

第3章

SAS 函数与 CALL 子程序

本章内容包括：

- SAS 函数定义；
- SAS 函数自变量与结果；
- SAS 函数分类；
- 日期时间函数；
- 常用概率分布函数；
- 分位数函数；
- 样本统计函数；
- 随机数函数；
- SAS Call 子程序。

3.1 SAS 函数定义

3.1.1 函数定义

SAS 函数是一个子程序，对自变量返回一个结果值。

SAS 函数的形式：

函数名 (X1, X2, ...)

3.1.2 函数用法

1. 用于组合表达式和条件语句

函数用于组合表达式可以简化 DATA 步编程和一些统计计算。

例 3.1 表达式中用函数。

```
data a(keep=date y d min);          /* 数据集 A 只保留 date, y, d, min 4 个变量 */
set ResDat.idx000001;
```

```
y=year(date); /* 函数 year 给出变量 date 年份 */
d=weekday(date); /* 函数 weekday 给出变量 date 在一周内的哪一天 */
min=min(sum(oppr, hipr, lopr, clpr),1000);
run;
```

任何可用 SAS 表达式的地方都可以用 SAS 函数,这样可以简化条件语句。

例 3.2 条件语句中使用函数。

```
data a;
set ResDat.idx000001;
where year(date)>1996; /* Where 语句中使用 Year 函数 */
run;
```

2. 用于处理多变量

例 3.3 用 SAS 函数处理多变量可以简化程序。

不用 SAS 函数:

```
totx=x1+x2+x3+x4+x5+x6+x7+x8+x9+x10;
if totx<y then least=totx;
else least=y;
```

使用 SAS 函数:

```
least=min(sum(of x1-x10),y);
```

例中,两段程序的结果相同,第二段只有一个语句,非常简单。

3.2 SAS 函数自变量与结果

3.2.1 函数自变量

1. 自变量类型

- 变量名;
- 常数;
- 函数;
- 表达式。

例 3.4 函数及其自变量类型举例。

函 数 举 例	自变量类型
Max(X,Y);	变量名 X,Y
X=Repeat('—', 20);	常数
Least=Min(sum(of x1-x10),y);	变量名和函数
Num=sum((if x1>0), (if x2>0))	表达式

2. 自变量个数

有些函数不需要自变量,有些只需要一个自变量,有些函数有多个自变量。SAS9.1 中,自变量个数不能超过 32767。如果函数的自变量是表达式,在函数被调用之前,首先计算所有表达式自变量。

3. 自变量表示法

当函数有多个自变量时,必须用逗号分隔开。

4. 自变量名缩写方法

- 函数名(OF 变量名 1—变量名 n);
- 函数名(OF 变量名 1…变量名 n)。

例 3.5 正确与错误的表示方法。

正确的表示方法	错误的表示方法
sum (x, y, z)	sum (x y z)
sum (of x1—x10)	sum (x1—x10)
sum (of x1—x10 d y1—y100)	sum (x1—x10 d y1—y100)

例 3.6 数组元素自变量的缩写方法。

```
array y{10} y1-y10;      /* 定义数组 y{10} */
x=sum (of y{* });        /* y{* }中的 * 是显式下标数组的特殊表示符号 */
z=sum (of y1-y10);
```

例中,两个 SUM 函数的表示法是等价的。

3.2.2 函数结果

通常情况下,函数的结果由其自变量的属性决定:

- 自变量是字符时结果变量为字符;
- 自变量是数值时结果变量为数值。

PUT 函数是一个例外,它不管自变量是什么类型,其结果总是字符值。根据默认规则,对大多数函数来说,数值目标变量的长度是 8。目标变量长度不用默认规则的函数由表 3.1 列出。

表 3.1 目标变量长度不用默认规则的函数

函 数	目标变量类型	目标变量长度
INPUT	数值或字符	函数自身规定的输入格式的宽度
PUT	字符	函数自身规定的输出格式的宽度

函 数	目标变量类型	目标变量长度
SUBSTR	字符	自变量的长度
TRIM	字符	自变量的长度

3.2.3 显示函数值的简单方法

SAS 系统显示函数值的简单方法是用 PUT 语句在 LOG 窗口显示。

例 3.7 显示概率值和分位数。

```
data;  
Y=probnorm(1.96);      /* 标准正态分布小于 1.96 的概率 */  
put Y;  
q1=tinv(.95, 2);       /* 自由度为 2 的 t 分布的 0.95 分位数 */  
q2=tinv(.95, 2, 3);    /* 自由度为 2, 非中心参数为 3 的 t 分布的 0.95 分位数 */  
put  q1=q2=;  
run;
```

