SAS APPLICATIONS

Mervyn Marasinghe

SAS Example A10

data sales;
infile "U:\Documents\..\sales.txt";
format Month monyy5. Revenue dollar12.2;
run;
proc sort;
  by Region State Month;
run;
proc print;
run;
proc print label;
  by Region State ;
  format Expenses dollar10.2 ;
  label State= State Month= Month Revenue= "Sales Revenue" Expenses="Overhead Expenses";
  id Region State;
  sum Revenue Expenses;
  sumby Region;
  title " Sales report by State and Region";
run;

Output Delivery System (ODS)

- ODS allows flexibility in presenting output from SAS procedures
- ODS can arrange output in more presentable ways
- It also can create output in a variety of formats (called destinations)

Examples of currently available ODS destinations:
- LISTING: produces traditional monospace output
- PS or PDF: output that is formatted for a high-resolution printer such as PostScript or PDF
- HTML: output that is formatted in various markup languages such as HTML
- RTF: output that is formatted for use with Microsoft Word

Traditionally, results of a SAS procedure were displayed in the output window in the LISTING format.
Use the Results tab in the Preferences window to change HTML to LISTING format.
Current Default Settings in SAS 9.3

- The default destination in the SAS windowing environment is HTML (changed from traditional LISTING destination)
- The default style for the HTML destination is HTMLBlue
- This new style is an all-color style that is designed to integrate tables and modern statistical graphics
- In the past, SAS users used SAS/Graph to produce high quality graphics. We’ll call these traditional SAS graphics.
- **Template-based graphics** produced by ODS Graphics (different from traditional SAS graphics) is now used for producing graphs in most SAS procs.
- Previously, in the SAS Windowing environment, ODS GRAPHICS ON statement was used to enable (and ODS GRAPHICS OFF statement to disable) ODS Graphics

SAS ODS Example 1

```sas
data oranges;
  input Variety $ Flavor texture Looks;
  Total=Flavor+Texture+Looks;
  datalines;
  navel 9 8 6
  temple 7 7 7
  valencia 8 9 9
  mandarin 5 7 8
;proc sort data=oranges;
  by descending Total;
run;
ods rtf file="U:\Documents\..\oranges.rtf";
proc print data=oranges;
  title 'Taste Test Results for Oranges';
run;
ods rtf close;
```

PROC MEANS

PROC MEANS options ;
- CLASS variables ;
- VAR variables ;
- TYPES request(s);
- WAYS list;
- BY variables ;
- FREQ variable ;
- WEIGHT variable ;
- ID variables ;
- OUTPUT OUT=SAS_dataset statistics ;

Options: DATA= , PRINT, MAXDEC= , FW= , MISSING, NWAY, IDMIN, DESCENDING, ORDER= , VARDEF= , statistics

```
MAXDEC = n no. of decimal places (2)
FW = n field width (12)

ORDER=FREQ, DATA, INTERNAL, FORMATTED
VARDEF=DF, N, WGT, WDF
```

PROC MEANS (continued)

Statistics:

```
N, NMISS, MEAN, STD, MIN, MAX, RANGE,
SUM, VAR, USS, CSS, CV, STDERR, T, PRT,
SUMWGT
```

Example of OUTPUT statement ;
```
output out=stats mean=Av_Ht  Av_Wt
  stderr=SE_Ht SE_Wt;
```
SAS Example B5

```sas
data biology;
input Id Sex $ Age Year Height Weight;
datalines;
7389   M  24  4  69.2  132.5
3945 F  19  2  58.5  112.0
4721 F  20  2  65.3  98.6
...
6327 M  20 1 70.2  135.4
8472 F  20 4  66.8  142.6
4875 M  20 1 74.2  160.4
;
proc means data=biology  fw=8 maxdec=3;
class Year Sex;
var Height Weight;
output out=stats mean=Av_Ht   Av_Wt stderr=SE_Ht SE_Wt;
run;
proc print data=stats;
title "Biology Class Data Set: Output Statement";
run;
```

PROC UNIVARIATE

```sas
PROC UNIVARIATE options ;
   VAR variables ;
   CLASS variable-1 <(v-options)>;
       variable-2 <(v-options)>;
       ...< / KEYLEVEL= value1 | ( value1 value2 ) >;
   BY variables ;
   ID variable ;
   OUTPUT < OUT=SAS-data-set >
       < keyword_1=names...keyword_k=names >
       < percentile-options >;

Some Options: DATA = , NOPRINT, PLOT, FREQ, NORMAL, ALPHA=, CIBASIC (TYPE= ALPHA=), MU0= , TRIM= (TYPE= ALPHA=) , PCTLDEF=, VARDEF=,
PCTLDEF=1,2,3,4 OR 5 (5 methods of computing percentiles)
VARDEF=DF, N, WEIGHT, WDF (divisor for computing variance)
NORMAL
   computes the Shapiro-Wilk statistic W if n ≤ 2000 or
   the Kolmogorov-Smirnov statistic D if n > 2000
```

