

哈尔滨工程大学出版社电脑丛书

Fox Pro

数据库组织和统计分析

李伟凯 蔡德利 著



哈尔滨工程大学出版社

TP311.13
1.55

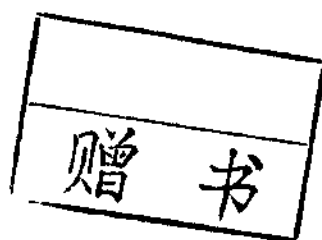
425454

FoxPro 数据库组织和统计分析

李伟凯 蔡德利 著



00425454



哈尔滨工程大学出版社

内 容 简 介

本书概述了 FoxPro 2.5 数据库组织和程序设计的基本内容,系统地介绍了试验资料的整理、假设测验、方差分析、回归分析、协方差分析、通径分析、主成分分析、聚类分析等多种数理统计分析的方法、原理以及实例,重编和新编了这些方法的 FoxPro 源程序。

本书可供农业工作者、大中专学生、计算机爱好者(特别是 FoxPro 程序设计人员)参考。

JS/51/07

FoxPro 数据库组织和统计分析

FoxPro Shujuku Zuzhi he Tongjin Fenxi

李伟凯 蔡德利 著

责任编辑 李 英

哈尔滨工程大学出版社出版发行

哈尔滨市南通街 145 号 哈工程大学 11 号楼

发行部电话(0451)2519328 邮编:150001

新华书店经销

哈尔滨工业大学印刷厂印刷

开本 787mm×1 092mm 1/16 印张 15 字数 500 千字

1998 年 12 月第 1 版 1998 年 12 月第 1 次印刷

印数: 1~1000 册

ISBN 7-81007-917-4

TP·82 定价: 30.00 元

前 言

农业科学一直被认为是一门描述性科学,研究手段比较落后,周期长,研究进程也较缓慢。自从有了微机和数据库后,人们可以把平时获得的资料随时存入计算机内,进而研究数据间错综复杂的关系,大大提高了农业科学的研究水平。如农业气象预报、产量预报、灾情预报等;建立农作物种植、品种资源等生产信息数据库;应用计算机的农业普查、农业规划、实时控制等。

过去应用计算机进行农业研究,在农业数据资料的统计分析方面应用较多,科研工作者利用 FORTRAN 或 BASIC 语言编制了大量的统计分析程序,新的 C 语言程序员也利用 C 语言编写统计程序。这些程序在一定程度上方便了计算机的应用,但也存在不少问题,主要表现在出错时调试困难;数据和处理数据的程序在一起,通用性差等。另外,大量新的复杂的统计分析方法的应用,很难无偿得到它们的计算程序,这些都给计算机在农业上的应用带来了障碍。

美国 Microsoft 公司 FoxPro 数据库系统的发展给计算机在农业上的应用带来了新的契机。FoxPro 2.5 版后,运行速度很快。由于采用了 Rushmore 优化技术,对大型数据库的操作这一优点更加突出。它还有强大的屏幕、报表、菜单生成器,对于不十分熟悉界面设计的农业应用工作者尤为实用。由于减少了界面开发工作量,人们很容易设计出漂亮的应用软件。过去人们不用数据库编制统计分析程序,是因为 dBase、FoxBase、Clipper 等的数学处理函数较少,不能满足科学计算要求的原因。现在 FoxPro 提供的命令和函数已经多达 600 多个,无论从性能、指标、理论和方法上都比以前的 xBase 系统高出许多,所以 FoxPro 广泛应用于商务、管理、信息业各领域。

基于以上特点,我们本着数据与代码分离的原则,用 FoxPro 设计了大量应用于农业的通用程序,以拓展 FoxPro 在农业上的应用。本书既是一本源程序集,也是一本 FoxPro 在农业上应用的实例集,读者可在数据管理基础上迅速建立决策分析系统。在教学上本书可作为数据库课程或农业生物统计课程的后续软材,提高计算机应用能力。

黑龙江八一农垦大学生物统计教研室张树光副教授、栽培教研室李金峰副教授、计算机中心许善详老师对本书的续写和出版给予了大力支持,张树光副教授审阅了全书,在此对他们表示衷心的感谢。

