**11. 对数据做简单的描述统计**

1. **使用proc means描述数据**

用proc means过程步，可以对数据做简单的描述统计，包括：非缺失值个数、均值、标准差、最大值、最小值等。

基本语法：

PROC MEANS data = 数据集 <*可选项>*;

VAR 变量列表;

CLASS 分组变量;

<BY 变量;>

<WEIGHT 变量;> （加权平均的权数）

<FREQ 变量;> （相应观测出现的频数）

说明：（1）可选项 “MAXDEC = n”用来指定输出结果的小数位数；

（2）默认是对数据集的所有数值变量的非缺失值做描述统计，若想包含缺失值，加上可选项“MISSING”；

（3）VAR语句指定要做描述统计的变量；CLASS语句指定按分组变量对数据进行分组分别做描述统计；BY语句同CLASS语句（需要事先按BY变量排好序）；

（4）默认输出非缺失值个数、均值、标准差、最大值、最小值；也可以自己指定需要输出的描述统计量：

MAX ——最大值；

MIN——最小值；

MEAN——均值；

MEDIAN——中位数；

MODE——众数；

N——非缺失值个数；

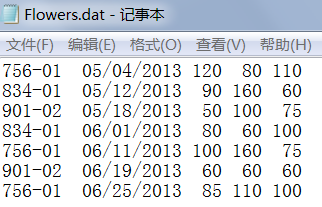
NMISS——缺失值个数；

RANGE——极差；

STDDEV——标准差；

SUM——累和；

**例1** 鲜花销售的数据（C:\MyRawData\Flowers.dat），变量包括顾客ID，销售日期，petunias，snapdragons，marigolds三种花的销量：



读取数据，计算新变量销售月份month，并使用proc sort按照月份排序，并使用proc means的by语句来按照月份描述数据。

代码：

**data** sales;

infile 'c:\MyRawData\Flowers.dat';

input CustID $ @**9** SaleDate MMDDYY10. Petunia SnapDragon Marigold;

Month = MONTH(SaleDate);

**proc** **sort** data = sales;

by Month;

/\* Calculate means by Month for flower sales; \*/

**proc** **means** data = sales MAXDEC = **0**;

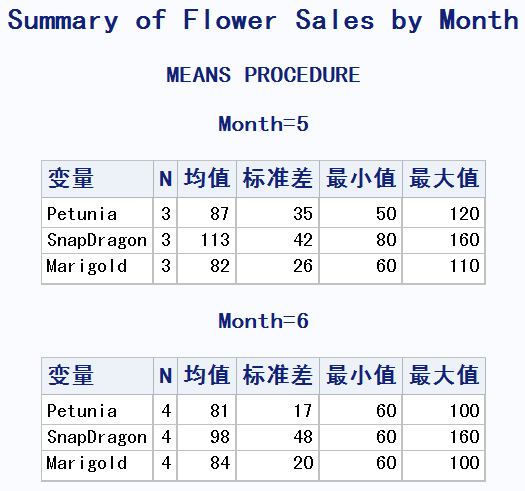
by Month;

var Petunia SnapDragon Marigold;

title 'Summary of Flower Sales by Month';

**run**;

运行结果：



**（二）使用统计量**

有时候需要将统计量存入新数据集，以便进一步做数据分析，或者与原数据集合并。

**一、将统计量存入新数据集**

可以用ODS（Output Delivery System）系统或者OUTPUT语句，下面介绍用OUTPUT语句。

语法：

OUTPUT OUT = *数据集名 统计量名(变量列表)=新列名*;

示例：

**PROC** **MEANS** DATA = zoo NOPRINT;

VAR Lions Tigers Bears;

OUTPUT OUT = zoosum MEAN(Lions Bears) = LionWeight BearWeight;

**RUN**;

**例2** 仍然是 **例1**的鲜花数据，读取数据，按照CustomerID排序，使用proc means过程，将mean和sum值存入新数据集totals中。

代码：

**data** sales;

infile 'c:\MyRawData\Flowers.dat';

input CustID $ @**9** SaleDate MMDDYY10. Petunia SnapDragon Marigold;

