

# 《小白学SAS》

- 人人都能看懂的代码和结果
- 15年SAS经验的通俗演绎
- 210分钟高清配套视频讲解

作者交流



粉丝互动



# 前言

## 为什么写这本书

在我教授研究生SAS基础课的时候，每届研究生总会问一个问题：能不能给推荐一本通俗易懂的SAS基础参考书。而每次听到这个问题，都让我难以回答。客观地说，国内关于SAS的书很多，但专门写SAS基础的书屈指可数，而真正适合那些毫无基础的SAS新手的书更是寥寥无几。当然这并不是说国内缺乏优秀的SAS基础书，只是写作风格不一定适合新手，这些书都是把SAS当作一个软件来讲，从开始介绍如何安装，然后逐一介绍各种命令、函数的语法结构。这种教材式的书对有一定基础的人来说，可能并没什么太大问题。但对于毫无基础的人来讲，却无异于经历一场噩梦，这种看不懂的痛苦只有亲身经历的人才深有体会。回想10多年前，我也是看着这种教程历经磨难一路走来，正是因为有这种亲身经历，我才不想让后来人继续体验这种痛苦，所以才有了这本书的问世。

## 本书的特点

本书与国内仅有的几本SAS基础书相比，最大的不同在于，不是以软件为导向来逐一介绍各种命令语法，而是以解决实际问题为导向，介绍如何用SAS来协助实际工作。比如，你可以从以往书中学到count函数的用法，但却不知道原来用count函数还可以分析淘宝商品的评价；你可以从以往的书中学到数组，但却不知道原来结合数组可以写出一个查找缺失值的万能程序；你可以从以往书中学到proc sgplot绘图命令，但却学不到如何利用它来综合绘制任意想要的图形；等等。

本书的目的不在于教你了解这些SAS命令或函数的基本用法，这些很简单，你只要看SAS Help，比什么教程写得都详细。本书的目的在于通过介绍这些命令和函数，让你了解隐藏在它们背后的实际用途。比如看到round函数，如果你只知道它可以用来对数值四舍五入，这是最基础的，不值一提。但你如果还能利用这种四舍五入的功能对变量进行自动分组，那才是思路的开阔。

本书另外一大特色是，是以对话形式引导读者去思考和学习。书中不少对话其实都是来源于现实中研究生和我的对话，他们也是从零开始学SAS，正如本书的主人公小白一样，都是对SAS一无所知。所以本书中小白的很多问题，相信也是现实中很多SAS小白想问的问题。只有走过的人才明白，从零开始学习一门软件语言是多么的痛苦和无助，如果再没有一本能让你看得下去的辅导

书，那更是一种折磨。所以本书采用对话形式，很大程度上是想拉近与读者的距离，让读者消除初学SAS的紧张和枯燥。

### 本书的主要内容

本书内容并非包罗万象，恰恰相反，内容很少，主要涉及三大部分：数据整理、作图和制表。对于SAS软件而言，任何一本书都不可能包括所有内容。越是包罗万象的书，有时反而越不实用，因为它们内容太多，每部分的内容都只能是蜻蜓点水。对于这种书，我只能说，你为什么不直接去看SAS Help呢？如果仔细阅读国外优秀的SAS基础书，不难发现一个特点：绝不贪多，要么不介绍，介绍的话就会很深入，一定要让你明白为止，哪怕翻来覆去地说。这看起来似乎有点傻，但却不得不令人感慨：看非母语的英文书反而比看中文书更加清晰。所以本书也是基于这一思想，凡是提到的内容，尽量用白话讲得透彻，争取将读者引领进SAS的大门，至于能不能登堂入室，还得看你自己的努力程度。

本书的主要内容包括以下几个方面：

- **建立数据集的基础和高级议题。**从最简单的建立SAS数据集开始，逐步深入，介绍如何产生新变量、如何对数据集中的变量进行修饰、如何合理地运用输入和输出格式来完成一些实际工作，等等。
- **介绍了比较实用的几个函数。**不仅是介绍函数本身，更重要的是介绍函数的使用思路，介绍如何应用函数来处理实际问题，而这正是目前国内大部分书中所缺乏的。
- **介绍了数据清洗过程。**如何查找重复值、异常值、缺失值等，这是所有数据的必需过程。本书不仅提供了这些处理过程的思路和命令，更是直接给出了一些现成的打包程序，你可以直接拿来主义，用它们来执行数据清洗。
- **SAS做图。**这是本书的一大特色。目前国内尚无如此详细介绍SAS绘图的书籍，本书花了大量篇幅介绍如何利用SAS的菜单和命令绘制各种常规和非常规的图形，相信看了本书的介绍以后，你再也不用为绘图而犯愁了。
- **SAS制表。**不仅介绍了常规的制表程序，而且介绍了很多高级的深入用法，如对表格的详细修饰，将表格直接输出为标准的三线表等。这些都是实际中非常实用的例子，不管你是公司还是事业单位，相信都会用到上这种实用的报表。
- **附录。**限于篇幅，本书有的内容没有在书中体现，但以电子版的形式作为附录提供给读者。附录主要包括两部分：一是给SAS新手看的，如常见的SAS错误提示、SAS一些高效

和低效写法的比较等。二是写给SAS高手看的，相信不少SAS高手都用过R软件，本书介绍了如何在SAS中调用R，将二者功能结合起来，更能体现出SAS的优势。

## 本书的对象和希望达到的目的

如果你跟本书的主人公一样，是一个零基础的SAS小白；如果你曾经对SAS望而却步，但却又始终抱着一个学习SAS的梦想；如果你曾经学过SAS，但始终感觉仍在门口徘徊，无法登堂入室；如果你的工作需要处理一大堆的数据，正在发愁如何提高数据管理的效率；如果你还在抱怨SAS做不出精美的图形；如果你觉得看SAS Help太吃力，想看一本解释得更加通俗的SAS基础参考书……凡此种种，你都可以拿起本书，相信它可以解决你的这些问题。

我希望本书能达到这样的目的：以往对SAS敬而远之的人，看了后会喜欢上SAS；零基础的小白，看完之后可以骄傲地宣称，我已经入门了；曾经自学过SAS基础的人，看完后会后悔没有早点看到这本书；整天跟数据打交道的人，看完后效率会成倍地提高。我希望本书能成为任何一个想学习SAS基础的人的必备案头书。

## 配套资源下载

本书的配套资源主要包括三部分：一是书中所有的程序（带有书中对应的页码），方便读者自行练习；二是9个视频，与书中内容对应，尤其是有些在书中难以详细介绍的内容，在视频中可以更生动地说明；三是给SAS新手和SAS老手的一些实用的提示（也就是前面提到的附录）。这些内容可以通过访问“知了帮”网站下载：<http://read.zhiliaobang.com/pages/article/35>

## 读者交流平台

我们给读者提供了更立体化的资源建设，不定期地给大家分享数据分析、数据挖掘、PPT等方面的精彩内容。在阅读过程中如果遇到疑难问题可以直接提问，我们会在力所能及的范围解答图书相关问题。读者也可以扫描下面的二维码予以关注。



## 致谢

本书的出版要感谢成都道然科技有限公司的姚先生，正是他的大力支持和推动，才使得本书能尽快面世。

感谢北京大学在读博士石瑀，她帮我把书中每一个程序都运行了一遍，以保证书写无误。而且还帮我指出了一些小错误，使得本书在正式出版前能够将一些错误消灭在萌芽之中。

另外，感谢我周围的朋友、同事和学生的支持，他们经常问我：这本书什么时候出版，我什么时候才能买到？每次我都回答：我正在抓紧写。他们总会很满意地接受这个答复。

虽然我们竭尽全力排除每个错误，但疏漏之处难免，欢迎读者的批评指正。

# 目 录

## 第1章 初识SAS /1

### 1.1 SAS初步印象 /2

- 1.1.1 SAS：强大的航空母舰 /2
- 1.1.2 SAS的工作环境 /4
- 1.1.3 小白认识的第一个SAS程序 /7

### 1.2 建立简单的SAS数据集 /9

- 1.2.1 如何在SAS中输入数据 /9
- 1.2.2 建立一个永久的SAS数据集 /11
- 1.2.3 如何从别的软件导入数据 /15

### 1.3 本章小结 /19

## 第2章 SAS数据集建立的高级议题 /21

### 2.1 SAS变量的输入格式 /22

- 2.1.1 数值型变量的输入格式 /22
- 2.1.2 字符型变量的输入格式 /23
- 2.1.3 日期型变量的输入格式 /24
- 2.1.4 两个特殊输入符——:和& /25

### 2.2 SAS变量的输出格式 /29

- 2.2.1 数值型变量的输出格式 /30
- 2.2.2 字符型变量的输出格式 /31
- 2.2.3 日期型变量的输出格式 /32

### 2.3 自定义输入和输出格式 /34

- 2.3.1 用informat和format自定义格式 /35
- 2.3.2 用picture照个输出模板 /38

### 2.4 如何产生新变量 /40

- 2.4.1 利用表达式或函数直接产生新变量 /40
- 2.4.2 利用if-then语句产生新变量 /42
- 2.4.3 利用retain语句和累加语句产生新变量 /43

2.4.4	利用do循环语句产生新变量	/46
2.4.5	指定新变量的类型与长度	/47
2.5	@符号在输入方式中的应用	/50
2.6	SAS函数应用技巧	/52
2.6.1	与数值计算有关的函数	/54
2.6.2	与字符有关的函数	/56
2.6.3	与日期和时间有关的函数	/69
2.6.4	与变量类型转换有关的函数	/74
2.6.5	与概率和分布有关的函数	/78
2.6.6	dif和lag函数	/82
2.7	本章小结	/84
第3章	SAS数据清洗和加工	/85
3.1	数据合并	/88
3.1.1	利用set语句进行纵向合并	/88
3.1.2	利用merge语句进行横向合并	/94
3.2	数据对比	/97
3.3	数据清洗——查找和删除重复值	/100
3.3.1	查找和删除重复值	/100
3.3.2	扩展内容：first.变量和last.变量	/103
3.4	数据清洗——查找缺失值	/105
3.4.1	补充内容：数组	/106
3.4.2	补充内容：自动变量	/110
3.4.3	超值礼包：查找缺失值的万能程序	/112
3.5	数据清洗——查找异常值	/113
3.5.1	补充内容：if和where的区别	/113
3.5.2	超值礼包：查找异常值的万能程序	/115
3.6	缺失值的填补	/119
3.6.1	缺失数据的填补	/119
3.6.2	缺失数据的更新	/122
3.6.3	扩展内容：常见SAS语句及数据集选项	/124
3.7	产生数据子集	/128