由于作者水平所限,时间仓促,书中难免错误之处,敬请广大读者批评指正。

作 者

1998, 12, 08

目 录

1 FoxPro 数据库组织	1
1.1 FoxPro 基础知识	1
1.1.1 性能指标	1
1.1.2 数据类型	1
1.1.3 变量	1
1.1.4 运算符和表达式	2
1.1.4.1 算术运算符	2
1.1.4.2 字符串运算符	2
1.1.4.3 关系运算符	2
1.1.4.4 逻辑运算符	2
1.1.5 命令结构	2
1.1.6 基本输入输出	3
1.1.7 流程控制	3
1.1.7.1 选择结构	3
1.1.7.2 循环结构	4
1.1.8 数组	4
1.1.9 过程和函数	5
1.2 数据库组织	7
1.2.1 数据库的建立、打开、关闭	7
1.2.1.1 建立数据库	7
1.2.1.2 打开数据库	9
1.2.1.3 关闭数据库	11
1.2.2 记录的添加、插入、修改和删除	11
1.2.2.1 添加记录	11
1.2.2.2 修改记录	12
1.2.2.3 插入记录	12
1.2.2.4 删除记录	13
1.2.3 记录的浏览	13
1.2.4 与外部文件的数据交换	14
1.3 FoxPro 界面要素	14
1.3.1 窗口	14
1.3.1.1 窗口的定义	14
1.3.1.2 窗口的激活	15
1.3.1.3 窗口的关闭和删除	15
1.3.2 菜单	16
1.3.3 屏幕	24
1.3.3.1 屏幕控制项	24
1.3.3.2 文件名的处理方法	27
2 试验资料的整理	29
2.1 试验资料的类型	29

2.2 次数分布	29
2.2.1 次数分布分析方法	29
2.2.2 次数分布源程序	30
2.3 基本特征参数	34
2.3.1 平均数	34
2.3.1.1 算术平均数	34
2.3.1.2 几何平均数	35
2.3.1.3 调和平均数	35
2.3.1.4 中位数	36
2.3.1.5 众数	36
2.3.2 变异数	36
2.3.2.1 样本方差	36
2.3.2.2 样本标准差	37
2.3.2.3 变异系数	37
2.3.3 基本特征参数源程序	37
2.4 统计数的分布	45
2.4.1 随机抽样	45
2.4.1.1 从有限总体中进行个体可重复的抽样	45
2.4.1.2 从有限总体中进行个体无重复的抽样	46
2.4.1.3 随机抽样源程序	46
2.4.2 正态分布	48
2.4.2.1 正态分布上概率	48
2.4.2.2 正态分布百分点	49
2.4.2.3 正态分布计算源程序	49
2.4.3 Beta (1/2 整倍数) 分布	52
2.4.3.1 贝塔分布函数和密度函数	52
2.4.3.2 贝塔分布百分点	53
2.4.3.3 贝塔分布计算源程序	53
2.4.4 学生氏 t 分布	56
2.4.4.1 t 分布上概率	56
2.4.4.2 t 分布百分点	57
2.4.4.3 t 分布计算源程序	57
2.4.5 F 分布	59
2.4.5.1 F 分布上概率	59
2.4.5.2 F 分布百分点	60
2.4.5.3 F 分布计算源程序	60
3 假设测验	63
3.1 单个样本平均数的假设测验	63
3.1.1 原理与实例	63
3.1.2 单个样本平均数假设测验源程序	65
3.2 两个样本平均数的假设测验	68
3.2.1 成组数据的比较	68
3.2.1.1 已知两样本总体方差或大样本($n>30$), 用 u 测验	68
3.2.1.2 总体方差未知, 且小样本情况下, 可用 t 测验	69
3.2.1.3 成组数据比较源程序	70