**proc** **sort** data = sales;

by CustID;

/\* Calculate means by CustomerID, output sum and mean to new data set; \*/

**proc** **means** NOPRINT data = sales;

by CustID;

var Petunia SnapDragon Marigold;

output out = totals

mean(Petunia SnapDragon Marigold) = MeanP MeanSD MeanM

sum(Petunia SnapDragon Marigold) = Petunia SnapDragon Marigold;

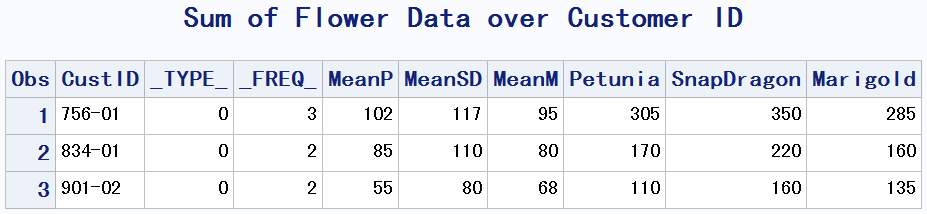
**proc** **print** data = totals;

title 'Sum of Flower Data over Customer ID';

format MeanP MeanSD MeanM **3.**;

**run**;

运行结果：



程序说明：

（1）“NOPRINT”告诉SAS不产生任何输出结果，因为已经存入新数据集zoosum;

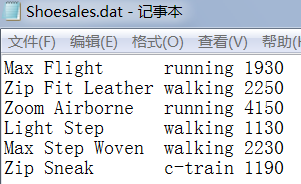
（2）保存统计量的新数据集，包括想要的统计量、BY/CLASS语句的变量、\_TYPE\_、\_FREQ\_;

（3）BY语句变量有3个水平：756-01、834-01、901-02，结果是3个观测，CLASS类似。

**二、合并统计量到原数据集**

1. 前面讲到将按“BY变量”分组统计量保存为统计量数据集，进一步可以用MERGE语句，将统计量数据集按照公共的“BY变量”，通过一对多匹配合并到原数据集。

**例3** 运动鞋经销商的销售数据（C:\MyRawData\Shoesales.dat），变量包括风格名称、运动类型、第四季度销量：



市场经理想要一个报表，每种风格运动鞋占该运动类型销量中的百分比。

代码：

**data** shoes;

infile 'c:\MyRawData\Shoesales.dat';

input Style $ **1**-**15** ExerciseType $ Sales;

**run**;

**proc** **sort** data = shoes;

by ExerciseType;

**run**;

/\* Summarize sales by ExerciseType and print; \*/

**proc** **means** NOPRINT data = shoes;

var Sales;

by ExerciseType;

output out = summarydata sum(Sales) = Total;

**run**;

**proc** **print** data = summarydata;

title 'Summary Data Set';

**run**;

/\* Merge totals with the original data set; \*/

**data** shoesummary;

merge shoes summarydata;

by ExerciseType;

Percent = Sales / Total \* **100**;

**run**;

**proc** **print** data = shoesummary;

by ExerciseType;

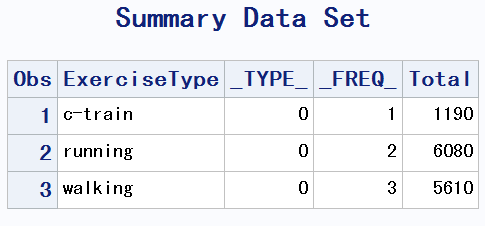
id ExerciseType;