- 3.7.1 产生特定记录的子集 /128
- 3.7.2 生成变量子集 /130

### 3.8 本章小结 /132

## 第4章 SAS与数据可视化 /133

### 4.1 用菜单绘制统计图 /134

- 4.1.1 绘制数据的分布图 /138
- 4.1.2 绘制箱式图 /143
- 4.1.3 绘制散点图和回归线图 /146
- 4.1.4 绘制序列图、带状图和阶梯图 /151
- 4.1.5 绘制金字塔图/旋风图/蝴蝶图 /158

### 4.2 用proc sgplot绘制不一样的图 /162

- 4.2.1 绘制几个常规图 /168
- 4.2.2 绘制双坐标轴图 /171
- 4.2.3 绘制多样的柱状图 /174
- 4.2.4 绘制瀑布图 /179
- 4.2.5 绘制气泡图 /181
- 4.2.6 绘制风险图 /184
- 4.2.7 绘制自己的卡通图 /187
- 4.2.8 绘制森林图 /190

### 4.3 用proc template制作绘图模板 /194

- 4.3.1 画一个绚丽的饼图 /197
- 4.3.2 马赛克图 /200
- 4.3.3 截断坐标轴图 /203
- 4.3.4 三维直方图 /206
- 4.3.5 热图 /208

### 4.4 用proc gradar绘制雷达图 /209

- 4.4.1 绘制雷达图 /210
- 4.4.2 绘制日历图 /213

### 4.5 用proc gmap绘制地图 /215

- 4.5.1 利用SAS自带地图文件绘制中国地图 /218
- 4.5.2 利用SAS自带地图文件绘制省地图 /221



## >> 小白学SAS

4.5.3 利用shape文件绘制中国地图 /225

4.5.4 利用shape文件绘制地区地图 /227

### 4.6 本章小结 /230

## 第5章 SAS与表格展示 /231

### 5.1 用tabulate过程制表 /231

5.1.1 制作三维列联表 /233

5.1.2 表格修饰 /235

5.1.3 生成定量资料的描述表 /240

5.1.4 制作描述多变量的统计表 /242

5.1.5 制作标准的三线表 /246

### 5.2 用report过程制作统计报表 /249

5.2.1 列表显示符合条件的观测 /251

5.2.2 分组显示统计量 /258

5.2.3 生成列联表 /262

### 5.3 用ODS系统传送结果 /264

5.3.1 把结果输出到Word、Excel或PDF /265

5.3.2 用template制作表格模板 /266

### 5.4 本章小结 /272

# 第 1 章

## 初识SAS

话说小白已经新入职一周了，在这一周中，令她最郁闷的一件事就是公司全都在用SAS处理数据，而这恰恰是她的薄弱环节。虽然她的简历上写着“熟悉SAS软件”，而且在面试时也信誓旦旦地说会用SAS，但只有她自己心里清楚她所说的“会”到底是什么意思。事实上，她也就是知道SAS软件打开是个什么样子。俗话说“先下手为强”，只管先拿到工作再说，不会的抓紧时间学习。

幸亏小白运气好，遇到了她生命中的一个贵人。与她同在一个办公室的Ms.周不仅是SAS高手，而且乐于助人。其实在面试的时候Ms.周作为技术人员就发现小白可能不懂SAS，但被她这种“即使不懂也敢应聘”的劲头打动，认为她还是值得培养的。因此在简单地接触一周后，当小白提出要拜他为师时，他也就欣然接受了这个对SAS犹如一张白纸的弟子。

今天是小白跟Ms.周学习SAS的第一天，心情不免有些紧张，生怕自己学不会。Ms.周看出小白的心情，于是先跟她聊了一些轻松的话题，以放松她的心情。

**Ms.周：**以前你接触过SAS吗？

**小白：**曾经有一段时间想自学一下，也买了本书，结果发现几乎看不懂啊，只好放弃了。还是师兄师姐们说得对，SAS太难学了。我师兄师姐他们都不用SAS，全都用SPSS。

**Ms.周想了一下：**那我问你一个问题，假如你患了某种疾病，你会咨询我还是咨询医生？

**小白随口答道：**肯定会咨询医生啦，他们对疾病更熟悉，你对疾病不熟悉。

**Ms.周：**既然如此，那如果你想知道SAS是否难学，应该问熟悉SAS的人还是问不熟悉SAS的人呢？

**小白一怔：**是啊，我以前还真没想过这个问题。我师兄师姐他们也没用过SAS，我为什么要咨

## >> 小白学SAS

询他们呢？

**Ms.周：**其实我碰到过很多像你这样的人，他们都没接触过SAS，却一致认为SAS很难学，这让我很奇怪，既然没有用过SAS，怎么会知道难不难呢？所以有时候不能轻信传言，一定要自己试过才知道。

**小白：**那SAS到底难不难呢？

**Ms.周：**其实SAS并不难学，为什么有人觉得SAS难学呢，我想主要有下面几个原因。

一是他们天生对编程有恐惧心理，一听SAS要编程，立刻就认为很难学，好像编程有多可怕一样。其实SAS的编程是有固定格式的，和菜单差不多，并不像有的人想象的那么困难。

二是教SAS的老师讲的方式不对，像江南七怪教郭靖一样，教而不明其法，学而不得其道。不少SAS教师不结合实际，空讲编程理论，自然会让人产生厌烦心理。

三是市面上关于SAS的书大多写得比较空洞，甚至有的编者本身就缺乏实践经验，只空讲一些理论，根本不知读者需要的是什麼，实践性比较差，很难让人真正看明白。

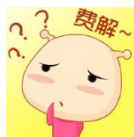
四是像你这种情况，对道听途说的传言信以为真，认为别人学不会的自己也学不会，对自己缺乏信心，而且违背了毛主席教导我们的“实践出真知”的理论。

**小白满脸通红：**我就是第四种人。那你觉得我能学会SAS吗？

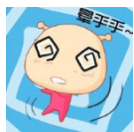
**Ms.周：**从今天起，我就教你来学习SAS应用，只要我讲的时候你仔细听，我敢保证你会很轻松地掌握SAS。



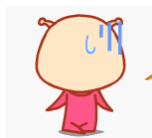
编程? 太可怕了



费解~  
不知道他在讲什么



好晕啊



这么难! 还是放弃吧

## 1.1 SAS初步印象

### 1.1.1 SAS：强大的航空母舰

**Ms.周：**在讲SAS编程之前，我先简单说一下SAS的功能，让你知道SAS有多么强大。

SAS的全称是Statistical Analysis System，从名字就能看出，它不仅仅是个软件，而且是一个包含各种模块的系统，所以有人称SAS是统计软件中的航空母舰。它的功能十分强大，全球有超过135个国家的65 000多家客户都在采用SAS解决方案，其中包括2012年《财富》全球500强企业前100家企业中的90家。正因为SAS是个系统，所以它不只是个统计软件，而是多个产品的集合，表1.1列出了几个非常实用的SAS模块。

表1.1 常见的几个SAS模块及简介

SAS模块	功能
Base SAS	提供强大的数据管理和描述功能，不管你对数据有什么操作要求，SAS都可以帮你实现
SAS/STAT	统计分析模块，提供上百种分析方法，从简单的统计描述到复杂的模型分析，SAS都可以完成
SAS/GRAPH	绘图模块，在以前版本中，SAS的绘图功能一直是弱项。但从SAS 9.3开始提供了丰富的绘图功能，可以绘制各种与任何软件媲美的专业图形
SAS/ETS	计量经济学和时间序列分析的模块，你想预测股市的涨跌吗？你想知道公司几年以后的发展趋势吗？那就需要学习掌握这个模块
SAS/QC	质量控制模块，如果企业想对产品进行最优化设计、可靠性分析、质量控制分析，这是必不可少的得力工具
SAS/EM	SAS的数据挖掘模块，这是大数据时代必须掌握的工具，像决策树、神经网络等听起来似乎很先进的东西都在这里，还有什么可犹豫的呢
SAS/GIS	地理信息系统模块，如果你想绘制地图或者对不同地区的公司数据进行分析，这个模块的漂亮结果会让别人对你刮目相看
SAS/GENETICS	遗传分析模块，听起来很专业，好像只适用于生物医学科研人员，但鉴于目前不少公司对基因的关注，这一模块也注定会更火

除了表1.1中的模块，其实还有很多，只不过可能用得较少，比如SAS/OR是与运筹学有关的模块，SAS/IML可以实现一些简单的矩阵运算，SAS/INSIGHT可以通过菜单方式进行数据探索，SAS/AF是为高手提供的自行开发程序的平台等。

**小白兴奋地说：**哇，原来SAS能做这么多事情啊，我一开始还以为它就能做个统计分析呢。听你介绍完，我还真的动心了，一定要跟你好好学SAS。

**Ms.周：**看你这么激动，那我就趁热打铁，顺便给你介绍一下SAS的基本构成吧。

## 1.1.2 SAS的工作环境

**Ms.周：** SAS 9.4中文版打开后的界面是下面这个样子，如图1.1所示。

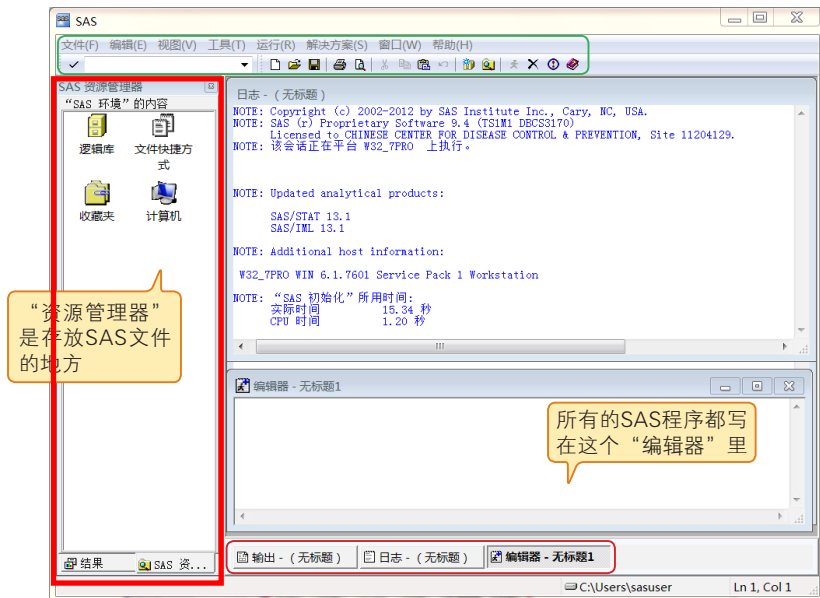



图1.1 SAS 9.4中文版启动界面

看起来好像有点眼花缭乱，其实主要就三个地方，绿色框起来的部分是菜单和工具栏，紫色框起来的是资源管理器，红色框起来的是编辑器、日志和输出窗口。


SAS是以编程为主，所以最主要的窗口是“编辑器”窗口，所有的SAS程序都在这个窗口书写，写好后单击工具栏的  就开始执行操作。默认执行窗口中的所有程序，也可选中部分语句，执行部分程序。程序输出的结果显示在“输出”窗口，而“日志”窗口则是记录程序运行过程中的一些信息，如果程序有错误，就会在这个窗口给出提示。