3.2.2 成对数据的比较	74
3.2.2.1 原理与实例	74
3.2.2.2 成对数据相比较源程序	75
3.3 样本百分数的比较	76
3.3.1 单个样本百分数的假设测验	76
3.3.2 两个样本百分数相比较的假设测验	77
3.3.3 样本百分数比较源程序	78
4 方差分析	82
4.1 方差分析的意义	82
4.2 方差分析的基本原理	82
4.2.1 自由度和平方和的分解	82
4.2.2 F 测验	83
4.2.3 多重比较	83
4.3 单向分组资料的方差分析	84
4.3.1 组内观测值相等完全随机资料的方差分析	84
4.3.2 组内观测值不等完全随机资料的方差分析	85
4.3.3 组内又分亚组(系统分组)资料的方差分析	86
4.3.4 单向分组资料的方差分析源程序	87
4.4 两向分组资料的方差分析	95
4.4.1 两向组内只有单个观察值完全随机资料的方差分析	95
4.4.2 两向组内有重复观测值完全随机资料的方差分析	96
4.4.3 两向分组资料的方差分析源程序	99
4.5 常用试验设计的方差分析	107
4.5.1 单因素随机区组试验的方差分析	107
4.5.2 拉丁方试验的方差分析	109
4.5.3 二因素随机区组试验的方差分析	110
4.5.4 常用试验设计方差分析源程序	113
5 双变数回归和相关	124
5.1 直线回归与相关	124
5.1.1 直线回归方程的建立	124
5.1.2 回归方程估计标准误	125
5.1.3 回归方程的显著性测验	125
5.1.4 线性相关分析	126
5.1.5 相关系数的显著性测验	126
5.2 非线性回归分析	127
5.2.1 $y=a+blgx$	128
5.2.2 $y=ab^x$	129
5.2.3 $y=ae^{bx}$	130
5.2.4 $y=axe^{bx}$	132
5.2.5 $y=ax^b$	133
5.2.6 $y=ae^{bx^2}$	134
5.2.7 $y=(a+bx)/x$	136
5.2.8 $y=1/(a+bx)$	137
5.2.9 $y=x/(a+bx)$	138

5.2.10 对称型 S 形曲线(正态累积函数).....	139
5.2.11 不对称型 S 形曲线.....	140
5.2.12 Logistic 生长曲线.....	141
5.3 双变数回归和相关分析源程序.....	144
6 协方差分析.....	150
6.1 协方差分析的意义.....	150
6.2 单向分组资料的协方差分析.....	150
6.2.1 自由度和乘积和的分解.....	150
6.2.2 协方差分析.....	151
6.2.3 回归系数的假设测验.....	151
6.2.4 处理平均数的回归矫正.....	152
6.2.5 矫正平均数的假设测验.....	152
6.2.6 单向分组资料协方差分析的源程序.....	155
6.3 两向分组资料的协方差分析.....	160
6.3.1 自由度和乘积和的分解.....	160
6.3.2 协方差分析.....	161
6.3.3 矫正平均数及差异显著性测验.....	161
6.3.4 两向分组资料的协方差分析源程序.....	163
7 多元回归与相关.....	169
7.1 多元线性回归分析.....	169
7.1.1 多元线性回归的数学模型.....	169
7.1.2 多元线性回归方程的求解.....	170
7.1.3 回归方程的显著性检验.....	170
7.1.4 回归系数的显著性检验.....	171
7.1.5 多元线性回归分析源程序.....	173
7.2 逐步回归分析.....	179
7.2.1 逐步回归分析的基本思想.....	179
7.2.2 逐步回归分析的计算步骤.....	179
7.2.2.1 准备阶段.....	179
7.2.2.2 逐步计算阶段.....	180
7.2.2.3 结尾阶段.....	180
7.2.3 逐步回归分析源程序.....	184
7.3 二次多项式回归分析.....	189
7.3.1 原理和实例.....	189
7.3.2 二次逐步回归.....	193
7.3.3 二次多项式回归分析源程序.....	194
8 通径分析.....	199
8.1 通径分析的意义.....	199
8.2 通径分析步骤.....	199
8.2.1 通径分析的基本原理.....	199
8.2.2 通径系数的计算.....	199
8.2.3 通径系数的显著性测验.....	200
8.3 通径分析源程序.....	202

9 主成分分析	205
9.1 总体主成分	205
9.2 样本主成分	206
9.3 主成分分析源程序	209
10 聚类分析	215
10.1 数据变换处理	215
10.2 距离系数	216
10.3 系统聚类法	216
10.4 系统聚类法源程序	218
参考文献	227

1 FoxPro 数据库组织

1978 年, 美国人 Wayne Ratliff 为了在家用计算机上监视足球场的情况, 设计了第一个数据库“VULCAN”, 这实际上是一个文件系统。后来, George Tate 说服 Ratliff 将这个系统的版权卖给他, 经过进一步开发变成 Ashton-Tate 公司的 dBASE II 2.0 版。经过该公司的大力宣传、再开发、维护和培训, 到 dBASE III 时, 已经成为微机上最受欢迎的数据库之一。

