列分析等)中, 没有学习过 GARCH、BP 检验之类的内容, 则可以不介绍相关功能。

其它

结构变点检验种, BP 检验属于高级内容, 可以不讲或者选讲。

教案 9

授课方式	讲授+演示+上机	课次	9
		课时	3
教 学 内 容 (提 纲) ——第八章：多元时间序列分析			
1. 平稳向量自回归模型 1. 定阶和估计 2. 脉冲响应分析 3. 格兰杰因果检验			
2. 非平稳向量自回归模型 1. 协整检验 2. 误差修正模型			
3. 多元波动模型 1. DCC 模型 2. 边际 GARCH 模型			
教学目的、要求、重点和难点			
目的： 掌握多元时间序列模型的建模方法及其 SAS 实现；包括均值建模和方差建模；均值建模包括平稳向量自回归模型建模和非平稳向量自回归模型建模（协整分析）。			
要求： 1. 用 proc varmax 建立 VAR 模型。 2. 用 proc varmax 进行脉冲响应分析、格兰杰因果检验。 3. 用 proc varmax 进行协整分析，并建立误差修正模型 4. 用 proc varmax 建立多元波动 DCC 模型			
重点： 1. VAR 模型建模，协整分析，误差修正模型建模； 2. 多元波动 DCC 模型建模			
难点： 1. 协整检验			

教学过程

第一部分：平稳向量自回归建模

多元时间序列的均值模型 VAR(p)

$$\mathbf{y}_t = \mathbf{c} + \Phi_1 \mathbf{y}_{t-1} + \Phi_2 \mathbf{y}_{t-2} + \cdots + \Phi_p \mathbf{y}_{t-p} - \Theta_1 \mathbf{a}_{t-1} + \Theta_2 \mathbf{a}_{t-2} + \cdots + \Theta_q \mathbf{a}_{t-q} + \mathbf{a}_t, \mathbf{a}_t \sim N(\mathbf{0}, \Sigma)$$

建模过程：

→模型定阶

$$\text{AIC} = \log(|\hat{\Sigma}|) + \frac{2r_b k}{T}$$

$$\text{AICC} = \log(|\hat{\Sigma}|) + \frac{2r_b k}{T - r_b}$$

$$\text{HQC} = \log(|\hat{\Sigma}|) + \frac{2r_b k \log(\log(T))}{T}$$

$$\text{SBC} = \log(|\hat{\Sigma}|) + \frac{2r_b k \log(T)}{T}$$

$$\text{FPE} = \left(\frac{T + r_b}{T - r_b}\right)^k |\hat{\Sigma}|$$

→模型估计

```
model y1 y2 y3/options;
```

→阶数设定: $p=n | (n1, n2, \dots, nk)$

$q=m | (m1, m2, \dots, mk)$

→选择估计方法

```
Method=CLS | ML | ULS
```

→变量选项

```
Center: 变量零均值化
```

```
nonint: 模型没有常数向量。
```

```
trend=linear | quad: 模型趋势项。
```

→脉冲响应分析

```
model y1 y2 y3/print=(impulse(<options>)) ;
```

→格兰杰因果检验

```
causal group1=(varlist1) group2=(varlist2);
```

第二部分：非平稳向量自回归建模

VAR(p) 模型的误差修正形式

$$\Delta \mathbf{y}_t = \alpha \beta' \mathbf{y}_{t-1} + \Phi_1^* \Delta \mathbf{y}_{t-1} + \Phi_2^* \Delta \mathbf{y}_{t-2} + \cdots + \Phi_{p-1}^* \Delta \mathbf{y}_{t-p+1} + \mathbf{a}_t$$

$$\mathbf{y}_t = \mathbf{c} + \Phi_1 \mathbf{y}_{t-1} + \Phi_2 \mathbf{y}_{t-2} + \cdots + \Phi_p \mathbf{y}_{t-p} - \Theta_1 \mathbf{a}_{t-1} + \Theta_2 \mathbf{a}_{t-2} + \cdots + \Theta_q \mathbf{a}_{t-q} + \mathbf{a}_t, \mathbf{a}_t \sim N(\mathbf{0}, \Sigma)$$

→协整检验 (Johansen 方法):

检验模型:

→**type1**: ECM 模型没有常数项, 检验模型为

$$\Delta \mathbf{y}_t = \alpha \beta' \mathbf{y}_{t-1} + \Phi_1^* \Delta \mathbf{y}_{t-1} + \Phi_2^* \Delta \mathbf{y}_{t-2} + \cdots + \Phi_{p-1}^* \Delta \mathbf{y}_{t-p+1} + \mathbf{a}_t$$

