第九章　Splus

§9.1　简介

Splus由MathSoft统计科学部开发，供统计、应用数学和科学研究者使用的通用工具包。其前身是AT&T的Bell实验室Becker, Chambers和Willks研制的S 语言。1988年S系统被彻底改写从而称为"New S"，其3.0、3.1、3.2版分别于1991、1992 和1994年引入。其设计目的是向用户提供：动态、交互和高品质的图形，探索性数据分析方法，统计方法，进行数学计算。

Splus引导用户进行探索性、数据驱动和面向图形的分析。它允许用户编程和与其它语言的接口对进行功能扩展。Splus也有专用工具箱，如S+BOX用于工业设计。

§9.2　操作使用

运行环境：为运行图形用户接口(GUI)的工作站(如openLook 、motif) 和Microsoft Windows，或X-终端。在Splus中可以用命令?Devices列出软件所支持的设备名。典型的环境下，用户可以录入Splus表达式、阅读帮助文件、显示图形并与操作系统交互。Splus也可以在非图形终端运行。

§9.2.1　开始与结束

以Microsoft Windows系统为例，运行时需要home和shome两个环境变量。可以在系统配置文件config.sys或自动批处理文件autoexec.bat中使用命令：

```
set home=c:\splus
set shome=c:\splus
```

随后在Windows内就可以启用SPLUS.EXE了。

又以UNIX系统为例，在系统提示后键入：Splus

稍微停顿后，出现提示：>

使用命令q()退出S-Plus。

在系统下使用命令：setenv S_CLEDITOR emacs，则使用：Splus - e 进入Splus时将启用与emacs相容的命令行编辑功能：

^P(Previous，上次命令)、^N(Next，下一次命令)、^ K(Kill，删行)、^A(Begin，开始)、^E(End，行末)、^D(Delete，删字符)、^X(Insert，插入)等。命令history()用于显示曾经键入的命令，而again("dta",ed=T)则使包含"dta"的命令再次显示从而进行编辑。

S-Plus也可以运行命令文件：Splus BATCH <命令文件名><输出文件名>

在交互方式下也可用source()命令调用外部命令文件，类似SAS的include。

Splus具有解释语言的特征，系统执行读入后的每个命令；Splus又有功能性语言的特征，录入的表达式是系统功能调入，数据是调用的参量，同时系统又返回数据；最重要的是Splus是一个面向目标的程序语言，一个通用的功能可以由多种类型的目标调用。目标是Splus的数据、结果及函数，它们以系统文件的形式贮存，用命令objects()得到它们的列表，用rm()命令进行删除。

如上所述，用户用表达式与Splus进行交互，表达式多种多样，但在交互式环境下最主要的是命名和功能调用。打入Splus函数名就会得到相应的定义。一个功能调用通常是函数名及其参数(一般放在括号内)。所有的Splus表达式返回一个值，通常这个值会打印出来。

Splus对命令字符大小写是敏感的，因此Age与age是不同的。当表达式做为命令键入时，则其值被计算、打印并被舍弃(存于隐含变量.Last.value)。赋值语句(用<-引导)能够计算表达式的值却不自动打印，如：

```
> 12+3
[1] 15
>x<-c(1,2,3,4,5,6,7,8,9)
>m<-mean(x);v<-var(x)
>m/sqrt(v)
```

多个命令用分号分隔，若命令在一行未有打完，则系统以"+"提示。

Splus中的线性统计模型采用通用的Wilkinson-Rogers记号，操作符有+,-,*,/，%和:，:表示交互项。. 是Splus有效命名的一部分，如car.weight；同时它又可以表示公式默认的左端或右端，如update(model,.~.-Age) 。lm(skip ~.^2, data= solder. balance) 表示使用solder. balance中所有变量的主效应和二阶交互。

sink("文件名")可以把运行过程存贮于文件，sink()将关闭文件。

!用于引导系统外壳命令(与Stata相同)，如:!csh进入Unix系统外壳。

§9.2.2 取得帮助

>help(mean) 或?mean

列出具体命令的语法，在UNIX系统下若事先使用setenv EDITOR < 编辑软件名称> 则可在使用v命令后进入相应的编辑，存为文本文件。

在UNIX Windows状态下，用help.start()和help.off()进入和退出帮助系统。前者可以带有自己的参量，如：help.start(gui="openlook")。在Windows下结合Winhelp的编辑剪贴功能可以将帮助内容以文件形式保存下来。