Ashton-Tate 公司的 dBASE 数据库软件升级速度较慢, 后来出现了一系列称为“xBASE”的兼容产品, 其中最突出的就是 Fox Software 公司 1985 年推出的 FoxBASE。该系统一问世就吸引了 dBASE 的大部分用户。后来, dBASE 和 FoxBASE 的一场版权官司, 导致两家公司分别被另外的公司收购。Fox 公司并入 Microsoft 公司, 使 Fox 数据库产品得到更好的发展, 其 FoxPro 2.5/2.6 for DOS 是 DOS 系统下速度最快的数据库软件之一, Windows 下的数据库 Visual FoxPro 已发展到 6.0 版。

本书适用于 FoxPro 2.5B for DOS。

1.1 FoxPro 基础知识

1.1.1 性能指标

FoxPro 采用关系型数据库模型, 每个数据库最多 254 个字段, 最多 10 亿条记录。在标准版中, 打开 DBF 最多 25 个, 扩展版中最多可达 225 个。

缺省内存变量的个数为 256 个, 最多内存变量标准版 3600 个, 扩展版 6500 个, 最多数组个数与最多内存变量数相同。

1.1.2 数据类型

FoxPro 中数据类型有下列几种:

(1) 字符类型(Character), 由中文字符、英文字符、数字字符及其他符号组成, 一个字符型字符最长 254 个字符。

(2) 数值类型(Numeric), 只能由字符、小数点和正负号构成, 每个数值型数据最长 20 位, 小数位数 0-16 位。

(3) 逻辑类型(Logic), 存储表示逻辑判断结果的值, 只能取表示真值的数据, 如 Y/N 或 T/F, 长度固定为一位。

(4) 日期类型(Date), 长度为 8, 有固定的格式, 如美国格式为 mm/dd/yy。可以用 SET DATE TO 语句设置日期类型。

(5) 备注类型(Memo), 长度不限。

(6) 浮点类型(Float), 使用方法与 Numeric 类型相似。

(7) 通用类型(General), FoxPro for Windows 下使用。

(8) 图片类型(Picture), FoxPro for Windows 下使用。

1.1.3 变量

FoxPro 中变量分为字段变量、内存变量、数组变量、系统变量四类。

字段变量是构成数据库结构的变量。

内存变量是一种临时变量，程序执行时供程序使用，程序执行完成后，这些变量将自动释放。按变量的有效期可将变量分为局部变量和全局变量，但 FoxPro 中还有一种特殊的变量称为区域变量，用 REGIONAL 来进行说明。

FoxPro 中的局部变量是指只能在当前程序模块中加以利用的变量，可以用 PRIVATE 关键字来进行说明，但 FoxPro 的局部变量与 C 语言或 QBASIC 中的局部变量在含义上稍有差异，即 FoxPro 的局部变量一旦定义，在当前模块以及当前模块调用的模块中均有效，本模块结束时才释放。

1.1.4 运算符和表达式

1.1.4.1 算术运算符

按优先级排列为

- () 分组括号
- **或^ 乘方
- *, /、% 乘、除、取余
- +, - 加、减

由数值型常数、变量、函数和算术运算符等构成的表达式为数值表达式。

1.1.4.2 字符串运算符

- +
- 完全连接运算符
- 不完全连接运算符
- \$ 包含运算符

由字符型常数、变量、函数和字符运算符构成的表达式为字符表达式。

1.1.4.3 关系运算符

- < 小于
- <= 小于等于
- = 等于
- == 恒等于
- > 大于
- >= 大于等于
- <>或!= 不等于

由字符表达式或数值表达式和关系运算符构成的表达式为关系表达式。

1.1.4.4 逻辑运算符

按优先级排列为

- .NOT. 逻辑非
- .AND. 逻辑与
- .OR. 逻辑或

由逻辑常数、变量、函数和逻辑运算符构成的表达式为逻辑表达式。

1.1.5 命令结构

FoxPro 有六百多个命令和函数，每条命令都有确定的格式，一般格式为：

命令动词 子句

如打开数据库：USE <数据库文件名> alias <别名>

在一些命令中，都有一个表示范围的子句，可有如下范围：

RECORD <N>	表示指定第N个记录
NEXT <N>	表示从当前记录向后N个记录
ALL	表示全部记录
REST	表示从当前记录至文件结束的所有记录

命令的详细信息可参照“FoxPro 2.5 命令与函数参考”。

1.1.6 基本输入输出

? | ?? <表示式>

计算表达式的结果并输出到屏幕上。相当于 BASIC 语言中的 PRINT 语句。使用单个“?”号时，先换行，表达式的值输出在屏幕在下一行。值用双问号“??”时，不换行，在当前光标位置输出表达式值。输出时采用电传打字方式，输出内容后，光标向后移动，当前行输出不下，换到下一行继续输出，至屏幕或窗口最后，要向上滚行。