左边的“资源管理器”主要是SAS数据集的存放，我们用到的数据都会被存放到“逻辑库”里，可以随时调用。这里你先知道它的大致作用就可以了，后面我还会详细说明。

菜单和工具栏在SAS中其实用处不大，SAS用到菜单的地方不多，绝大多数情况下通过在编辑器窗口写程序就可以了。表1.2中简单列了各个菜单的功能。

表1.2 SAS中各个菜单的主要功能

菜单	功能
文件	SAS文件的打开、关闭、保存以及数据导入、数据导出等
编辑	撤销、剪切、复制、粘贴、选定、清空、查找、替换等
视图	可切换不同窗口，如编辑窗口、图形窗口、日志窗口、结果输出窗口等
工具	图形、报表等的编辑，以及对SAS的一些系统设置
运行	可指定提交写好的部分或全部程序
解决方案	提供了不少基于菜单式的分析，如数据挖掘、地理信息系统、交互数据分析等
窗口	主要用于各个窗口的排列、大小调整等
帮助	提供了SAS中各个模块所有命令的帮助，而且带有很多实例。你有任何不明白的地方，都可以在这里寻找答案。可以说，这是最全的SAS百科全书

工具栏小人图标  用来运行SAS程序，单击这个图标便开始执行SAS命令，输出结果。

**小白：**那我是不是可以这么理解，SAS的绝大多数功能都可以在编辑器窗口通过编程来完成，是吗？

**Ms.周：**没错，绝大多数的数据管理、统计分析等都是在编辑器窗口写程序来实现的，菜单主要是辅助功能，帮助实现一些简单的，如保存、剪切、复制等功能。所以我们重点都是讲如何在编辑器窗口编写程序。现在我写一个简单的SAS程序，示范一下它是如何运行的。

图1.2是在编辑器窗口写了一段SAS程序，它的作用是求男性和女性的平均考试成绩。现在你先用不管具体程序的含义，只要知道这是在编辑器窗口写的程序就可以了。

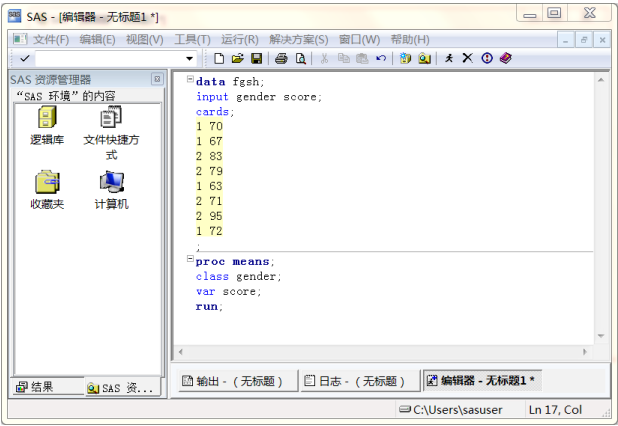



图1.2 编辑器窗口的SAS程序

>> 小白学SAS

图1.2所示的程序写完后，单击工具栏的  提交运行，然后输出窗口会显示运行的结果，如图1.3所示。

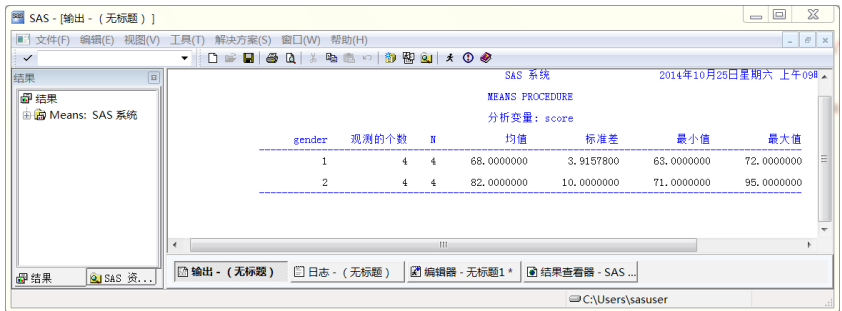


图1.3 输出窗口的SAS程序运行结果

同时，日志窗口也显示了一些程序运行的信息，如图1.4所示。

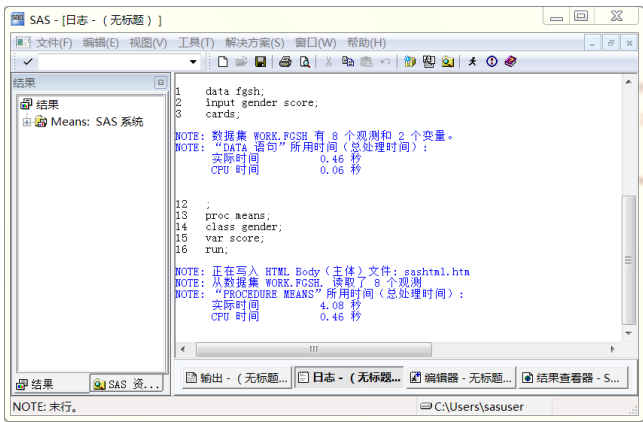
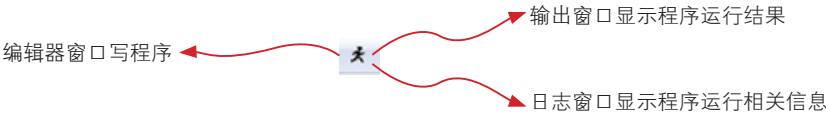


图1.4 日志窗口的SAS程序运行信息

图1.2至图1.4就是SAS程序运行及结果输出的过程，没有用到任何菜单操作，所以你只要把程序在编辑器窗口写好，提交运行，然后看结果就行了。即使你的程序有错误，日志窗口也会提醒你错误之处，你可以根据这些提示进行修改。

**小白：**我大概了解了，是不是就像下面我画的这个流程？



**Ms.周:** 画得不错，基本就是这一过程，至于SAS程序怎么写，就需要慢慢来学了。

**Ms.周:** 另外，你仔细看一下图1.3的运行结果，在下方还有一个“结果查看器”窗口，它与输出窗口的结果是完全一样的，但是显示得更漂亮一些，如图1.5所示。

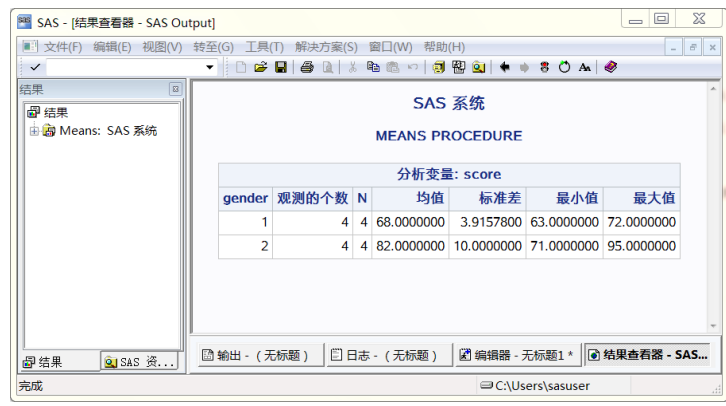


图1.5 结果查看器窗口的SAS程序运行结果

**小白:** 这个结果确实感觉整齐多了。

**Ms.周:** 不过输出这个窗口会增加运行的时间，如果不想自动产生这个窗口，可以在“工具”→“选项”→“参数选择”→“结果”的界面中，将“创建HTML(H)”前面小框中的“√”勾掉，此时SAS会将结果仅显示在输出窗口中。

**小白:** 我还是保留这个窗口吧，这个可比输出窗口的结果美观多了。

**Ms.周:** 如果你想把写好的程序保存到电脑上，先把编辑器窗口作为当前窗口，然后通过“文件”→“另存为”，保存到自己指定的位置就可以了。

**小白:** 那如果想保存输出结果，是不是把输出窗口作为当前窗口，然后另存为就可以了？

**Ms.周:** 对。结果可以保存为SAS格式，也可以选择把它保存为rtf文件，这样可以直接用Word打开了，方便查看。

## 1.1.3 小白认识的第一个SAS程序

**Ms.周:** 你已经知道SAS是怎么运行输出结果的了，现在我跟你说一下SAS程序的基本构成，让你先有一个大致的了解。你先看一下图1.6中的SAS程序，有什么特点？



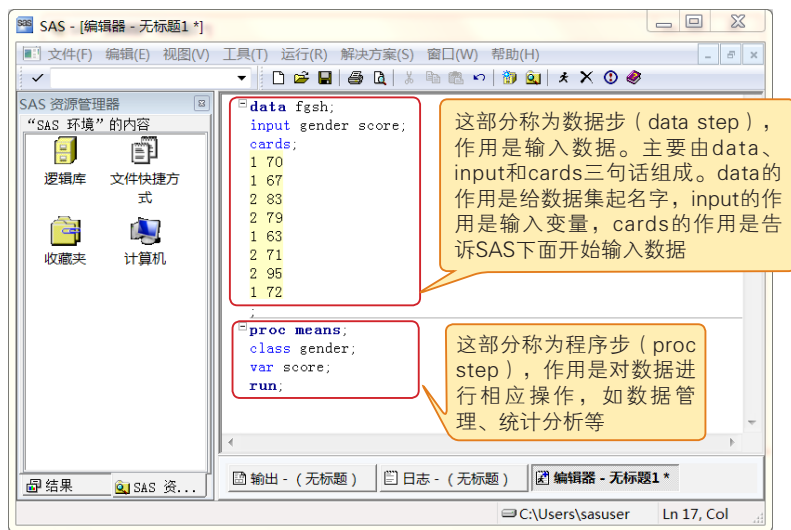


图1.6 SAS程序的基本构成

**小白仔细看了一下：**首先，整个程序分成了两部分，中间有一道灰色的分隔线；第二，不同的单词用不同的颜色标示。我就能看出这两点。

**Ms.周：**你观察得很仔细，这段程序虽然简单，但麻雀虽小，五脏俱全，可以看出SAS程序的基本构成。所有SAS程序都是由两部分构成，一是数据步，二是程序步。顾名思义，数据步就是输入数据的，在SAS中输入数据基本上都是由data、input和cards这三句话组成。程序步是运行程序执行相应操作的，程序步都是由proc开始，后面跟着相应的命令语句，不同的命令语句执行相应的操作，如proc sort是对数据做排序，proc sgplot过程可以作图，等等。