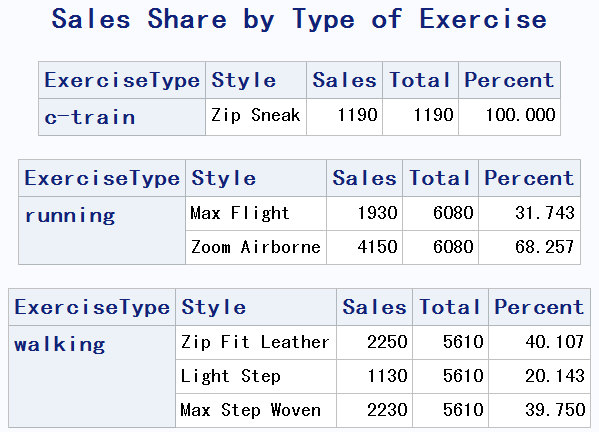
var Style Sales Total Percent;

title 'Sales Share by Type of Exercise';

**run**;

运行结果：





2. 若统计量是全体观测的汇总统计量（无BY变量做分组统计），此时，就不能直接采用一对多匹配（没有共同“BY变量”）合并。但可以用SET语句实现。

语法：

DATA *新数据集*;

IF \_N\_ = 1 THEN SET 汇总*统计量数据集*;

SET *原数据集*;

注：“汇总统计量数据集”只有一条观测值。创建新数据集时，第一次迭代循环（\_N\_=1）就读入该观测值，并一直为后续读入“原数据集”保留它（相当于RETAIN语句）。该语法机制也可用于：没有匹配变量情况下，将一条观测与多条观测合并。

**例4** 仍是**例3**的数据，市场经理想要一个报表，每种风格运动鞋占总销量的百分比。

代码：

**data** shoes;

infile 'c:\MyRawData\Shoesales.dat';

input Style $ **1**-**15** ExerciseType $ Sales;

**run**;

\* Output grand total of sales to a data set and print;

**proc** **means** NOPRINT data = shoes;

var Sales;

output out = summarydata sum(Sales) = GrandTotal;

**RUN**;

**proc** **print** data = summarydata;

title 'Summary Data Set';

**run**;

\* Combine the grand total with the original data;

**data** shoesummary;

if \_N\_ = **1** then set summarydata;

set shoes;

Percent = Sales / GrandTotal;

**run**;

**proc** **print** data = shoesummary;

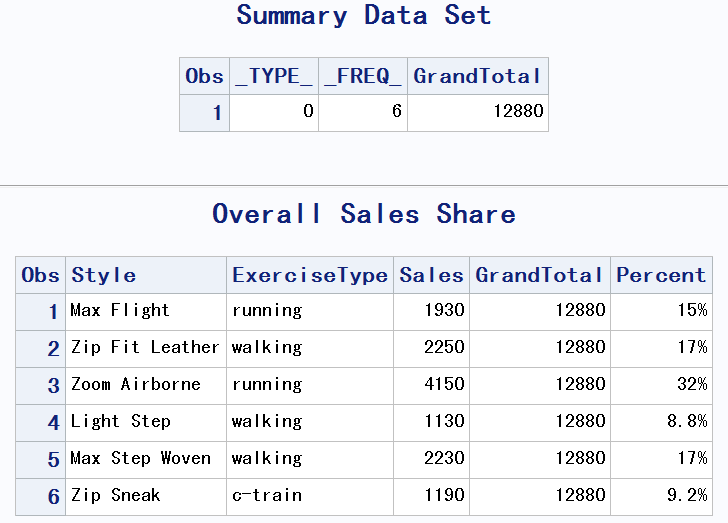
var Style ExerciseType Sales GrandTotal Percent;

format Percent PERCENT.2;

title 'Overall Sales Share';

**run**;

运行结果：



**（三）使用proc freq为数据计数**

用proc freq过程步可以对数据集的变量计算频数，再用tables命令以表格的形式输出：只输出一个变量（单向表）；输出两个变量（双向表）；输出多个变量（交叉表）。

频数表也可以用来检查错误数据。

语法：

PROC FREQ data = 数据集;

TABLES *变量组合 </可选项>*;