可以输出各种类型的数据，如数值、字符串。

1.1.7 流程控制

1.1.7.1 选择结构

(1) 两路分支可用 IF 语句：

```
IF <逻辑表达式>
    <语句 1>
[ELSE
    <语句 2>]
ENDIF
```

两路分支语句的执行过程是，先判断逻辑表达式的值，若为真则执行语句 1，否则执行语句 2。语句 2 可以省略，当无语句 2 时，ELSE 也应同时省略。

(2) 多路分支可用 DO CASE 语句：

```
DO CASE
CASE <逻辑表达式 1>
    <语句 1>
[CASE <逻辑表达式 2>
    <语句 2>
.
.
.
CASE <逻辑表达式 N>
    <语句 N>]
[OTHERWISE
    <语句 N+1>]
ENDCASE
```

多路分支结构执行时分多种情况，即有多个 CASE 子句，当满足逻辑表达式 1 时，执行语句 1；满足逻辑表达式 2 时，执行语句 2，……所有逻辑表达式都不满足时，执行 OTHERWISE 后的语句 N+1。每当有一种情况满足，执行完该情况的语句后，都结束（跳出）DO CASE 语句。CASE 子句可以有若干个，OTHERWISE 子句可以省略。

1.1.7.2 循环结构

(1) 计数循环——FOR 循环:

```
FOR <内存变量>=<初值> TO <终值> [STEP <步长>]
    <语句体>
[EXIT]
[LOOP]
ENDFOR | NEXT <内存变量>
```

FOR 循环执行时, 首先将初值赋予控制循环的内存变量, 该赋值语句仅执行一次。然后判断内存变量是否超过终值, 若不超过则执行语句体, 该语句体也称为 FOR 循环的循环体。当循环体执行后, 内存变量加上步长值, 再进行下一轮循环。

循环体内可以出现 EXIT 语句, 用于终止循环。LOOP 语句用于强迫循环进入下一轮, LOOP 其后的语句本轮不再执行。EXIT 和 LOOP 语句常于 IF 语句搭配使用。

(2) 条件真循环——DO WHILE 循环:

```
DO WHILE <逻辑表达式>
    <语句体>
[EXIT]
[LOOP]
ENDDO
```

执行 DO WHILE 循环时, 首先判断逻辑表达式的值是否成立, 若为真, 则执行循环体, 否则不执行循环。循环体内也可以使用 EXIT 和 LOOP 语句, 意义与 FOR 语句中使用相同。

(3) 遍历数据库记录的 SCAN 循环:

```
SCAN [NOOPTIMIZE] [<范围>][FOR <逻辑表达式>][WHERE <逻辑表达式>]
    <语句体>
[EXIT]
[LOOP]
ENDSCAN
```

SCAN 只能用于针对某一数据库进行循环, 从数据库的第一记录起, 每次跳到下一记录, 可实现对数据进行遍历。

1.1.8 数组

在 FoxPro 中, 数组被定义为一组变量的集合, 它的使用非常灵活。数组中的元素可以是不同的数据类型, 在运行时也可以随时发生变化。

可以用 PUBLIC 或 DECLARE 来定义全程数组, 用 DIMENSION 来定义局部数组, 用 REGIONAL 定义区域数组。如:

```
DIMENSION aa(10)
PUBLIC bb[20]
```

在 FoxPro 中定义数组既可以使用圆括号, 也可以使用方括号, 但两者不能混用。

若定义数组 DIMENSION aa[30], 又定义变量 aa=20, FoxPro 会认为它们是同一变量, 编辑时并不给提示, 运行时可能会发生错误。

FoxPro 中允许数组重新定义, 如:

```
DIMENSION aa[3]
```

```
DIMENSION aa[10]
DIMENSION aa[5]
```

数组一旦定义后，该数组的每个元素均赋值为逻辑类型的.f.（假），但数组元素被认为是一个内存变量，所以可以象其他内存变量一样，随时赋以其他的值来改变数据类型或内容。当这个数组在使用完后，可以值用如下一些命令释放数组所占的内存：