→ **type2**: ECM 有常数项但没有时间趋势项, 常数项在误差修正项内, 检验模型为

$$\Delta \mathbf{y}_t = \alpha(\beta' \mathbf{y}_{t-1} + \mathbf{d}) + \Phi_1^* \Delta \mathbf{y}_{t-1} + \Phi_2^* \Delta \mathbf{y}_{t-2} + \dots + \Phi_{p-1}^* \Delta \mathbf{y}_{t-p+1} + \mathbf{a}_t$$

检验模型的常数项为 \mathbf{ad} , 是受约束 (restricted) 的常数项。

→ **type3**: ECM 模型有常数项但没有时间趋势项, 常数项以截距形式出现, 不受约束。检验模型为

$$\Delta \mathbf{y}_t = \mathbf{c} + \alpha \beta' \mathbf{y}_{t-1} + \Phi_1^* \Delta \mathbf{y}_{t-1} + \Phi_2^* \Delta \mathbf{y}_{t-2} + \dots + \Phi_{p-1}^* \Delta \mathbf{y}_{t-p+1} + \mathbf{a}_t$$

检验模型的常数项为 \mathbf{c} 。

→ **type4**: ECM 模型中有常数项和时间趋势项, 常数项以截距形式出现, 不受约束, 时间趋势项在误差修正项内。检验模型为

$$\Delta \mathbf{y}_t = \mathbf{c} + \alpha(\beta' \mathbf{y}_{t-1} + \mathbf{d}t) + \Phi_1^* \Delta \mathbf{y}_{t-1} + \Phi_2^* \Delta \mathbf{y}_{t-2} + \dots + \Phi_{p-1}^* \Delta \mathbf{y}_{t-p+1} + \mathbf{a}_t$$

检验模型的时间趋势项为 \mathbf{adt} , 是 (系数) 受约束的时间趋势项。

→ **type5**: ECM 模型有常数项和时间趋势项, 常数项和时间趋势项均以截距形式出现, 不受约束。检验模型为

$$\Delta \mathbf{y}_t = \mathbf{c} + \mathbf{c}_1 t + \alpha \beta' \mathbf{y}_{t-1} + \Phi_1^* \Delta \mathbf{y}_{t-1} + \Phi_2^* \Delta \mathbf{y}_{t-2} + \dots + \Phi_{p-1}^* \Delta \mathbf{y}_{t-p+1} + \mathbf{a}_t$$

模型中时间趋势项为 $\mathbf{c}_1 t$ 。

语句及选项:

Model 语句选项 / cointetest = (johansen <= (type = value) >) 和选项 noint 和 trend = 配合

→ 建立误差修正模型:

cointeg rank = n <options>;

例子:

第三部分: 多元波动模型建模

DCC 模型:

$$\text{Var}(\mathbf{a}_t | \mathcal{F}_{t-1}) = \Sigma_t$$

$$\Sigma_t = D_t \boldsymbol{\rho}_t D_t, D_t = \text{diag}(\sigma_{1,t}, \sigma_{2,t}, \dots, \sigma_{k,t})$$

$$\sigma_{i,t}^2 = \omega_i + \alpha_i a_{i,t}^2 + \beta_i \sigma_{i,t}^2, i = 1, 2, \dots, k$$

$$\boldsymbol{\rho}_t = (\text{diag}(Q_t))^{-\frac{1}{2}} Q_t (\text{diag}(Q_t))^{-\frac{1}{2}}$$

$$Q_t = (1 - \theta_1 - \theta_2) \bar{Q} + \theta_1 \hat{\boldsymbol{\eta}}_{t-1} \hat{\boldsymbol{\eta}}'_{t-1} + \theta_2 Q_{t-1}, 0 < \theta_1 + \theta_2 < 1$$

DCC 模型设定: →

GARCH form = DCC subform = GARCH | EGARCH | TGARCH | ...;

例子:

讨论题、思考题和作业

讨论题：协整检验模型中如何设定常数项和时间趋势项？

思考题：（1）协整检验中，迹检验和最大特征值检验的检验结果总是一致的吗？

（2）如果协整检验结果表明，协整关系个数等于向量维数，这意味着什么？如何理解检验结果？

作业：第八章习题 1、2、3、4、5

课后小结

1. 本章内容为多元时间序列数据的基本分析方法，基本要求是掌握平稳 VAR 模型的建模、脉冲响应分析和格兰杰因果检验等常用操作。

2. 如果在前导课程（如金融计量经济学、时间序列分析等）中，没有学习过多元 GARCH，则可以不介绍相应功能。

其它

结构变点检验种，BP 检验属于高级内容，可以不讲或者选讲。

教案 10

授课方式	讲授+演示+上机	课次	10
		课时	3
教 学 内 容 (提 纲) ——第九章：线性高斯状态空间模型			
<p>1. 线性高斯状态空间模型的引入</p> <p>1. 时变系数回归模型</p> <p>2. 测量误差模型</p> <p>3. 因子模型</p> <p>2. 线性高斯状态空间模型的估计</p> <p>1. 模型设定</p> <p>2. Kalman 滤波和平滑</p> <p>3. proc ssm 的应用</p>			
教学目的、要求、重点和难点			
<p>目的： 掌握线性高斯状态空间模型的建模方法及其 SAS 实现。</p> <p>要求：</p> <ol style="list-style-type: none"> 1. 设定 proc ssm 过程中的状态方程。 2. 设定 proc ssm 过程中的观测方程。 3. 用 proc ssm 过程估计状态空间模型。 4. 用 proc ssm 过程进行建模效果评价。 <p>重点：</p> <ol style="list-style-type: none"> 1. proc ssm 中的模型设定； 2. 状态空间模型应用 <p>难点：</p> <ol style="list-style-type: none"> 1. 状态方程的设定； 2. 时间序列模型和面板数据模型的状态空间表示 			
教学过程			
<p>第一部分：线性高斯状态空间模型</p> <p>模型引入→</p> $y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \dots + \Phi_p y_{t-p}$			

→变系数回归模型—时变 CAPM

$$r_t = \alpha + \beta r_{mt} + \epsilon_t, t = 1, 2, \dots, T$$

$$\beta_t = c + \phi \beta_{t-1} + \eta_t, t = 1, 2, \dots, T$$

→因子模型

$$\mathbf{y}_t = \mathbf{c} + \mathbf{Z}\boldsymbol{\alpha}_t + \boldsymbol{\epsilon}_t$$

$$\boldsymbol{\alpha}_t = \boldsymbol{\Phi}\boldsymbol{\alpha}_{t-1} + \boldsymbol{\eta}_t$$

模型一般形式→

$$\begin{cases} \mathbf{y}_t = \mathbf{Z}\boldsymbol{\alpha}_t + \mathbf{X}_t\boldsymbol{\beta} + \boldsymbol{\epsilon}_t & \text{观测方程} \\ \boldsymbol{\alpha}_t = \boldsymbol{\Phi}\boldsymbol{\alpha}_{t-1} + \mathbf{W}_t\boldsymbol{\gamma} + \boldsymbol{\eta}_t & \text{状态方程} \\ \boldsymbol{\epsilon}_t \sim i.i.N(\mathbf{0}, H), \boldsymbol{\eta}_t \sim i.i.N(\mathbf{0}, Q) & \text{误差项} \\ \{\boldsymbol{\epsilon}_t\} \perp \{\boldsymbol{\eta}_t\} & \end{cases}$$

→模型估计

→Kalman 滤波:

$$\ln p(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T) = -\frac{T}{2} \ln(2\pi) - \frac{1}{2} \sum_{t=1}^T [\ln |\mathbf{F}_t| + \mathbf{v}_t' \mathbf{F}_t^{-1} \mathbf{v}_t]$$

$$\mathbf{v}_t = \mathbf{y}_t - \mathbf{Z}\boldsymbol{\alpha}_{t|t-1}, \mathbf{F}_t = \mathbf{Z}\boldsymbol{\Sigma}_{t|t-1}\mathbf{Z}' + H$$

$$\boldsymbol{\alpha}_{t+1|t} = \boldsymbol{\Phi}\boldsymbol{\alpha}_{t|t-1} + \mathbf{K}_t\mathbf{v}_t, \boldsymbol{\Sigma}_{t+1|t} = \boldsymbol{\Phi}\boldsymbol{\Sigma}_{t|t-1}\mathbf{L}'_t + Q$$

$$\mathbf{K}_t = \boldsymbol{\Phi}\boldsymbol{\Sigma}_{t|t-1}\mathbf{Z}'\mathbf{F}_t^{-1}, \mathbf{L}_t = \boldsymbol{\Phi} - \mathbf{K}_t\mathbf{Z}$$