**Ms.周：**再说这些颜色。SAS中用不同颜色表示相应的语句，蓝色的都是SAS固定的格式，不能更改，比如cards，必须这么写，少一个字母也不行。而黑色的是你根据自己的需要来写的，比如fgsh，你也可以改为abc等其他字母。

我想你也注意到了，data和proc means都是深蓝色，它们分别代表了数据步和程序步的开始，run也是深蓝色，这是程序写完后必加的一句话，告诉SAS程序已经写完了，可以run了。而input、class等浅蓝色的字，则是作为一句话并且包含在相应的数据步或程序步中了。SAS中对不同语句冠以不同颜色，是有道理的。**如果某个语句写错了，它就会显示为红色，那你就很容易发现写错了。**

**小白：**哦，是这么回事啊，这一点对我这种菜鸟应该很有帮助。

## 1.2 建立简单的SAS数据集

### 1.2.1 如何在SAS中输入数据

**Ms.周:** 俗话说，巧妇难为无米之炊。SAS所有的分析都是基于数据进行的，所以学SAS首先就要学如何建立数据集，也就是数据步。这一步学会后，再学习程序步，了解如何对数据进行分析处理。

**Ms.周:** SAS中建立数据集主要有两种方式：一种是像刚才那样，在SAS中直接输入数据，另一种是把数据从别的数据库软件导入到SAS中。如果数据比较少，一般直接输入就可以了，但如果数据量很大，我们通常会先把数据输入到其他专业数据录入软件中，如Excel、Access、Oracle等，然后再导入到SAS中。因为SAS不是专业的数据录入软件，大量的数据录入并不是很方便。

**Ms.周:** 我们先说第一种方式，在SAS编辑器窗口直接建立数据集，就像刚才提到的，一共就三句话：data、input和cards。

#### 1. 用data语句起名字

data语句是给数据集起名字，这是建立数据集的开始。data语句的基本格式是：

`data 数据集名;`

data和数据集名之间用空格隔开。你可以起一个自己喜欢的数据集名，不过要遵循几个简单的规则：只能由**英文**、**数字**和**下画线**组成，而且第一个词不能是数字；不能超过32个字符；不能包含空格、中文和特殊字符。

**小白:** 特殊字符都是哪些呢？

**Ms.周:** 你问得很仔细，这里的特殊字符主要是指数字键上方的那些字符，如图1.7所示。如果不确定哪些是特殊字符，那就只写字母好了，只要不超过32个字符，肯定没事。



图1.7 SAS命名不能用的几个特殊字符

#### 2. 用input语句输入变量

输入变量用input语句，input语句的基本格式是：

`input 变量1 变量2 .....;`

## >> 小白学SAS

各个变量之间用空格隔开，SAS默认是以空格作为分隔符来区分不同变量的，也就是说，当SAS读到空格时，会认为当前变量已经读完了，后面就是下一个变量了。变量的名字与SAS数据集名的规则基本一样，保险起见最好就是用英文字母命名（事实上，如果你在SAS中加一句`options validvarname=any;`，也可以使SAS识别中文名，但我不建议你这么做）。至于input后面写多少个变量，根据你的数据来定。通常情况下不用担心变量数的问题，即使你有上万个变量，SAS依然可以处理。

### 3. cards语句引领数据输入

输入数据用cards或datalines均可。它的基本格式是：

```
cards;  
数据  
;
```

数据是与变量相对应的，每行的多个数据之间最好用空格隔开。数据可以是数值，也可以是日期或数值以外的字符（包括中文）。旧版本SAS一直用cards，因为以前的机器都是打卡式的输入，后来为了意义明确，新增了datalines，但cards仍然可以用。我因为学得比较早，所以习惯用cards，你可以自己选择，这纯粹是个人习惯。

**小白做了个鬼脸：**既然老师都用cards，那我也用cards吧，而且还可以少写几个字母。对了，我看每一个SAS语句后面都跟着个分号，这是必需的吗？

**Ms.周：**这个问题问得好。在SAS中，分号是一句话结束的标志，一句话结束后，一定要跟着一个英文状态的分号。像data、input、cards都是一句话，所以后面一定要跟着分号。但如果是数据输入结束，分号必须另起一行写，不能直接跟在数据后面。

**小白：**明白了，这些我一定注意。

**Ms.周：**那我现在考考你，你在SAS中建立一个数据集，起名为first，变量包括我们两个人的性别和年龄，性别男和女分别用1和0表示。看看你掌握了多少。

**小白想了一阵，写出了她的第一个SAS程序：**

```
data first;  
input gender age;  
cards;  
1 30  
0 24  
;
```

**Ms.周:** 写得不错，可以打100分了。你看，你已经可以自己建立数据集了，看来很快就可以出师了。

**小白谦虚地说:** 主要是老师教得好。看来SAS的确不是很难啊，这下我有信心继续往下学了。

## 1.2.2 建立一个永久的SAS数据集

**Ms.周:** 那我再问你一个问题，这些数据集建立后都放到什么地方了？

**小白一下语塞了:** 这个……

**Ms.周:** 这个你不会是很正常的，我正要跟你讲呢。我们前面提到过，SAS工作环境中有一个资源管理器，这里面的逻辑库就是存放SAS文件的地方，所有建立的数据集都放到这里了。双击打开逻辑库，可以看到6个文件夹，其中有一个名为Work，你刚才建立的数据集就是放到这里面了，如图1.8所示。

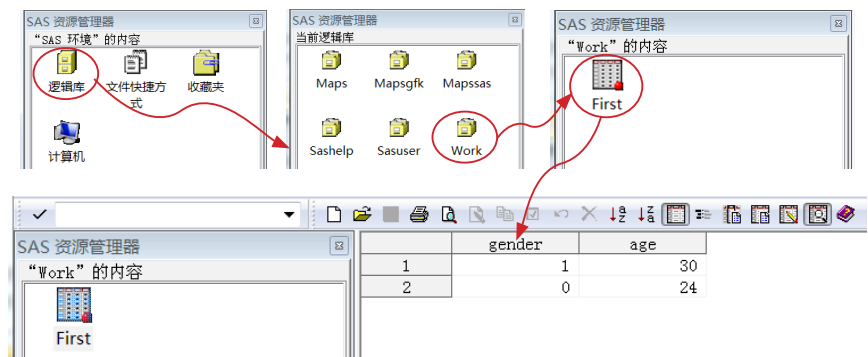


图1.8 SAS数据集的存放示意图

**小白:** 为什么存到Work，而不是其他文件夹中呢？

**Ms.周:** 逻辑库中的6个文件夹除了Work外，其他都已经存放了SAS自身的一些数据，比如Maps、Mapsgfk及Mapssas存放的是一些与地图有关的数据，Sashelp存放的是一些帮助文件，sasuser存放了一些供你练习的数据。Work文件夹是个空文件夹，是SAS留给用户存放数据用的。正常情况下，SAS中建立的数据集都默认存放到这个文件夹下。但这个文件夹有个问题，一旦你关闭SAS，文件夹就自动清空，里面存放的数据集就全都没了。**所以一般将建立在Work文件夹下的数据称为临时数据集。**

**小白立刻着急了:** 那我如果下次还想用怎么办？

**Ms.周:** 很简单, 把数据存放到别的文件夹就可以了, 其他文件夹里的内容不会因为SAS的退出而清空, **建立在非Work文件夹下的数据集一般称为永久数据集**。只要你在数据集名字前面加上相应文件夹的前缀, 就相当于告诉SAS把数据集存放到这个文件夹下, 如:

```
data sasuser.first;  
input gender age;  
cards;  
1 30  
0 24  
;
```

**Ms.周:** 运行后, 数据集first就存放到sasuser文件夹下, 而不是work文件夹下了, 此时first就是个永久数据集。

**小白:** 我明白了, 那我如果把data sasuser.first;改成data sashelp.first;, 那first数据集就存放到sashelp这个文件夹里了?

**Ms.周:** 聪明。而且你也可以自己新建一个文件夹, 把数据集都放到这个文件夹里。

**小白:** 怎么在逻辑库建立自己的文件夹呢?

**Ms.周:** 这个一共需要两步。第一, 先在自己的电脑中找个位置建立一个文件夹, 文件夹的名字随意取, 中英文均可。不过低版本的SAS不支持中文, 所以最好是英文名。比如我们在G盘新建一文件夹命名为excel。

第二, 在SAS资源管理器中打开逻辑库, 单击右键→“新建”, 出现“新建逻辑库”窗口(图1.9)。名称填写你喜欢的文件夹名字(如fgs), 引擎默认即可。然后在“路径”栏后单击“浏览”按钮, 在出现的窗口找到你在电脑上建立的文件夹, 然后单击“确定”按钮。此时逻辑库就新增了一个你自己建立的文件夹。