CLEAR ALL	清除所有项
CLEAR MEMORY	释放内存中的变量和数组
RELEASE <内存变量表>	释放内存变量
RELEASE ALL [LIKE <通配符> EXCEPT <通配符>]	释放指定的变量

1.1.9 过程和函数

一个大程序是由若干过程和函数构成的，在 FoxPro 中不论过程还是函数，均为由若干语句构成的能完成一定相对独立功能的程序段。当需要一个返回值时，我们往往把程序段设计成函数，不需要返回值时设计成过程，在 FoxPro 中，有将过程和函数统一为函数的趋势，就像 C 语言一样。

FoxPro 中的函数分为系统函数和用户自定义函数，系统函数有几百个。用户自定义函数是用户根据需要编写的函数，其格式如下：

```
function 函数名称
    [<parameters 形式参数表>]
    .
    .
    .
    [<return 返回值>]
```

函数定义由 function 开始，之后是函数名称，其他程序调用该函数就通过名称进行，这部分为函数头。第二行为参数说明语句，由 parameters 开始，之后是形式参数表，通常函数参数传递采用的是传值法。在函数的最后可以有一个返回语句 return。

若函数不需要设计返回值，return 语句可以不写，这就与由 procedure 开始定义的过程一样：

```
procedure 过程名称
    [<parameters 形式参数表>]
    .
    .
    .
    [<return>]
```

在 FoxPro 中，通常过程参数传递采用的是传址法，即调用程序将实际参数的地址传递给过程的形式参数，形式参数在过程中参与各种计算等，其值随着程序的执行而变化，实际是修改了实际参数的值，当过程结束时，实际参数的值是最近被修改的值。

若定义了函数: function funName, 则调它的方法为: m.memoryVar=funName(), 或当我们不需要返回值时，用=funName()方法调用，这就如执行过程一样，所以我们可以用如下方法调用：

do funName （若有参数时可以加上 with <参量表>）

所以，我们习惯将 FoxPro 的过程也定义为函数的形式，这与 C 语言是一致的。

FoxPro 也允许将数组通过数组名称传递给函数或过程，就如同进行变量传递一样。FoxPro 提供了大量的函数，供我们在编程时使用，下面简要列举一些比较常用的函数：

at(<字符表达式 1>,<字符表达式 2>,[<数值表达式>])	检索字符表达式 1 在字符表达式 2 中第<数值表达式>次出现的位置
len(<字符表达式>)	测定字符表达式的长度
between(<被测表达式>,<下限>,<上限>)	测定被测表达式是否在上下限之间
file(<字符表达式>)	测定字符表达式代表在文件名是否存在
empty(<表达式>)	测定表达式是否为空
type(<字符表达式>)	测定字符表达式的类型
substr(<字符表达式>,<数值表达式 1>,[<数值表达式 2>])	从字符表达式的数值表达式 1 位置开始截取数值表达式 2 长度的子串
alltrim(<字符表达式>)	去掉字符表达式的前后空格
stuff(<字符表达式 1>,<数值表达式 1>,<数值表达式 2>,<字符表达式 2>)	从字符表达式 1 的数值表达式 1 位置起用字符表达式 2 替换, 长度为数值表达式 2
str(<数值表达式 1>,[<数值表达式 2>,[<数值表达式 3>]])	将数值表达式 1 转换为字符串, 转换后长度为数值表达式 2, 精度为数值表达式 3
evaluate(<字符表达式>)	计算表达式并返回结果
exp(<数值表达式>)	e 的指数计算
log(<数值表达式>)	以 e 为底的自然对数
log10(<数值表达式>)	以 10 为底的常用对数
sqrt(<数值表达式>)	计算平方根
rand([<数值表达式>])	以数值表达式为种子 0-1 间的随机数
val(<数字字符串>)	将字符串转换成数值
int(<数值表达式>)	对数值进行取整
abs(<数值表达式>)	求数值的绝对值
max(<表达式 1>,<表达式 2>,...)	求最大值
min(<表达式 1>,<表达式 2>,...)	求最小值
sys(3)	返回单一文件名
wexist(<字符表达式>)	测定字符表达式指定的窗口是否存在
wrows([<窗口名>])	返回窗口高度
wcols([<窗口名>])	返回窗口宽度
dbf([<别名> <工作区>])	返回数据库文件名称
used([<工作区> <别名>])	指定工作区或别名是否打开
fcoun([<工作区> <别名>])	返回指定工作区(别名)数据库的字段数
field(<字段序号>[,<工作区> <别名>])	返回数据库的指定字段的名称
alias([<工作区> <别名>])	返回数据库别名
select([0 1])	返回当前工作区号(0 缺省)或没有使用的最大工作区号(1)
reccount([<工作区> <别名>])	返回数据库最大记录号
recno([<工作区> <别名>])	返回当前记录号
delete([<工作区> <别名>])	返回记录的删除状态