注：（1）关于变量组合：一个变量的单向表，用“TABLES 变量;”；两个变量的双向表，用“TABLES 变量1 \* 变量2;”；

（2）常用的可选项有：

LIST——用list形式打印交叉表（而不是网格）；

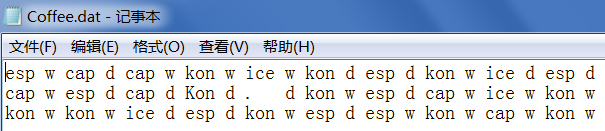
MISSING——频数统计量中包含缺失值；

NOCOL——强制在交叉表中不打印列百分比；

NOROW——强制在交叉表中不打印行百分比；

OUT=data-set——输出数据集；

**例5**咖啡店的销售数据（C:\MyRawData\Coffee.dat），记录了销售的咖啡种类（cappuccino, espresso, kona, iced coffee），以及每次购买的顾客是打包还是原地就饮：



读入数据，计算两个变量的频数。

代码：

**data** orders;

infile 'c:\MyRawData\Coffee.dat';

input Coffee $ Window $ @@;

\* Print tables for Window and Window by Coffee;

**proc** **freq** data = orders;

tables Window Window \* Coffee;

\* 输出两个表：Window的单向表、Window 和 Coffee 双向表;

**run**;

运行结果：



程序说明：

（1）默认缺失值不计入频数，使用可选项MISSING可以包含缺失值；

（2）观察表2，可以发现Kon是错误数据，应该是kon.

**（四）使用proc tabulate生成表格报表**

用proc tabulate过程步可以生成表格报表，相当于给普通输出报表做个“礼盒包装”，让报表更美观好看。

该过程还可以计算、加入、输出统计量。

基本语法：

PROC TABULATE <MISSING> data = 数据集 <指定格式>;

VAR *指定作统计分析的变量*;

CLASS *分组变量*; （按变量的值对数据做分组，分别处理）

TABLE *分页变量, 分行变量, 分列变量 </可选项>*;

说明：（1）TABLE语句定义表格形式：页维度、行维度、列维度；指定一个变量默认是列形式，指定两个变量默认是行列形式；

（2）tabulate可以加上如下的关键词（用在TABLE变量中计算各统计量）：

ALL——增加一行、一列或一页，显示总数目；

MAX——最大值；

MIN——最小值；

MEAN——平均值；

MEDIAN——中位数；

N——非缺失值个数；

NMISS——缺失值个数；

P90——90%分位数；

PCTN——某类的观测值百分数；

PCTSUM——某类值总和的百分数；

STDDEV——标准差；

SUM——求和；

（3）TABLE语句中，可以对变量做连接、交叉，或者它们的组合：

“连接”示例： TABLE Locomotion Type ALL;

“交叉”示例： TABLE MEAN \* Price;

“组合”示例： TABLE PCTN \*(Locomotion Type);

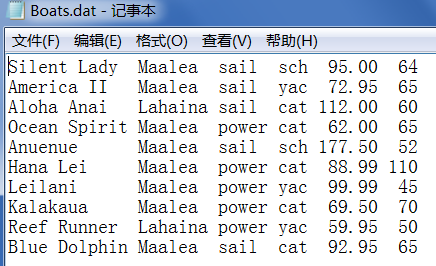
（4）可以在proc步用可选项<指定格式>（全局），指定所有数据的输出格式，例如，

**PROC** **TABULATE** DATA = boats FORMAT = COMMA10.0;

（5）TABLE语句的可选项：/BOX=’str’，可以给表格报表的左上角方框输出“说明性短句”；/MISSTEXT=’str’，可以给表格中的缺省数据指定一个“值”；例如，

TABLE Region, MEAN\*Sales / BOX='Mean Sales by Region' MISSTEXT='No Sales';