§9.2.3 数据

数据集存放在.Data目录(Unix系统)或__DATA目录(Microsoft Windows)，它们是永久性的，使用ls()命令可以看到。每个数据集可以经attach(目标数据集) 与detach()引用，数据目标可以经rm()命令删除。数据类型：有向量(vector)、数组(array)、矩阵(matrix)、列表(list)、数据框(data.frame)。

> x<-c(1:9,10) 生成一个向量，其中1:9表示1 到9 之间的数字，有如Pascal语言中的枚举类型。seq(-5,5,by=0.5)则生成[-5,5]之间以0.5等分的数列，类似地，req(x,times=5)则将x重复5次。

向量运算规则与其它软件相仿，如z <- 5 * x + y表示z的每个元素是x的每个元素乘5并与y相应的元素相加的结果。z <- y > x 是将y与x 间的逻辑运算结果存放在z中。x[1:5]是指x向量的第1至第5个值，其下标表示方式与Fortran 中的字符操作类似；但是x[-(1:5)则表示把1:5的数据排除在外]。

数组可以想象成有下标的同样类型的数据的集合，设z是有900个元素的向量，>dim(z)<-c(3,2,150)则使z作为3 x 2 x 150阶的数组。又如：x<-array(1:20, dim=c(4,5))。

列表是由各个部分组成的目标的有序集合，其各个组分用$加以区分并可以拥有自己的命令，如：>name$component，各组分的命令可以用>names(name) 给出，这样做不需要打印出具体的数据。[[]]用于捡出列表的一个元素而[]用于向量的下标，两者的作用是不相同的。

数据框架是可以包含字符的矩阵，也可看成紧密排列的列表，行列均可以拥有自己的标号，其列可以做为列的的组分来处理。数据框架可以用命令data.frame生成。Splus中缺失值用NA表示，常用操作是na.action=na.omit，如用于lm()指令。

函数sort()用于对数据进行排序，其最简单的形式是用一个参量，如：sort(age)。更灵活的方法是使用sort.list产生一个索引。因此x[sort. list(x)] 与sort(x)的结果相同而x[sort.list(-x)]是降序结果。进一步有order()，它可以取任意数目的参量。

§9.2.4 读入转贮外部数据

可以通过赋值语句、scan()和read.table()来进行。

> counts <- scan()

将等待用户从键盘读入数据，直至文件结束符如UNIX的^D结束。diet<-scan(,"")则是读入字符类型数据。scan()与read.table() 中可以指示文件名，如

auto<-read.table("auto.dat")。

read.table()主要用于读取数据框架，

外部数据的转贮可以用write.table()完成，write()函数写出向量或矩阵。

利用专用程序，可以读取SAS数据集。假设在UNIX上用户name有一个SAS文件是test.ssd，则使用以下命令读至data。

data<-sas.get("/usr2/user2/name/",mem="test")

names<-sas.contents(lib=unix("echo /home/sparc6a/zhao"),mem="test")

§9.2.5 图形

Microsoft Windows下使用win.graph激活图形显示，在UNIX的motif 下使用命令motif()激活图形显示设备，graphics.off()，dev.off()关闭设备。其它的设备如：x11()、openlook()、sunview()等，通讯软件Kermit或NCSA的telnet 均可以使用tek4014()进行仿真。S-Plus 并不需要象SAS那样繁复的options语句，绘图存贮时只需导以postscript()，则自动生成PostScript格式文件。以(-3.14, 3. 14) 内cos(x)的做图为例，其命令只需要两条：

> angle <- seq(-pi, pi, len=100)

> plot (angle, cos(angle), type="1")

绘图函数大致分类如下：

单变量数据：	barplot	条图
	hist	直方图
	dotchart	点图
	pie	圆图
	stem	枝叶图

双变量数据	plot	散点图
	boxplot	盒式图
	qqnorm	单样本正态概率图
	qqplot	两样本分位点图
	plot.surv.fit	生存曲线图
	shewhart	Shewhart质量控制图
	cusum	cusum质量控制图
三维图形	contour	轮廓图
	persp	透视或mesh图
	image	影象图
多变量数据	coplot	条件图
	faces	脸谱图
	maplot	maplot
	pairs	两两散点图
	stars	星形图
	symbols	绘图符号
时序数据	tsplot	一元或多元时序图
	acf	自相关函数图
	spectrum	图谱
动态图象	brush	链接散点矩阵
	spin	可旋转三维图