1.2 数据库组织

数据是一种物理符号的序列，它是用来表达事物的各种属性的。数据本身亦有数据类型和数据值两种属性，相同类型的数据记录事物的同一性质。

数据经过提炼和加工成为有用的信息，我们可以建立数据库来保存它们，进而还可以对其进行更深一步的处理，所以需要编写对数据库进行分类、加工、计算、检索、传输的管理系统，数据库和数据库管理系统构成了数据库系统。

在 FoxPro 的支持下，很容易实现数据库的建库、输出、修改操作，所以我们想到用它来管理我们日常获得的数据信息，这是数据库组织过程中完成工作。因为还需要用各种统计方法来处理数据库中的数据信息，所以根据统计方法的原理来编制处理程序。将数据库组织部分和统计分析部分放到一起能够顺利完成信息的采集到整理加工的全部过程，所以我们设计了一套称为 fSTAT 的数据库组织和统计分析软件，贯穿于本书的就是这套软件的设计原理、使用实例和源程序。

1.2.1 数据库的建立、打开、关闭

1.2.1.1 建立数据库

进行数据库操作的第一步就是建立数据库，为数据库确定适当的结构，可以通过系统菜单 FILE 下的 NEW 菜单项来建立数据库，也可以通过 CREATE 命令来建立。

FoxPro 数据库最多支持 254 个字段，支持字符、数值、日期、逻辑、备注等数据类型。建立数据库就是确定的个数，以及字段的名称、类型、长度和小数位数。

在 fSTAT 中，我们建立了一个用于保存系统工作参数的数据库，称为 fSTAT 系统配置数据库，文件名为 FSTATCFG.DBF，其结构和各字段的意义如下：

```
Structure for database: D:\FSTAT\FSTATCFG.DBF
Number of data records:      1
Date of last update   : 05/27/98
Code Page             : 0
```

Field	Field Name	Type	Width	Dec	Index	Collate
1	DATABASE	Character	48			上次打开的数据库
2	NUMIC	Numeric	4			数据库中的数值型字段数
3	OUTFILE	Character	48			结果输出文件
4	OUTWID	Numeric	2			输出数据项占据的宽度
5	OUTDEC	Numeric	2			数值输出的小数位数
6	OUTALIGN	Character	6			输出数据项对齐方式
** Total **			111			

在 fSTAT 中管理数据库的第一步也是使用“数据库[F]”菜单下的“创建[N]”命令建立新数据库，操作与 FoxPro 本身完全一样。源程序如下：

```
*-----*
*
* function createProc                                创建数据库菜单过程
*
*-----*
function createProc
    private m.fileSpec,m.mark
```



```

push key clear
m.fileSpec=putfile('建立数据库:', 'default', 'dbf', '创建')

m.mark=.t.
if empty(m.fileSpec)                                && 未输入建立数据库之文件名
    m.mark=.f.
else
    if file(m.fileSpec)                            && 文件已存在
        if warnDialog('文件已存在, 是否继续?', '', '覆盖[\<O];取消[\<C]')=2
            m.mark=.f.                            && 取消, 退出
        endif
    endif
endif

if m.mark
    if used('maindbf')                            && 当前已有打开的数据库
        select maindbf
    else
        select 0
    endif
    create (m.fileSpec)
    if used()                                        && 已在建立新数据库
        m.fileSpec=dbf()                          && 新数据库文件名称
        use
        do setTableName with m.fileSpec
    endif
    do openTable                                    && 打开表
endif
pop key
if !wexist('browWindow')
    activate menu _msysmenu pad _mainmenu1
endif
return

```

上述程序中, 函数 setTableName 的作用是将要打开的数据库写入 fSTAT 系统配置数据库 (STATCFG.DBF) 的 DATABASE 字段中, 系统配置文件在程序运行时已打开, 别名为 sysConfig。

setTableName 函数数据程序为:

```

*-----
*
*  function setTableName                            将要打开的数据库名存入系统配置
*
*-----
function setTableName
    parameter m.fileSpec
    private m.oldArea

```