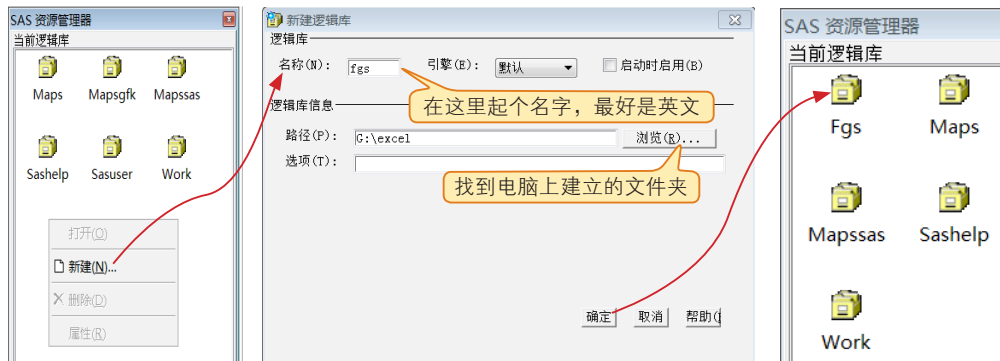


图1.9 新建逻辑库的界面

**Ms.周:** 如图1.9所示新建逻辑库的过程也可以通过SAS语句来实现。非常简单，就是一句话：

`libname 逻辑库文件夹名 硬盘的文件夹路径和名称;`

如图1.9所示的过程写成SAS语句就是：

```
libname fgs "g:\excel";
```

它的意思是，在SAS逻辑库中建立一个名为“fgs”的文件夹，并将它与电脑上的“G:/Excel”文件夹关联起来。（当然Excel文件夹已经提前建立好了。）

**小白:** 看起来好像也不难，我就记住libname就行了。

**Ms.周:** 相对而言，其实SAS语句比菜单操作还要简单，用习惯就好了。

**小白:** 建立了文件夹后，那我该如何往这个文件夹中添加SAS数据集呢？

**Ms.周:** 只要记住一点，**在data语句中加上文件夹名作为数据集的前缀就可以了。**如：

```
data fgs.first;  
input gender age;  
cards;  
1 30  
0 24  
;
```

**Ms.周:** 这段简单的程序就是把first数据集添加到了fgs文件夹中，也就是添加到G盘excel文件中。

**Ms.周:** 有一点要记住，**只有逻辑库中已经有有了一个文件夹，才能在data语句的数据集名前面加上这个文件夹名字作为前缀。**逻辑库中如果没有fgs这个文件夹，那你就不能直接写data fgs.first;，因为SAS找不到fgs这个文件夹，就不知道该把数据存到哪儿。

**小白:** 那我在用data语句前，必须先用菜单或libname语句建好文件夹的关联是吧？

**Ms.周:** 对，用语句的话，一定要在建立数据集之前就用libname语句写好这种关联。

**小白:** 在SAS逻辑库中新建文件夹之前，为什么必须得先在电脑上建立一个文件夹呢？

**Ms.周:** 因为SAS只是个软件，它所产生的所有数据集需要在电脑里有个地方存放，像sasuser、sashelp等里面的数据都是在SAS安装目录下。所以如果你在逻辑库里新建一个文件夹，也必须关联一个具体的硬盘位置，这样所有的数据集都可以到这个位置去找。SAS逻辑库里的文件夹只是一个形式，跟它关联的电脑上文件夹的位置才是放数据集的地方，就像哆啦A梦的任意门一样，虽然是在哆啦A梦的家里开门，但却连接到了另一个地方。你在SAS逻辑库打开文件夹，其实是连接到了电脑上对应的文件夹。

**小白:** 你一说哆啦A梦的任意门我就明白了。也就是说，我们所建立的SAS数据集都放到硬盘

## >> 小白学SAS

上对应的位置了，SAS逻辑库里自带的文件夹里的数据集都在SAS的安装目录下，而我们自己建的文件夹里的数据集就在我们自己指定的目录下。

**Ms.周：**完全正确。看来你的理解力还是很不错的，很多人学了很久都始终不明白逻辑库是干什么的，没想到你一下就理解了。

**小白：**哈哈，主要是因为你提到了哆啦A梦，这是我的最爱，我立刻就有兴趣了。

**Ms.周：**还有一种非常简单的建立永久数据集的方式，就是直接用data语句指定一个路径和SAS数据集名。如：

```
data "g:\excel\first";
input gender age;
cards;
1 30
0 24
;
```

**Ms.周：**这相当于在G盘的Excel文件夹中生成一个名为first的SAS数据集，当然前提还是Excel文件夹必须已经存在于G盘中了。同时在SAS的逻辑库中会自动生成一个名为“Wc000001”的文件夹，文件夹下也有first这一SAS数据集。

**小白：**这种方式好像比刚才说的方法更简单啊，也很容易记住。

**Ms.周：**这种方式的好处是简单，不好的地方是逻辑库中的文件夹是SAS自动生成的，名字也很奇怪，不像自己建立的文件夹可以随心所欲地改为自己想要的名字。总之，自己根据喜好选择吧。

**小白：**建立了永久数据集，以后我们任何时候都可以直接调用是吗？

**Ms.周：**是的，只要你不删除，无论什么时候打开SAS，都可以调用。但对于自己建立的逻辑库，在调用数据集之前还需要用libname语句把逻辑库与文件夹建立关联。SAS中所有以proc开头的过程都有一个选项，即data=，这个选项的作用是指定对哪个数据集进行相应的操作。如果有了永久数据集，我们就没有必要每次录入数据了，而是直接用data=选项指定数据集名调用就行了。比如我们用proc print过程输出数据集fgs.first，就可以写为：

```
libname fgs "g:\excel";           /*调用之前仍需先用libname建立关联*/
proc print data=fgs.first;
run;
```

**Ms.周：**如果是用data "g:\excel\first";语句建立的永久数据集，则调用的程序为：

```
proc print data="g:\excel\ first ";
run;
```

1.2.3 如何从别的软件导入数据

**Ms.周:** 刚才说的是自己输入数据，我们再来说一下如何从别的地方把数据导入到SAS中。SAS的数据导入可以通过菜单，也可以自己写程序实现，我推荐你使用菜单，既快捷又方便，而且可以自动生成SAS数据导入的编程语句。

**小白:** 太好了，我现在还没编程基础，还担心听不懂呢。

**Ms.周:** 首先单击“文件”→“导入数据”，出现数据导入的窗口（图1.10）。这里有很多数据库类型可以选择，包括常见的Excel、Access、DBF等数据库以及SPSS、Stata等统计软件。

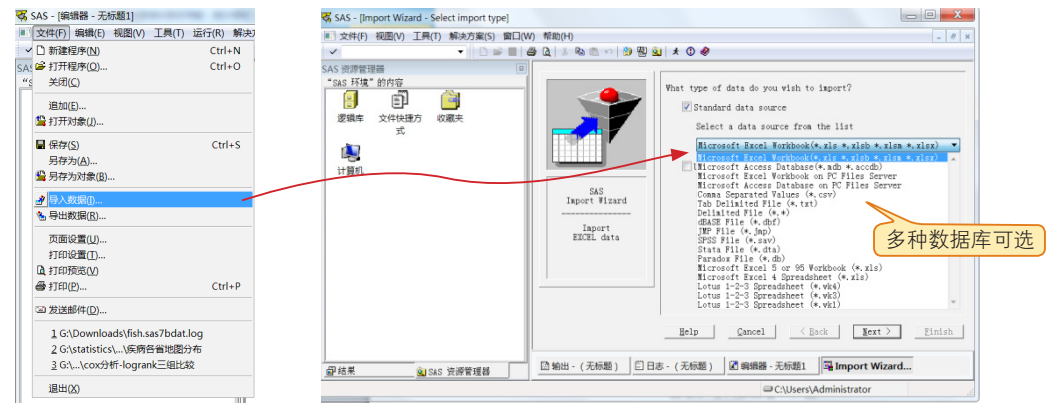


图1.10 导入数据窗口界面

先选择Excel格式，然后单击“Next”按钮，进入到“Connect to MS Excel”界面（图1.11）。

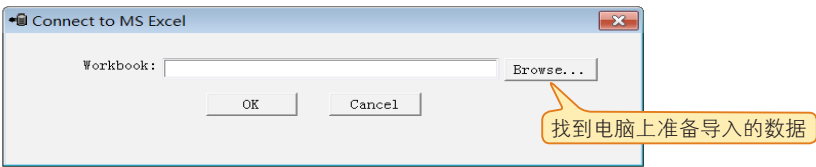


图1.11 导入Excel的界面

在图1.11中单击“Browse”按钮，找到准备导入的Excel文件，然后返回到图1.12所示界面。

在图1.12中，单击“OK”按钮，进入到更详细的选择步骤（图1.13）。由于Excel中包含多个sheet，此时你可以通过下拉菜单选择相应的Sheet。在exam数据集中，数据是在Sheet1中，因此选择默认的“Sheet1\$”即可。