→Kalman 平滑:

$$\boldsymbol{\alpha}_{t|T} = \boldsymbol{\alpha}_{t|t-1} + \boldsymbol{\Sigma}_{t|t-1}\mathbf{q}_{t-1} \quad \boldsymbol{\Sigma}_{t|T} = \boldsymbol{\Sigma}_{t|t-1} - \boldsymbol{\Sigma}_{t|t-1}\mathbf{M}_t\boldsymbol{\Sigma}_{t|t-1}$$

$$\mathbf{q}_{t-1} = \mathbf{Z}'\mathbf{F}_t^{-1}\mathbf{v}_t + \mathbf{L}_t\mathbf{q}_t \quad , \quad \mathbf{M}_{t-1} = \mathbf{Z}'\mathbf{F}_t^{-1}\mathbf{Z} + \mathbf{L}'_t\mathbf{M}_t\mathbf{L}_t$$

$$\mathbf{q}_T = \mathbf{0} \quad \mathbf{M}_T = \mathbf{0}$$

第二部分: 过程 proc ssm

状态方程设定→**state** name (d) <options>;

→选项

模型类别

type=wn|rw|varma(p=1)|varma(p(D)=1)|...

误差项方差

Cov(value1)|<value2>

非预设模型

T(value1)<=value2>

→常数项的处理

设成误差项为 0 的随机游动

观测方程设定→**model** y=x1 x2 ... xm;

→定义成分

component 语句

→定义误差项

irregular 语句

参数设定→**parms** v1<=n1> v2<=n2> ... vk<=nk></options>;

→设定初值

var=n
 → 设定上下界
 lower=(L1 L2 ... Lk)
 upper=(U1 U2 ... Uk)

似然函数和优化方法 → **proc ssm options;**

→ 似然函数选择
 Like=diffuse|marginal|profile
 → 优化方法选择
 Optimizer(<technique=value>)

第三部分: proc ssm 应用

Arma 模型的状态空间模型表示

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} - \theta_1 a_t + a_t, a_t \sim i.i.N(0, \sigma^2), t = 1, 2, \dots, T$$

$$y_t = \phi_0 + (1, 0) \begin{pmatrix} s_{1,t} \\ s_{2,t} \end{pmatrix} \quad \text{测量方程}$$

$$\begin{pmatrix} s_{1,t} \\ s_{2,t} \end{pmatrix} = \begin{pmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{pmatrix} \begin{pmatrix} s_{1,t-1} \\ s_{2,t-1} \end{pmatrix} + \begin{pmatrix} 1 \\ -\theta_1 \end{pmatrix} a_t \quad \text{状态方程}$$

动态面板数据模型的状态空间表示

$$\begin{aligned} lsales_{t,i} &= \rho lsales_{t-1,i} + \mu_i + \zeta_t + \beta_1 lprice_{t,i} + \beta_2 lndi_{t,i} + \beta_3 lpimin_{t,i} + \epsilon_{t,i} \\ &= \rho lsales_{t-1,i} + \mu_i + \zeta_t + \epsilon_{t,i} \end{aligned}$$

$$\begin{bmatrix} lsales_{t,1} \\ lsales_{t,2} \\ \dots \\ lsales_{t,10} \end{bmatrix} = \begin{bmatrix} \alpha_{t,1} \\ \alpha_{t,2} \\ \dots \\ \alpha_{t,10} \end{bmatrix} \quad \text{测量方程}$$

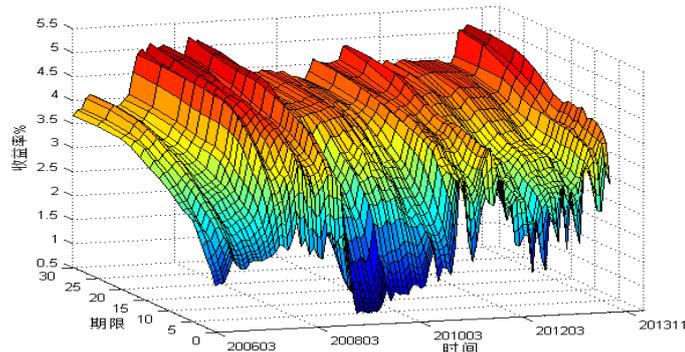
$$\begin{bmatrix} \alpha_{t,1} \\ \alpha_{t,2} \\ \dots \\ \alpha_{t,10} \end{bmatrix} = \begin{bmatrix} \rho & 0 & 0 & \dots & 0 \\ 0 & \rho & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \rho \end{bmatrix} \begin{bmatrix} \alpha_{t-1,1} \\ \alpha_{t-1,2} \\ \dots \\ \alpha_{t-1,10} \end{bmatrix} \quad \text{状态方程}$$