绘图命令有大量的选择项，如：lty=n指示线型，pch=”c”指示画点用的符号。

points	向当前图形增加点
lines	向当前图形增加线
text	向当前图形增加文字
abline	画点斜式直线

交互式命令有identify、locator、legend。

§9.2.6 概率和统计

主要内容有：概率分布、综合统计量、统计检验、统计模型。与概率分布有关的函数有d分布名、p分布名、q分布名、r分布名。其分布有：

```
beta            binomial        Cauchy          chi-squared
exponential     F               gamma           gemetric
hypergeometric  log-normal      logistic        negative binomial
normal          Poisson         stable          Student's t
uniform         Weibull         Wilcoxon rank sum
```

ppoints()函数用于产生0-1间上的等分点。summary()给出描述统计量。下列语句画一个贝塔分布的直方图和分布图。

```
> sample.data <-rbeta(100,2,9)
> hist(sample.data, den=-1, prob=T)
> p<-ppoints(100)
```

```
> lines(qbeta(p,2,9),dbeta(qbeta(p,2,9),2,9)
> title(main="Data from Beta(2,9) distribution")
```

综合统计量用summary函数得到，根据参数的类别不同，它可以给出相应的结果。综合统计函数有：

mean	算术均值
median	中位数
var	向量的方差、矩阵的协方差阵
cor	向量或矩阵的相关
quantile	经验分位点
location.m	M-估计
mad	平均绝对偏差
scale.m	Bisquare A 方差估计
scale.tau	Huber的方差估计
robloc	M估计和Huber方差估计
cov.mve	多变量数据的稳健位置和方差估计

常用的统计检验有：

t.test	t-检验：单样本、两样本、配对、方差等或不等。
wilcoxon.test	秩和与符号秩次检验
var.test	两方差齐性检验
kruskal.test	单向设计的Kruskal-Wallis检验
friedman.test	无重复区组设计的Friedman秩和检验
cor.test	零相关检验，包括Pearson、Kendall和Spearman相关
binom.test	单个率的精确检验
prop.test	率相等的检验
chisq.test	两维列联表Pearson卡方检验
fisher.test	两维列联表Fisher精确检验
mcnemar.test	两维列联表的McNemar卡方检验
mantelhaen.test	三维列联表的Mantel-Haenszel检验

下面指令演示了两样本t-检验：

```
> x<-rnorm(10)
> y<-rnorm(5, mean=1)
> t.test(x,y)
```

实验设计：fac.design(析因设计)、oa.design(正交设计)、alias()给出实验设计混杂的结构(完全或部分)。

统计模型：Splus中的许多模型是一个统一的框架，数据是一个数据框架，待拟合的模型用一个公式表示出来。公式用符号～引导，如：yield～ Temp+ Conc，log(Mileage)～Weight+ploy(HP,2)。

crosstabs	从一系列因素生成多维列联表
aov, manova	拟合一元和多元方差分析模型
lm	线性模型
glm	拟合广义线性模型
gam	拟合广义加性模型
loess	拟合局部回归模型
tree	拟合分类或回归树模型
nls,ms	拟合参数非线性模型
factanal	因子分析
princomp	主成分分析

回归分析还有l1fit进行L1回归、rreg进行稳健回归，ltsreg 进行最小截尾均方回归等。

生存分析	
surv.fit	拟合Kaplan-Meier生存模型
coxreg	拟合Cox比例风险模型
agreg	拟合Anderson-Gill推广Cox模型
时间序列分析	
ar	一元或多元自回归模型
arima.mle	ARIMA模型
spectrum	时间序列谱估计
质量控制图	
shewchart	Shewchart(xbar、s、R、p、np、u和c型控制图)
cusum	cusum图(xbar、s、R、p、np、u和c)

§9.2.7 数学计算

Splus可以进行积分和导数、插值、逼近和最优化，如：

```
>integrate(sin,0,pi) [1:2]
>D(expression(3+x^2),"x")
>approx(spline(1:10,(1:10)^2),xout=1.5:3.5)
>polyroot(c(6,-5,1))
```

最优化的函数有：	ployroot	复杂多项式的根
	imorppt	给定区间内一元函数的根
	peaks	一系列离散点的局部最大值
	soptimize	给定区间内一元函数的极值
	ms	多元函数的局部极值
	minib	多元函数的极值，变量有上下界约束
	nls	一个或多个多元函数平方和的极小值
	nlregb	一个或多个多元函数平方和的有界约束极小值
	nnls	系数非负的最小二乘解