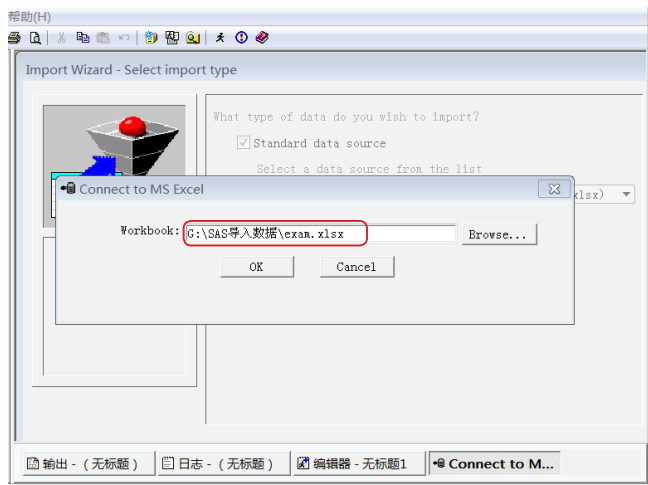


图1.12 含数据集名称的Excel导入的界面

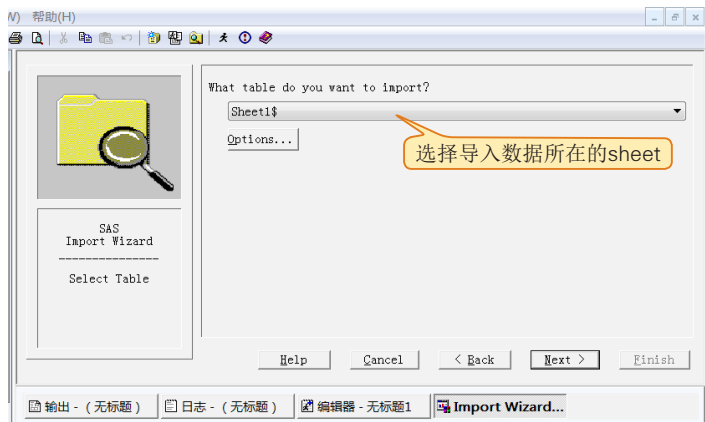


图1.13 详细选择Excel数据的界面

如果要定义其他选项，可以单击“Options”按钮，打开选项界面（图1.14）。

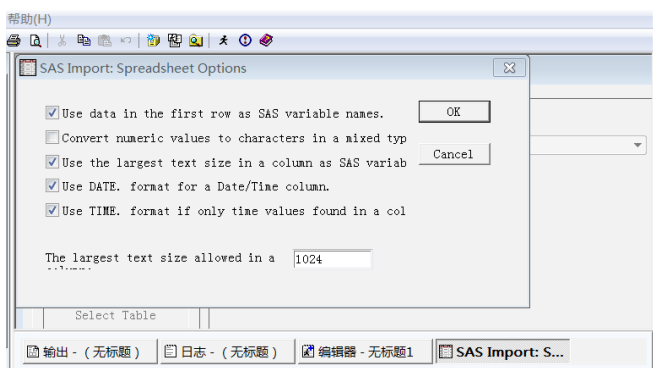


图1.14 导入数据的选项界面

选项界面中这6行选项的意思分别是：

（1）以Excel数据中的第一行作为变量名。如果你的Excel数据第一行不是变量，而是数据，就不要勾选这个。

（2）如果Excel中有的变量既有数值型又有字符型（这在Excel中并不少见），会自动将数值型转换为字符型。因为SAS是不允许一个变量同时有两种类型的。如果不勾选该选项，SAS将根据该变量前8行中占多数的那一类变量作为该变量的类型，如果前8行中数值型占多数，SAS就会认为该变量是数值型，字符型变量就被作为缺失值。如果前8行中字符型占多数，SAS就认为该变量是字符型，数值全部作为文本字符来对待。

（3）Excel中的字符型变量，用字节最多的那条记录的宽度作为该变量的宽度值。

（4）Excel中的日期变量，在SAS中定义为日期格式。

（5）Excel中的时间变量，在SAS中定义为时间格式。

（6）默认变量的最大宽度为1024个字节，这里可以改为小于32 767的其他数值。

一切选项搞定后，单击如图1.13所示的“Next”按钮，进入到下一步，选择存放位置（图1.15）。存放的默认文件夹是work文件夹，你也可以通过单击Library下拉菜单选择其他文件夹，如sasuser或你自己已经建立的文件夹等。在Member下拉文本框处输入SAS数据集名，给导入的数据集起个名字，这里随便给它命名为ex。

到这一步可以直接单击“Finish”按钮，整个数据导入过程就结束了。然后你会看到work文件夹下多了一个名为ex的数据集。

如果你想把整个操作过程让SAS用语句的形式记录下来，那就单击“Next”按钮，进入到下一

## >> 小白学SAS

步，选择合适的目录把数据导入SAS的过程保存下来（图1.16）。你可以单击“Browse”按钮，选择电脑上的一个文件夹，并给程序起个名字，把程序保存到电脑上。比如我选择保存到G盘根目录下，并给它起名为aa，然后单击“Finish”按钮，这时你会发现G盘根目录下多了一个名为“aa.sas”的SAS文件。

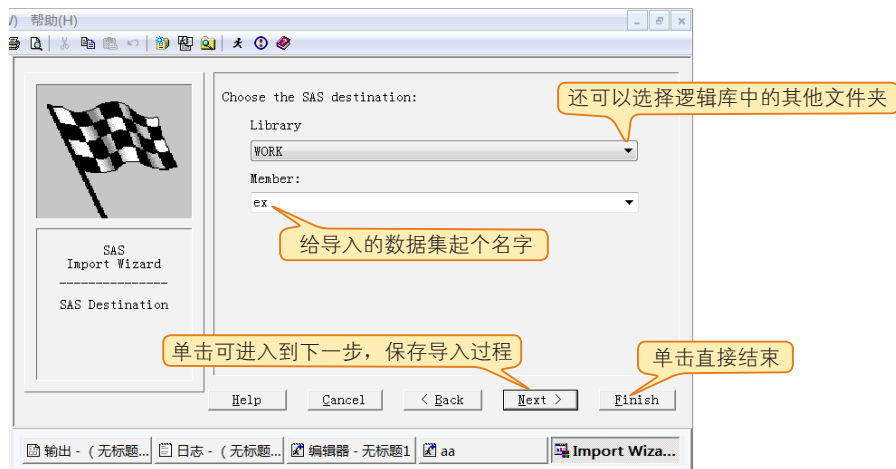


图1.15 数据集存放位置及名称的界面

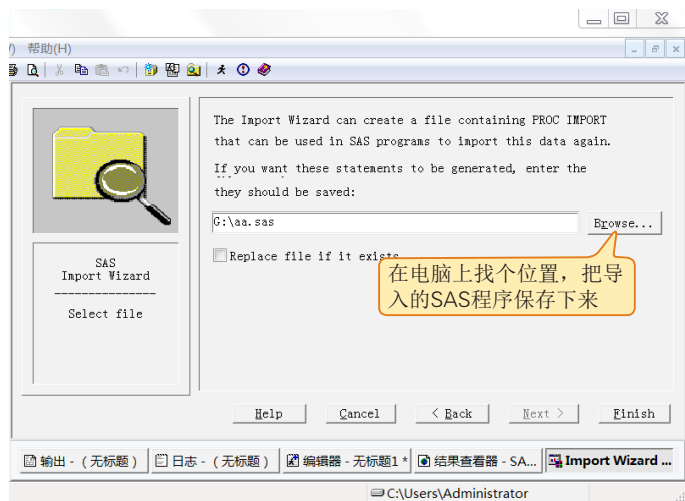


图1.16 SAS语句的保存界面

打开aa.sas这个文件，可以看到这实际上就是SAS数据导入的语句过程（图1.17）。这些语句与菜单导入是一一对应的，比如out=相当于如图1.14所示的作用；datafile=相当于如图1.11和图1.12所示的作用；dbms=相当于如图1.10所示的作用，replace表示SAS中如果已有同名数据集，就覆盖它；range=相当于如图1.13所示的作用；range下面的5行相当于如图1.13所示中默认的Options选项。

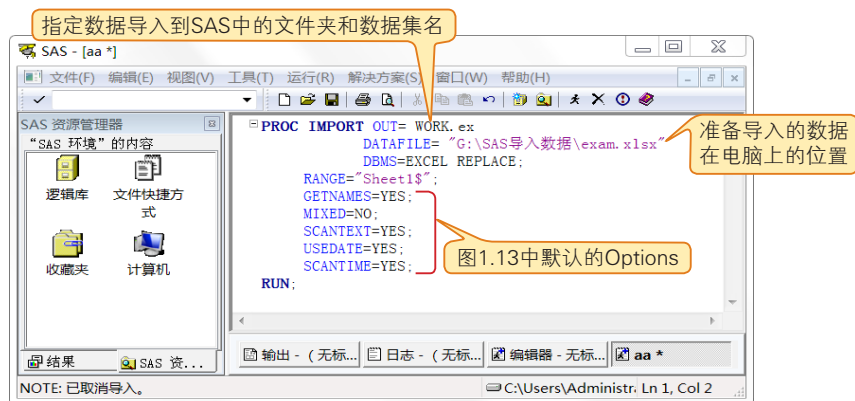


图1.17 SAS自动产生的数据导入的语句

**小白：**听你讲了一遍菜单导入数据，再看这个SAS程序，好像也不难了，只要记住这几个蓝色的语句就可以了。

**Ms.周：**所以说SAS的编程其实与菜单是差不多的，只要记住几个关键的词，写起来也很简单。

## 1.3 本章小结

**Ms.周：**今天是第一天，我们就少讲点，先把今天讲的内容总结一下。

- ◆ 首先要注意SAS输入数据的一些细节，初学者最容易犯一些低级错误，比如漏掉分号、语句单词写错、写了中文标点等。常见的各种错误类型可以参考附录A。对于初学者来说，在程序的句子后面添加注释（格式为：/\* 注释 \*/）是个很好的习惯，可以随时把自己的理解写上，便于以后翻看。

## >> 小白学SAS

- ◆ 掌握数据导入过程，这在以后会经常遇到。导入的SAS语句实在记不住也不要紧，但起码要记住如何通过菜单导入（当然这不是唯一的方式，但却是最简单、常用的方式）。实际中多数人习惯把数据录入到Excel中，因此导入Excel数据很常见。但一定要注意，Excel不是专业数据库，很容易出现同一个变量存在两种类型的情况，此时一定要注意导入时的选项，不同选项会使导入的数据生产不同的类型。
- ◆ SAS逻辑库并不是缥缈虚无的，而是与电脑上的相应文件夹相对应。所以每建的一个数据集，实际上都存到了电脑上，只不过默认放在work文件夹下的数据集不会给你永久保存，只要关闭SAS，它会立刻帮你把文件清空。所以如果想反复用某一数据集，还是把它放到sasuser文件夹中比较保险，当然也可以自己在逻辑库中新建一个文件夹用于存放自己的数据。
- ◆ 常见的建立永久数据集的方式有三种：一是在逻辑库中用菜单新建文件夹，并与电脑上已有的某个文件夹关联起来；二是用libname语句指定文件夹名，并与电脑上已有的某个文件夹关联起来；三是用data语句直接指定电脑上已存在的文件夹路径及SAS数据集名。

# 第2章

## SAS数据集建立的高级议题

**小白：**经过昨天的学习，我已经可以在SAS中输入数据了。

**Ms.周：**现在说会输入数据还言之过早，你只是掌握了SAS输入的基本结构，真正输入数据的时候，情况是很复杂的，而且也有很多技巧。比如，不同类型变量的输入方式是不同的，输出格式也是各有特点。

**小白：**等等，听起来好晕啊。

**Ms.周：**别着急，今天准备给你讲点高级的内容，虽然也是讲建立数据集，但是更实用一些。因为现实中在输入数据的时候，不会像昨天给你讲的那么简单，而是会遇到各种意外情况。今天主要就是教你处理这些情况。

**小白精神一振：**那我得仔细听一下了。

**Ms.周：**在讲之前，我先问你一个问题：你知道SAS中的变量有几种类型吗？

**小白有点心虚地说：**是不是数值型、字符型和日期型？

**Ms.周：**不用这么没底气，你说的没错，不过需要纠正一下，SAS中的变量其实只有两种类型：字符型（character）和数值型（numeric）。

**小白：**没有日期型吗？别的软件里都有日期型。

**Ms.周：**日期在SAS中是作为数值型，所有日期型变量被作为输入日期与1960年1月1日之差。如1980-1-1，SAS会默认这个值是7305，而且也会显示为7305。

**小白：**这么回事啊。那如果我需要用到日期怎么办？

**Ms.周：**这就需要你自己定义变量的输入格式和输出格式了。举个例子，如果你输入了7305这个数，当你把它定义为数值格式，它就是7305，当你把它定义为日期格式，它就是1980-1-1。所以定义输入和输出格式在有些情况下是很必要的。

## 2.1 SAS变量的输入格式

**Ms.周:** 输入格式就是你对变量规定一种格式，让SAS能按这种格式正确读取不同类型的变量。比如你输入20001020，如果规定输入格式是数值，SAS就会把它读为“两千万一千零二十”，如果规定输入格式是日期，SAS就会把它读为“二零零零年十月二十日”。