$$+ \begin{bmatrix} \mathbf{x}'_{t,1} & 1 & 0 & \dots & 0 \\ \mathbf{x}'_{t,2} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{x}'_{t,10} & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \beta \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_{10} \end{bmatrix} + \begin{bmatrix} \zeta_t + \epsilon_{t,1} \\ \zeta_t + \epsilon_{t,2} \\ \vdots \\ \zeta_t + \epsilon_{t,10} \end{bmatrix}$$

动态 Nelson-Siegel 利率期限结构模型

$$y_t(\tau) = L_t + \frac{1-e^{-\lambda\tau}}{\lambda\tau} S_t + \left[\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau} \right] C_t + \epsilon_t(\tau)$$

$$\begin{bmatrix} L_t - \mu_L \\ S_t - \mu_S \\ C_t - \mu_C \end{bmatrix} = \Phi \begin{bmatrix} L_{t-1} - \mu_L \\ S_{t-1} - \mu_S \\ C_{t-1} - \mu_C \end{bmatrix} + \begin{bmatrix} \eta_{L,t} \\ \eta_{S,t} \\ \eta_{C,t} \end{bmatrix}, \tau = m_1, m_2, \dots, m_N$$



讨论题、思考题和作业

讨论题：状态空间模型分析时间序列有什么优势？

思考题：（1）观测方程的常数项应如何处理？

（2）状态方程能设定为 ARMA 模型吗？设成 ARMA 模型有什么好处？

作业：第九章习题 1、2、5

课后小结

本章内容时间序列分析中的高级方法。如果学生没有对应的时间序列分析基础，学起来较为吃力。这种情况下可以少讲原理，多讲操作和应用。

其它

教案 11

授课方式	讲授+演示+上机		课次	11												
			课时	3												
教 学 内 容 (提 纲) ——第十章：矩阵编程语言 IML																
<p>1. 矩阵创建</p> <p>2. 矩阵运算和矩阵函数</p> <p>3. SAS 数据集转换为矩阵</p>																
教学目的、要求、重点和难点																
<p>目的： 了解矩阵编程语言的语法，初步掌握矩阵编程语言。</p> <p>要求：</p> <ol style="list-style-type: none"> 1. 能够通过各种方法创建简单的矩阵，熟悉矩阵基本操作。 2. 能够讲 SAS 数据集导入 IML 形成矩阵。 3. 能够编写函数并对函数实施最优化操作。 4. 能够在 IML 环境中调用 R 函数。 <p>重点：</p> <ol style="list-style-type: none"> 1. 矩阵的基本操作； 2. IML 编程语句。 <p>难点：</p> <ol style="list-style-type: none"> 1. 矩阵缩减运算中的下标运算符； 2. 函数中的符号表。 																
教学过程																
<p>第一部分：矩阵创建和操作</p> <p>IML 的操作对象→</p> <p style="padding-left: 40px;">矩阵创建</p> <p style="padding-left: 80px;">→元素列示：A={1 2,3 4,0.5 -0.6}</p> <p style="padding-left: 80px;">→生成函数：J(n,m,0), I(n), 4:19, do(v1,v2,d)</p> <p>矩阵操作→</p> <p style="padding-left: 40px;">→运算</p> <table border="1" data-bbox="272 1935 1348 2060"> <thead> <tr> <th>算符</th> <th>类型</th> <th>功能</th> <th>举例</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>前缀</td> <td>矩阵取负号</td> <td>-A</td> </tr> <tr> <td>+</td> <td>二元</td> <td>矩阵相加</td> <td>特例：A+1， 将 A 的各个元素加 1</td> </tr> </tbody> </table>					算符	类型	功能	举例	-	前缀	矩阵取负号	-A	+	二元	矩阵相加	特例：A+1， 将 A 的各个元素加 1
算符	类型	功能	举例													
-	前缀	矩阵取负号	-A													
+	二元	矩阵相加	特例：A+1， 将 A 的各个元素加 1													