§9.2.8 用例：图形、经典分析、生存分析、局部回归

Splus系统提供了许多数据，用户用以直接引用，如乙醇数据ethanol。

```
>summary(ethanol)
>pairs(ethanol)
>attach(ethanol)
>loess(NOx~C*E,span=1/2,degree=2,parametric="c",drop.square="c")
>E.Intervals <-co.intervals(E, number=9,overlap=1/4)
>coplot(NOx~C|E, given=E.intervals, panel=function(x,y) panel.
+smooth(x,y,degree=1, span=1))
>m1<-lm(NOx~C+poly(E,2),data=ethanol)
>summary(m1)
>par(mfrow=c(2,2))
>plot(m1)
>plot.gam(m1,resid=T,rug=F)
>m2<-gam(NOx~C+lo(E,degree=2),data=ethanol)
>plot(m2,resid=T,rug=T)
>anova(m1,m2,test="F")
>m3<-gam(NOx~lo(C,E,span=1/4,degree=2),data=ethanol)
>anova(m2,m3,test="F")
```

fitted(), residuals(), summary(), predict(), family(), deviance(), formula()可用于获得gam目标的相应结果。

脊柱侧弯数据Kyphosis:

```
> attach(Kyphosis)
> kyph.gam1 <-gam(Kyphosis~s(Age)+s(Number)+s(Start), family=binomial)
> class(kyph.gam1)
> plot(kyph.gam1, residuals=T, rug=F)
> summary(kyph.gam1)
> kyph.gam2 <-gam(kyph.gam1, ~ . -s(Number))
> summary(kyph.gam2)
> plot(kyph.gam2, se=T)
> anova(kyph.gam1,kyph.gam2, test="Chi")
```

s()表示非参的平滑项，bs()与ns()则是B-样条和自然样条，lo()表示loess() 平滑，它们可以用df选项指示自由度。

最后的语句对两个模型进行比较，结果表明省略的项并不显著。step.gam()和predict.gam()可用于逐步模型选择和预测。

书写用户自定义函数一般的格式是：name <- function (arguments) body 。函数的参数在括号内给出，用逗号分开，使用等号可以给参数设默认值。函数体含在大括号内，函数返回的值是函数最后的计算值。函数体内的定义对于函数本身来说是局部的。以下是一个函数绘图的例子：

```
> f.plot
function(f,minx,maxx,nx=100, type="1",...)
{
```

```
        x <- seq(minx, maxx, length.out=nx)
        y <- f(x)
        plot(x, y, type = type, ...)
}
```

参数是minx与maxx指示x的取值范围，nx指示等分点数，默认各点用线是连接起来。可以用指令：

```
>f.plot(cos, -pi, pi)
>f.plot(function(x) {1+2*x+x^2},-10,10)
```

设要在UNIX系统下使用pico编辑，可以进行如下操作：

```
> !pico myeditor
function (data,file,editor="pico")
{
   if (missing(data))
      ed(editor=editor)
   else if (missing(file))
      ed(data,editor=editor)
   else ed(data,file,editor=editor)
}
>pico <-source("myeditor")
>pico (d)
>pico (d,"e")
```

首先启用系统的pico，然后进行函数定义并存于文件myeditor ，最后作为Splus目标存起来，以后就可以在Splus内不用系统外壳直接使用pico了。

if/else语句用于控制转向，for 用于迭代。

Splus主要有五种方法调用C或Fortran函数，它们是：

1. 静态调用. 产生Splus函数的用户拷贝，包括所有子程序。它启动编译程序并且产生执行文件local.Sqpe(Microsoft Windows 下为nsplus.exe)如调用gee 进行广义估计方程程序：

   $ Splus LOAD gee.c

2. 动态调用(dyn.load). 每个文件用通常的方式编译，如果多于一个文件，它们应累积为单一的可重新定位的目标文件：

   $ ld -r -d objects.o chi.o lgamma.o other.o

   在SunOS用-d而在Sun Solaris用-dn。之后在Splus内启用dyn.load("objects .o")动态调用。Unix Splus拥用COMPILE工具，Microsoft Windows用Watcom编译。

3. 共用库(dyn.load.shared). 在SGI和DEC Alpha这是唯一的方法，如：

   dyn.load.shared("./shlib.so")

   其参数必须是绝对路径。共享库用Splus SHLIB -objects.o chi.c lgamma.c other.c 产生。

4. 增强动态调用(dyn.load2). 基本上与dyn.load类似。

5. 动态链接库(dll.load). 在Microsoft Windows 3.2引入。

函数is.loaded可用以测试某个函数是否已被调入。

library() 引用用户自定义库。比较著名的如survival用于生存分析和oswald 用于长期数据分析。若在用户级安装这些库，要用lib.loc参量，如：

```
> assign(where=0, "lib.loc","/home/sphajiz/oswald")
> library()
```