**小白:** 我明白了，这么说的话，有的变量我们必须规定输入格式。

**Ms.周:** 不错，尤其是日期变量，一定要规定相应的输入格式。输入格式的基本形式是这样的：

```
input 变量1 输入格式1 变量2 输入格式2 .....;
```

**小白:** 看起来好像很简单的样子。

**Ms.周:** 格式很简单，不过不同类型变量的输入格式是不同的，所以需要分别来说一下。

### 2.1.1 数值型变量的输入格式

**Ms.周:** 数值型变量的输入格式主要是`w.d`。其中，`w`表示数值的总位数或宽度（包括小数点），`d`表示数值的小数部分的位数，如6.3表示总位数是6，包括3位小数；再如4.表示总位数是4，没有小数。一定要注意，**即使没有小数，点号也一定要加上。**

**小白:** 所有的数值都需要指定输入格式吗？

**Ms.周:** 这个问题很有意思，其实绝大多数情况下，数值型变量后面是不加任何输入格式的，就像昨天你做的例子，`gender`和`age`后面什么都没加。尤其变量是整数的时候，更不能随便加，否则会出现意想不到的问题。

**小白:** 我不是很明白。

**Ms.周:** 我给你举个例子就清楚了。你看下面这段程序，觉得SAS会怎么读数据呢？

```
data fh;
input x 4.2;           /*变量后的4.2表示变量x的宽度共4位，其中小数有2位*/
cards;
12
2.1
15.6
23.46
;
proc print;             /*这句话的作用是让读入的数据显示在输出窗口*/
run;
```

**小白:** x指定格式4.2, 那就是说一共读取4位数, 其中包括2位小数。这样的话, 这4个数应该是12.0、2.10、15.60、23.46, 对不对?

**Ms.周:** 我运行一下, 让结果来告诉你对不对吧。顺便说一下, `proc print;`的作用是输出数据集, 把读取的数据显示到输出窗口, 后面我们会一直用这个过程来显示数据集。

Obs	x
1	0.12
2	2.10
3	15.60
4	23.40

**小白大吃一惊:** 怎么跟我想象的差这么大, 这是怎么回事呢?

**Ms.周:** 让我们来仔细研究一下吧, 首先, SAS先根据设定的总位数来读取数据, 也就是说, 每个数只读取前4位, 这样4个数分别是12、2.1、15.6和23.4。

**小白:** 我明白了, 23.46算上小数点一共是5位, 第5位的6没有被读取, 所以就变成了23.4了。但这跟最后显示的结果还是不一样啊。

**Ms.周:** 别急, 刚才只是根据总位数这一个条件读取的结果, 还得继续看小数位数的规定。小数位数规定了2位, 所以每个数必须有2位小数点, 所以2.1、15.6和23.4分别变成了2.10、15.60和23.40。

**小白:** 原来如此啊。但是12为什么不是12.00, 反而是0.12呢?

**Ms.周:** SAS读整数就是这样, 没有小数点就自动把自己降为小数。如果你在录入的时候是12.0, 那就会读成12.00, 但12就读成了0.12。

**小白:** 原来整数和小数的读取规则是不一样的。整数就把自己降为小数了, 而小数只是小数部分改变了。那我以后得仔细了。

**Ms.周:** 所以我说, 大多数情况下, 数值是不加输入格式的, 但有时加上反而是画蛇添足。

### 2.1.2 字符型变量的输入格式

**Ms.周:** 字符型变量的输入格式主要是\$ *w.*。其中\$符号是必须加的, w表示字节数(1个中文占2字节), 一般只有在变量宽度超过8字节的时候才有必要加。因为**SAS默认对字符只读取8位**, 超过8位的部分被截取掉了。所以如果你的变量宽度超过8字节, 最好加上宽度值, 以保证SAS能全部读取。比如下面这段程序:



```
data fh;
input pro $12.;
cards;
山东省青岛市
;
proc print;
run;
```

如果不加“12.”，只加一个\$符号，那么SAS最多对pro读取8字节，也就是4个中文字符，结果就会显示“山东省青”。加上“\$12.”才能让SAS完整读取这6个字。

**小白：**明白了。格式中的“.”是必需的吗？

**Ms.周：**对，只要加上宽度值，后面必须跟着“.”，但如果只写\$符号而没有宽度值，就不用加“.”。

## 2.1.3 日期型变量的输入格式

**Ms.周：**日期型变量的输入格式最多，你想想有多少种日期写法就知道了。表2.1列出几种常用的日期型输入格式。其中w表示日期的宽度，这个宽度不能超过32，具体写多少需要根据录入的数据宽度而定。

表2.1 几种常见的SAS日期格式

常见输入格式	宽度w范围	举例	具体输入格式
YYMMDDw.	6~32	20130728	YYMMDD8.
		2013/07/28	YYMMDD10.
MMDDYYw.	6~32	072813	MMDDYY6.
		07282013	MMDDYY8.
DDMMYYw.	6~32	280713	DDMMYY6.
		28/07/2013	DDMMYY10.
DATEw.	7~32	28JUL13	DATE7.
		28JUL2013	DATE9.
MONYY7.	5~32	JUL2013	MONYY7.
		JUL13	MONYY5.

**小白：**也就是说，我输入数据的形式不同，就需要指定和它对应的输入格式，对吗？

**Ms.周：**没错，比如你的数据是072813，这是按“月日年”的格式输入的，而且总宽度是6，

那就应该指定mmdyy6.，而不能写别的格式。

**小白：**年月日之间可以加分隔符吧？

**Ms.周：**可以，比如我们常用的分隔符“-” “/” “.”等，SAS都可以识别。不过分隔符也是占宽度的，如果加上分隔符，就要把它的宽度也算进去。如13-07-28，就应该指定yymmdd8.的格式。

**小白：**如果我想录入2013年5月8日，可不可以写成1358或201358呢？

**Ms.周：**如果年月日之间没有任何分隔符，此时SAS要求输入的月和日都是2位。如果你的月和日都只有一个数，那就在前面用零补上，就像你说的2013年5月8日，需要写成130508，然后用yymmdd6.的格式读取。但2013-5-8这种形式是允许的，因为中间有分隔符。

**小白：**我明白了，不过这些格式好多啊，我需要都记住吗？

**Ms.周：**其实国内一般最习惯用“年-月-日”的形式来输入日期，也就是yymmdd的格式，像date这样的格式一般国外用得比较多。如果你自己录入数据，只要掌握一种习惯的方式就可以了。不过有时你可能需要把已经录好的数据导入到SAS，这时它们的格式可能各种样子都有，比如05Apr2012，那就必须知道这样的输入应该指定date9.格式。

**Ms.周：**最后我再跟你强调一点，这些格式都是固定的，不能想当然地去修改。比如录入2013-05-12，有的人觉得2013是4位数，应该用yyyymmdd10.格式，这是错误的。SAS中只有yymmdd10.格式，绝不能根据自己的想象去改变格式。

2.1.4 两个特殊输入符——:和&

1. 冒号（:）的作用

**Ms.周：**刚才说的几种输入格式都记住了吗？

**小白：**记住了，我还做了笔记呢。

**Ms.周：**那好，我现在考你一下，用下面这两条记录建立一个SAS数据集：

city	zone
山东省蓬莱市	0536
山东省青岛市市南区	0532

**小白：**没问题。等等，好像有个问题，city这个变量第1个记录宽度是12，第2个记录宽度是18，我应该写哪个呢？

**Ms.周:** 现在发现也没有这么简单了吧。你觉得应该写多少?

**小白:** 如果写12, 第2个记录就读取不全, 所以我认为应该写18。变量zone不超过8字节, 可以不加宽度, 所以我只写\$符号就可以了。

```
data fh;
input city $18. zone$;
cards;
山东省蓬莱市 0536
山东省青岛市市南区 0532
;
proc print;
run;
```

Obs	city	zone
1	山东省蓬莱市 0536	山东省青

**看到结果, 小白傻眼了:** 怎么会这样呢?

**看到小白一副苦恼的样子, Ms.周说:** 还记得以前我跟你说过, SAS默认是以空格作为变量分隔的标志吗?

**小白:** 我记得啊, 我的变量和数据之间都空格了, 为什么不管用呢?

**Ms.周:** 我说的是“默认”, 但是一旦对变量指定了宽度, SAS就不再以默认的空格作为变量区分的标志了, 而是按指定的宽度来识别变量。就像这个例子, 你指定了city的宽度是18, SAS就会从头开始读取, 一直读到18位结束才认为是city的值, 不管中间读到什么, 哪怕是空格, 只要不到18位, SAS就认为这是city变量的值。

**小白:** 怪不得, SAS是把“蓬莱市”后面的空格和0536都读成city的值了。那为什么zone的值是“山东省青”呢?

**Ms.周:** 既然SAS把“山东省蓬莱市 0536”读作是city的值, 那么zone的值依次往后读, 应该是“山东省青岛市市南区”, 但zone没有指定宽度, 所以默认只能读取8位, 也就是“山东省青岛市市南区”的前4个中文字。

**小白:** 这下明白了, 怪不得第2个记录也显示不出来, 整个都乱了。那这种情况应该怎么办呢?

**Ms.周:** 解决的办法很简单, 在变量和输入格式之间加一个冒号(:)。冒号的作用是告诉SAS, 如果要读取下一个变量, 需要满足下面任一条件: 要么遇到空格, 要么变量的宽度读完了。我现在把你刚才的程序改一下:

```
data fh;
input city: $18.  zone$;          /*在city后加了一个冒号*/
cards;
山东省蓬莱市 0536
山东省青岛市市南区 0532
;
proc print;
run;
```