3.3 SAS 函数分类

在 SAS 系统下的帮助菜单下选 SAS FUNCTIONS: Function Categories 即可得到如表 3.2 所示的 SAS 函数分类。

表 3.2 SAS 函数分类

函 数 分 类	中 文 说 明
Application Response Measurement (ARM)	反应测量
Arithmetic Functions	算术函数
Bitwise Logical Functions	逐位逻辑函数
Character Functions	字符函数
Character String Matching Functions	字符串匹配函数
Currency Conversion	货币转换函数
Date and Time Functions	日期和时间函数
DBCS Functions	双字节函数
Dynamic Link Library	动态链接库
External File Functions	外部文件函数
Financial Functions	金融函数
Library and Catalog Functions	库和目录册函数
Mathematical Functions	数学函数
Probability and Density Functions	概率和密度函数
Quantile Functions	分位数函数

函 数 分 类	中 文 说 明
Random Number Functions	随机数函数
Sample Statistic Functions	样本统计量函数
SAS File I/O Functions	SAS 文件输入输出函数
Special Functions	特殊函数
State and Zip Code Functions	州和 ZIP 码换算函数
Trigonometric Functions	三角函数
Truncation Functions	截取函数
Variable Information Functions	变量信息函数
Noncentrality Functions	非中心分布函数
Web Tools	网络工具

SAS 函数与使用说明见附录 B。

SAS 9.1 系统提供标准函数比一般高级语言多得多。如此丰富的 SAS 函数,为用户编写 SAS 程序带来极大的方便。

下面介绍几类常用的 SAS 函数。

3.4 日期时间函数

3.4.1 日期时间函数

SAS 日期和时间存储标准是以 1960 年 1 月 1 日 0 时 0 分 0 秒为起点,然后以相应的间隔计时,如 1960 年 1 月 1 日 9 时 0 分,按日计的数值就是 0,按小时计的数值就是 9。1960 年 1 月 2 日 0 时 0 分,按日计的数值就是 1,按小时计就是 24 等。

由于日期函数是一类非常重要的函数,特别是对金融数据处理和金融计算,借助日期函数可以极大提高效率。如果不将日期变量作为日期型设置,就会失去很多宝贵的信息,数据处理也不方便。因此,本部分特意将 SAS 日期和时间函数及其应用作较为详细的讨论。日期时间函数如表 3.3 所示。

表 3.3 日期时间函数

函 数	功能与举例
DATDIF(sdate,edate,basis)	返回两个日期之间的天数: DATDIF('01jan2002'd,'01jan2003'd,'act/act'); DATDIF('01jan2002'd,'01jan2003'd,'30/360');
DATE()	返回当前日期的 SAS 日期值
DATEJUL(julian-date)	转换西洋 Julian 日期为 SAS 日期值
DATEPART(datetime)	返回 SAS 日期时间值的日期部分
DATETIME()	返回当前日期和时间

函 数	功能与举例
DAY(date)	返回 SAS 日期值为某月的某一日
DHMS(date,hour,minute,second)	返回由日期、小时、分钟和秒构成的 SAS 日期时间值
HMS(hour,minute,second)	返回由小时、分钟和秒构成的 SAS 日期时间值
HOUR(<time datetime>)	返回 SAS 日期时间或时间值的小时数
INTCK('interval',from,to)	返回时间间隔数字： INTCK('30','01jan2002'd,'01jan2003'd);
INTNX('interval',start-from,increment <'alignment'>)	按给定间隔推算日期、时间或日期时间值
JULDATE(date)	返回 SAS 日期的西洋 Julian 日期
MDY(month,day,year)	返回由年、月和日定义的 SAS 日期值： MDY('10','10','2002')
MINUTE(time datetime)	返回 SAS 日期值或日期时间值的分钟数
MONTH(date)	返回 SAS 日期值的月份
QTR(date)	返回 SAS 日期值的季度
SECOND(time datetime)	返回 SAS 时间值或日期时间值的秒数
TIME()	返回当前日的时间
TIMEPART(datetime)	返回 SAS 日期时间值的时间部分
TODAY()	返回当前日期的 SAS 日期值。同 DATE()
WEEKDAY(date)	返回 SAS 日期值为一周内的第几天
YEAR(date)	返回 SAS 日期值的年份
YRDIF(sdate,edate,basis)	返回两个日期之间的年数： yrdif('16oct1998'd,'16feb2003'd,'30/360'); yrdif('16oct1998'd,'16feb2003'd,'act/act');
YYQ(year,quarter)	返回由年和季定义的 SAS 日期值

3.4.2 应用举例

下面通过例子,列举出常用 SAS 日期函数在编程中的具体用法,供读者参考。

例 3.8 计算两个日期之间的天数。

```
data _null_;
sdate='01jan2002'd;
edate='01jan2003'd;
actual=datdif(sdate, edate, 'act/act');      /* 按每个月的实际天数算 */
days360=datdif(sdate, edate, '30/360');     /* 按每个月 30 天计算 */
put actual= days360=;
run;

data _null_;
```

```
actual=datdif('01jan2002'd, '01jan2003'd, 'act/act');
days360=datdif('01jan2002'd, '01jan2003'd, '30/360');
put actual= days360=;
run;
```