**例6** 关于船的数据（C:\MyRawData\Boats.dat），变量包括名称、港口、动力方式、类型、价格、船长（英尺）：



读入数据生成二维（列×行）表格报表：变量Locomotion和均值Mean分别作为报表的行和列，并增加一行、一列“计算总数目”（ALL关键词），并使用BOX和MISSTEXT选项增加表格的可读性。

代码：

**data** boats;

infile 'c:\MyRawData\Boats.dat';

input Name $ **1**-**12** Port $ **14**-**20** Locomotion $ **22**-**26** Type $ **28**-**30** Price **32**-**37** Length **39**-**41**;

**run**;

\* PROC TABULATE report with options;

**proc** **tabulate** data = boats FORMAT = DOLLAR9.2;

class Locomotion Type;

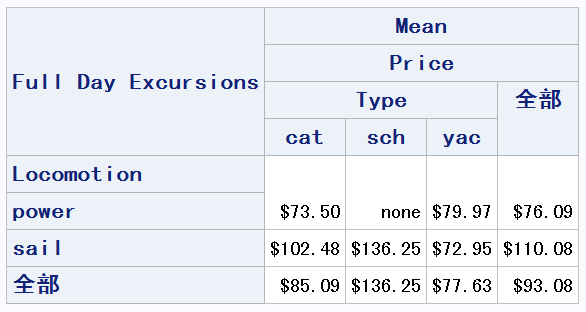
var Price;

table Locomotion ALL, MEAN\*Price\*(Type ALL) /BOX='Full Day Excursions' MISSTEXT='none';

title;

**run**;

运行结果：



程序说明：

（1）ALL关键词，英文版SAS软件中，显示是“All”而不是“全部”（可能是软件汉化的bug）；若要修改为过来，可以将代码中的table行，替换为：

table Locomotion ALL='All', MEAN\*Price\*(Type ALL='All') /BOX='Full Day Excursions' MISSTEXT='none';

（2）若想去掉“Locomotion”、合并“Mean”“Price”“Type”单元格使得表格更紧凑，可以用“将Table变量赋值为空字符的形式”实现：

**data** boats;

infile 'c:\MyRawData\Boats.dat';

input Name $ **1**-**12** Port $ **14**-**20** Locomotion $ **22**-**26** Type $ **28**-**30** Price **32**-**37** Length **39**-**41**;

**run**;

\* Changing headers;

**proc** **FORMAT**;

value $typ 'cat' = 'catamaran'

'sch' = 'schooner'

'yac' = 'yacht';

**proc** **tabulate** data = boats FORMAT=DOLLAR9.2;

class Locomotion Type;

var Price;

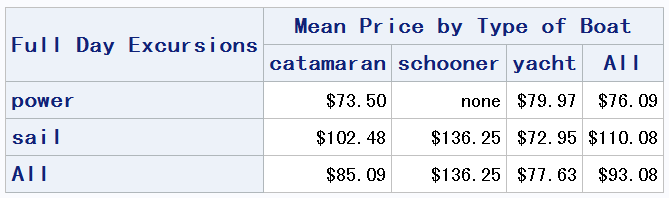
format Type $typ.;

table Locomotion='' ALL='All', MEAN=''\*Price='Mean Price by Type of Boat'\*(Type='' ALL='All') /BOX='Full Day Excursions' MISSTEXT='none';

title;

**run**;

运行结果：



注：table语句，是用赋值为空字符的形式去掉Locomotion、MEAN、Type单元格，并将Price单元格换成“Mean Price by Type of Boat”；有时候当行顶部被赋为空字符时，会留下一个空白空格，可以用/ROW=FLOAT选项强制去除它，示例：

TABLE MEAN=''\*Sales='Mean Sales by Region', Region='' /ROW=FLOAT;

（3）也可以不在proc步中指定全局格式，而是为不同的数值变量指定不同的输出格式，示例：

TABLE Locomotion ALL, MEAN \* (Price\*FORMAT=DOLLAR7.2 Length\*FORMAT=**2.0**) \* (Type ALL);