Obs	city	zone
1	山东省蓬莱市	0536
2	山东省青岛市市南区	0532

**小白:** 咦, 现在结果就对了。想不到加上一个小小的冒号竟然有这么大的作用。

**Ms.周:** 加上冒号之后, SAS读取第1条记录时, 当读到12位的时候, 就遇到了满足的条件之一, 即遇到了空格, 所以就把空格后的0536作为第2个变量值。读第2条记录时, 当读到18位的时候, 遇到了满足条件之一, 即指定的宽度读完了, 所以就把后面的0532作为第2个变量值。

**Ms.周:** 冒号在实际的数据输入中非常实用, 当同一变量的不同数据有不同宽度时, 可以指定所有数据中最大或更大的一个宽度值, 然后结合冒号读取。比如这个例子, 2条记录中, 位数最多的是18位, 你可以指定一个大于或等于18的宽度。因为冒号是双重判断标准, 就算指定的宽度与实际值不完全一样也不要紧, 还有空格这个条件, 总有一个会满足的。

**小白:** 为什么要指定一个数据中最大或更大的宽度值, 而不能小于这个值呢?

**Ms.周:** 因为如果指定的宽度小于记录中最大的位数, SAS会先满足“宽度读完”这一条件, 这样SAS就会认为第1个变量已经读完, 后面就是第2个变量, 此时就会出问题。

**小白侧着头想了一下:** 那就是说, 如果这个例子中我指定宽度为12, 那么第2个记录的city变量当读到“山东省青岛市”时就先满足了“宽度读完”条件, 这样就会认为city的值就是“山东省青岛市”, 对吗?

**Ms.周:** 完全正确。所以, 如果你用加冒号的方式读取数据, 一定要指定一个大的宽度值, 如果你不确定记录中最大的宽度是多少, 就直接指定个很大的值, 比如1000, 因为很少有字符型变量会写500个汉字或1000个英文。

**小白:** 那数值型变量是不是也可以用冒号呢?

**Ms.周:** 冒号的用法是通用的, 不管是数值型还是字符型都可以。比如下面这段程序, 你觉得输入格式写得对不对呢?

```
data fh;
input day1 yymmdd6. day2 yymmdd8.;
cards;
130125 20130528
;
proc print;
run;
```

**小白看了一下：** day1的值是6位，day2的值是8位，指定的宽度都对，而且yymmdd也和数据的形式一致，应该没什么问题吧？

**Ms.周：**乍一看好像没错，但你忘了我跟你强调的一句话，**一旦对变量指定了宽度，SAS就按指定的宽度来读取变量。**变量day1指定宽度为6，这没有问题，但是由于第2个变量也指定了宽度，这时SAS严格按宽度来读取。也就是说，第2个变量day2是从第7位往后读8位，也就是“空格+2013052”，这样就出现了读取错误。

**小白恍然大悟：**对，你刚才说过，如果指定了宽度，SAS就忽略空格的作用，而是把包括空格在内的所有字符都作为变量的值。那这里也可以用冒号来解决吧？

**Ms.周：**对，只要在day1和day2后加上冒号，然后把宽度都指定得大一点就可以了，比如像下面这样指定宽度为10。**不过要记住，日期型变量的宽度最大不能超过32。**

```
input day1: yymmdd10. day2: yymmdd10.;
```

**小白：**那是不是说，如果我有超过2个以上的变量指定了宽度，最好都用冒号结合一个大的宽度值的方式来输入呢？

**Ms.周：**这样写是一种非常保险的输入方式，你就永远不会读错数据了。

## 2. &符号的作用

**Ms.周：**还有一种特种输入方式需要跟你讲一下，尽管多数情况下可能用不到，但万一遇到，你应该知道怎么处理。这种情况就是，如果你的变量值本身包含空格怎么办？

**小白：**是啊。你说过，SAS默认以空格作为变量的分隔符，万一变量中有空格，SAS岂不是把它读成2个变量了？

**Ms.周：**确实如此，所以需要有一种方法来处理这种特殊情况。这种方法其实也很简单，就是在变量名后加上&符号。比如录入下面这2条记录。

name	city
Peter Parker	山东省 蓬莱市
Ross Geller	山东省 青岛市 市南区

**小白:** 这2个变量的值都包含空格，而且长度还不一样，好复杂啊。

**Ms.周:** 你可以先想一下，如果这两个变量值都没有空格，你会怎么输入呢？

**小白想了一下:** 如果没有空格，只是宽度不同，我会这么写input语句：

```
input name:$50. city:$50.;
```

**这就是按照你教我的原则:** 首先，这2个变量的记录宽度都大于8，所以必须指定宽度值。其次，每个记录的宽度又不同，没法指定一个统一的宽度，所以指定宽度的同时在变量后加个冒号。最后，加冒号最好指定一个比较大的宽度值，我目测了一下，2个变量的每个记录的宽度应该都不会超过50，所以我就写了50作为宽度。

**Ms.周:** 很好，看来你已经掌握了这种输入方式。变量值包含空格的情况其实就是简单地在变量后多加上一个&符号，就像下面这段程序：

```
data fh;
input name&:$50. city&:$50.;
cards;
Peter Parker  山东省 蓬莱市
Ross Geller   山东省 青岛市 市南区
;
proc print;
run;
```

**小白:** 咦，看起来就是比我刚才写的input语句多了2个&符号。这个&符号必须放在冒号前面吗？

**Ms.周:** 这个不一定，&和冒号可以互换，但它们最好都放在\$前面。而且你有没有注意到，这两个变量值之间隔得远，因为我中间加了2个空格。

**小白:** 你是说Parker和山东省之间、Geller和山东省之间是空了2个空格？

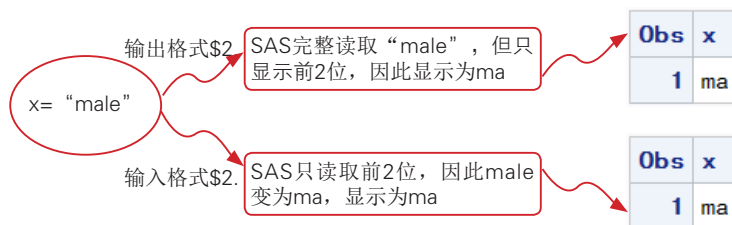
**Ms.周:** 是的，因为每个变量值本身就有空格，所以变量之间就需要用2个或2个以上的空格隔开，这样SAS就会把这2个或2个以上的空格作为变量分隔符。

## 2.2 SAS变量的输出格式

**Ms.周:** 刚才说的输入格式是给SAS指定一个格式，让它按这种格式来读数据。现在我们要说的输出格式，也是指定一个格式，但它不是让SAS按这种格式读取，而是按这种格式显示给你自己或其他人看。

**小白:** 好像不是很明白，它们有什么实际的差别吗？

**Ms.周:** 有一个差别很重要, 当你指定输入格式时, SAS按这种格式读取, 会真正改变你的输入值。而输出格式不会真正改变你的输入值, 只是SAS按你指定的样子显示而已。比如字符变量x的值为male, 如果指定x的输入格式为2., 那么SAS就会把x读成ma, 此时x的值就不再是male, 而是ma。如果指定x的输出格式是2., 那x仍然是male, 但它会显示为ma。



**小白:** 我明白了, 指定输入格式会改变变量的值, 而指定输出格式不会改变变量的值, 只是改变它们显示的样子。

**Ms.周:** 是的, 这一点很重要。输出格式通过在data步中增加一个format语句来定义, 多数情况下, format语句写在input和cards;之间, 其基本形式为:

```
format 变量1 格式1 变量2 格式2 .....;
```

## 2.2.1 数值型变量的输出格式

**Ms.周:** 数值型变量常用的输出格式有很多, 我这里主要介绍常用的三种: `w.d`、`commaw.d`和`percentw.d`。`w.d`的含义与输入格式中的`w.d`一样。`commaw.d`的作用是将数值的整数部分自右向左每三位用逗号隔开, 当数值位数较多时, 这是比较标准的表示方式。`percentw.d`的作用是将数据显示为百分比的形式, 它自动将变量值乘以100, 并加上%。

**小白:** 这里的`w`和`d`也是分别表示总宽度和小数宽度吗?

**Ms.周:** 是的。而且`commaw.d`中的`w`是包括逗号和小数点在内的, `percentw.d`中指定`w`时要预留出3个字节的位置显示%。这几种格式都很好理解, 我给你举个例子说明一下:

```
data wt;
input num cost prop;
format num 5.2 cost comma12.1 prop percent8.2;
cards;
50 10205600 0.1236
45 9580000 0.0361
;
```

```
proc print;
run;
```

Obs	num	cost	prop
1	50.00	10,205,600.0	12.36%
2	45.00	9,580,000.0	3.61%

**小白:** 这个显示结果我能看懂，num一共显示了5位，其中2位是小数点，cost一共显示了12位，其中1位是小数点，而且按每千位加了逗号，这就是comma的功劳吧？

**Ms.周:** 是的，而且你可以数一下10,205,600.0正好是12位，如果这里你指定的总宽度小于12，那它就没法兼顾，只能不显示“,”。所以在用comma格式时，总宽度一定要把可能出现的“,”的个数也考虑进去，否则你会发现comma好像没起作用一样。

**小白:** 这里percent格式必须指定8位吗？

**Ms.周:** 你可以看一下12.36%这个值，12.36占5字节，%要占3个字节，所以最少要指定8，否则显示不全。

## 2.2.2 字符型变量的输出格式

**Ms.周:** 字符型的输出格式与输入格式一样也是\$w.，w表示字节数（1个中文字符占2个字节）。

**小白:** 我记得你曾经说，SAS对字符型变量只默认读取8字节，那是不是也默认只输出8字节呢？

**Ms.周夸道:** 看来你学得挺不错。对于字符型变量来说，只要把变量正确读取进来，SAS就会原模原样地把它输出出去。换句话说，其实SAS中无须设置字符型变量的输出格式，让它按读入的格式原封不动地显示就好了。

**小白:** 那就是说，输出是没有8字节的限制了，读入多少就输出多少。那对字符型变量而言，其实我只要设定合适的输入格式就可以了，完全可以不设置输出格式？

**Ms.周:** 实际中确实如此，很少有人设置字符型变量的输出格式，除非为了某些特定的显示目的。不过这里我可以通过一个例子来帮助你理解输入和输出格式的区别：

```
data fs;
input x$ y$2.;
format x$2.;
x1=x+1;
```