-	二元	矩阵相减	特例: A-1, 将 A 的各个元素减 1
:	二元	创建指标向量	I=7:10 创建行向量 i=(7 8 9 10)
*	二元	矩阵相乘	A*B
#	二元	矩阵逐元素相乘	特例: A#3, 将 A 的各个元素乘 3
@	二元	矩阵直积	又称 Kroneck 积
**	二元	矩阵次方	A**2=A*A, A 为方阵
##	二元	逐元素次方	A##2 将 A 的每个元素平方; A##B 将 A 的每个元素按对应的 B 中元素取次方。
/	二元	逐元素相除	特例: A/2 将 A 的各个元素除以 2, 而 2/A 则是将 A 的各个元素除 2 得出的矩阵
	二元	水平拼接	将具有相同行数的两个矩阵水平拼接为一个矩阵
//	二元	垂直拼接	将具有相同列数的两个矩阵垂直拼接为一个矩阵
<>	二元	逐元素比较取大	逐元素比较两个矩阵, 结果矩阵对应元素为其中大的
><	二元	逐元素比较取小	逐元素比较两个矩阵, 结果矩阵对应元素为其中小的
&	二元	逐元素全非 0	逐元素检查两个矩阵, 如果对应元素全非 0, 结果矩阵元素为 1, 否则为 0。特例: A&1, A&0
	二元	逐元素有非 0	逐元素检查两个矩阵, 如果对应元素至少一个非 0, 结果矩阵元素为 1, 否则为 0。特例: A 1, A 0
^	前缀	逐元素检查是否 0	逐元素检查矩阵, 如果对应元素为 0, 结果矩阵元素为 1, 否则为 0。
<	二元	逐元素比较大小	逐元素比较两个矩阵, 比较关系式成立, 则结果矩阵元素为 1, 否则为 0。特例:A>2
>	二元	逐元素比较大小	逐元素比较两个矩阵, 比较关系式成立, 则结果矩阵元素为 1, 否则为 0。特例:A<2
=	二元	逐元素比较相等	逐元素比较两个矩阵, 比较关系式成立, 则结果矩阵元素为 1, 否则为 0。特例:A=2
<=	二元	逐元素比较大小	逐元素比较两个矩阵, 比较关系式成立, 则结果矩阵元素为 1, 否则为 0。特例:A<=2
>=	二元	逐元素比较大小	逐元素比较两个矩阵, 比较关系式成立, 则结果矩阵元素为 1, 否则为 0。特例:A>=2
^=	二元	逐元素比较不等	逐元素比较两个矩阵, 比较关系式成立, 则结果矩阵元素为 1, 否则为 0。特例:A^=0
`	后缀	取转置 (transpose)	A`
[]	后缀	下标算符	见后面内容

附: 广播机制

→矩阵下标和降维

→下标: []

→缩减算符

算符	+	#	##	<>	><	<:>	>:<	:
功能	相加	相乘	平方和	最大	最小	最大值位置	最小值位置	均值

矩阵线性代数函数和例程→det()、inv()、eigen()、svd()、root()

第二部分: SAS 数据集读入 IML

DATA 步 SAS 数据集→IML 矩阵

→打开数据集: use、edit

→读入 IML: read var{} into Mname

第三部分: IML 编程

以矩阵为操作单元, 与 DATA 步很多语言类似→

- 程序执行流程控制：分支、循环
- 函数
 - 函数定义：`start fname(args);... return(f); finish;`
 - 符号表 (symbol table)
 - 全局 (global)
 - 局部 (local)
- 非线性优化：`NLP*(rc,xr,"fun",x0<,options>)`

第四部分：R 函数调用

进入 R 环境：`sumit/R;... endsubmit;`

R 数据框和 IML 矩阵转换

→IML 矩阵转换为 R 数据框：

```
Call ExportMatrixToR("IMLm","Rdf")
```

→SAS 数据集转换为 R 数据框：

```
Call ExportDataSetToR("SASds","Rdf")
```

→R 数据框转换为 IML 矩阵：

```
Call ExportMatrixFromR("Rdf","IMLm")
```

→R 数据框转换为 SAS 数据集：

```
Call ExportDataSetFromR("Rdf","SASds")
```

R 函数调用

→进入 R 环境→R 语言提交→退出 R 环境

讨论题、思考题和作业

讨论题：为什么需要在 IML 环境中调用 R 函数？

思考题：(1) 元素列表创建矩阵时，变量能作为元素吗？
(2) 如何理解矩阵运算中的广播机制？

作 业：第九章习题 1、2、5、7、15、17

课后小结

本章是矩阵语言编程的内容，考虑到金融计量专业学生的编程基础和未来工作中的需要差别较大，只对基本内容进行讲解，其它高级内容进行简单介绍。

其它