例中,两段程序的结果一样,actual=365, days360=360。

例 3.9 计算两个日期之间的年数。

```
data _null_;
sdate='16oct1998'd;
edate='16feb2003'd;
y30360=yrdif(sdate, edate, '30/360'); /* 30/360 规定求年数的标准 */
yactact=yrdif(sdate, edate, 'ACT/ACT');
yact360=yrdif(sdate, edate, 'ACT/360');
yact365=yrdif(sdate, edate, 'ACT/365');
put y30360= yactact= yact360= yact365=;
run;
```

结果显示:

```
y30360=4.333
yactact=4.3369
yact360=4.4
yact365=4.3397.
```

例 3.10 以日为单位计算当前日期的天数。

```
data;
x=date ();
y=today();
put x= y=;
run;
```

例中,DATE()和 TODAY()结果一样,都是返回当天的天数。当然,也可以用一定的日期格式表示它们。

```
data;
x=date ();
y=today();
format x y yymmdd10.;
put x=y=;
run;
```

例 3.11 以秒为单位计算当前的日期和时间。

```
data;
n=datetime();
put n=;
```

```

x=datetime ();
put x=;
format x datetime20.4;          /* datetime20.4 为日期时间变量输出格式 */

y=datetime ();
put y=;
format y nldatm20.;            /* nldatm20.为日期时间变量输出格式 */

m=time ();
put m=;

z=time ();
put z=;
format z hhmm8.2;
run;

```

结果显示:

```

n=1347837287.5
x=16sep02:23:14:47.500
y=16sep2002:23:14:48
m=83687.5
z=23:14.79

```

例 3.12 计算日期值所在的年季月以及处于某月的第几天。

```

data a;
set ResDat.stk000001 (obs=10);
keep date year qtr month day;
year=year (date);
qtr=qtr (date);
month=month (date);
day=day (date);
proc print noobs;
run;

```

例 3.13 计算当前时间的小时数和分钟数。

```

data;
h=hour (datetime ()); put h=;
m=minute (datetime ()); put m=;
run;

```

例 3.14 将日期时间值换算为以秒计数的计算方法。

```

data;
mdy=mdy (08,18,2001);
put mdy=;

```

```
format mdy yymmdd10.;

hms=hms (21,50,51);
put hms=;
x=21 * 60 * 60+ 50 * 60+51; put x=;

dhms=dhms (date (), 21,50,51);
put dhms=;

dhms=dhms (15263,21,50,51);
put dhms=;
y=15263 * 24 * 3600+x; put y=;
run;
```

结果显示：

```
mdy=2001-08-18
hms=78651
x=78651
dhms=1347832251
dhms=1318801851
y=1318801851
```

3.5 概率分布函数

SAS 系统分布和密度函数见附录 B。这里介绍 10 个常用概率分布函数。

3.5.1 标准正态分布

PROBNORM(X)

计算标准正态分布随机变量小于 X 的概率。

例 3.15 计算 3 个特殊的正态概率值。

```
data;
P1=probnorm(0);
P2=probnorm(1.96);
P3=probnorm(2.5758293);
put P1=P2=P3=;
run;
```

结果显示：

```
P1=0.5
```

P2=0.9750021049

P3=0.9949999999

3.5.2 卡方分布

PROBCHI (X, DF, NC)

计算自由度为 DF, 非中心参数为 NC 的 χ^2 分布随机变量小于 X 的概率。NC 的默认值为 0, 即中心的 χ^2 分布。自由度 DF 可以不是整数。

例 3.16 计算自由度为 88, 非中心参数为 11 的 χ^2 分布小于 31 的概率值。

```
data;  
P=1-Probchi(31, 88, 11);  
put P=;  
run;
```

3.5.3 伽马分布

PROBGAM (X, A)

计算形状参数为 A 的伽马分布随机变量小于 X 的概率。

例 3.17 计算形状参数为 5.2 的伽马分布小于 7.5 的概率。

```
data;  
P=Probgam(7.5, 5.2);  
put P=;  
run;
```

3.5.4 贝塔分布

PROBBETA (X, A, B)

其中: $0 \leq X \leq 1$, $A > 0$, $B > 0$ 。

计算形状参数为 A 和 B 的贝塔分布随机变量小于 X 的概率。

例 3.18 计算形状参数为 4 和 2.5 的贝塔分布小于 0.75 的概率。

```
data;  
P=Probbeta(0.75, 4, 2.5);  
put P=;  
run;
```

3.5.5 F 分布

PROBF (X, NDF, DDF, NC)