PROC UNIVARIATE (continued)

```sas
TYPE= LOWER, UPPER, TWOSIDED
   (specify type of confidence intervals)
TRIM= list of integers or fractions specifying amount of trimming
ALPHA=0.05 (for both CIBASIC and TRIM specifications)

Class Options: The v-options are MISSING or ORDER=

ORDER = FREQ, DATA, INTERNAL, FORMATTED
   specifies how levels of the variable are ordered in the output

Some Keywords for use with OUTPUT statement:
N, NMISS, NOBS, MEAN, SUM, SD, VAR, SKEWNESS,
KURTOSIS, SUMWGT, MAX, MIN, RANGE, Q3, MEDIAN,
Q1, ORANGE, P1, P5, P10, P90, P95, P99, MODE,
SIGNRANK, NORMAL, PCTLNAME= , PCTLPTS= , PCTLPRE=

Example:
   proc univariate data=survey;
       class county;
       var acreage rainfall;
       output out=new mean=ave1 ave2 var= var1 var2;
   run;
```

SAS ODS Example 2

```sas
data biology;
input Id Sex $ Age Year Height Weight;
datalines;
7389 M 24 4 69.2 132.5
3945 F 19 2 58.5 112.0
4721 F 20 2 65.3 98.6
...
6327 M 20 1 70.2 135.4
8472 F 20 4 66.8 142.6
4875 M 20 1 74.2 160.4
;
ods rtf file="U:\Documents\progB2_out.rtf";
ods select BasicMeasures Quantiles TestsForNormality;
proc univariate data=biology Normal;
   var Height;
   title "Biology class: Analysis of Height Distribution";
run;
ods rtf close;
```
SAS Example C12 (simplified)

data rainfall;
input Control Seeded @@;
Logseed = log10(seeded);
label Logseed="Log of Seeded Rainfall";
datalines;
1202.6 2745.6 830.1 1697.8 372.4 1656.0 345.5 978.0 321.2 703.4
244.3 489.1 163.0 430.0 147.8 334.1 95.0 302.8 87.0 274.7
81.2 274.7 68.5 255.0 47.3 242.5 41.1 200.7 36.6 198.6
29.0 129.6 28.6 119.0 26.3 118.3 26.1 115.3 24.4 92.4
21.7 40.6 17.3 32.7 11.5 31.4 4.9 17.5 4.9 7.7
1.0 4.1 ;
proc univariate data=rainfall;
var Logseed;
probplot Logseed/normal(mu=est sigma=est);
run;

PROC TABULATE

PROC TABULATE options ;
■ CLASS variable ;
■ VAR variable ;
■ FREQ variable ;
■ WEIGHT variable ;
■ BY variable ;
■ TABLE expression, expression, ...
   / options ;
■ KEYLABEL keyword=‘text’ ...

■ Options: DATA= , MISSING, FORMAT= ,
   ORDER= , FORMCHAR( ) = ‘ ’,
   NOSEPS, DEPTH=

   FORMAT = any valid SAS or user defined format
   (BEST12.2)
   ORDER = FREQ, DATA, INTERNAL,FORMATTED
   FORMCHAR=|- - - -| + | - - -
   DEPTH = 10

PROC TABULATE (continued)

Examples of TABLE statements.

In the examples below, the required table format is specified with a TABLE statement and the output produced by different TABLE statements are sketched.

The TABLE statement used appears on top of the table produced.

PROC TABULATE (continued)
PROC TABULATE Examples

First we will consider only three variables: REGION, CITYSIZE, and POP where REGION and CITYSIZE are CLASS variables and POP is an analysis variable (numerical, continuous-valued). Each observation in the data contains data for a single city for many cities in several regions.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION</td>
<td>code for region of the country</td>
</tr>
<tr>
<td>CITYSIZE</td>
<td>code for relative population size (S=small, M=medium, L=large)</td>
</tr>
<tr>
<td>POP</td>
<td>urban population</td>
</tr>
</tbody>
</table>

Most applications like this need both CLASS and VAR statements in the PROC TABULATE step in addition to the TABLE statement:

```plaintext
PROC TABULATE;
TITLE " Population by Region";
CLASS REGION ;
VAR POP ;
```

Example 1: TABLE REGION,POP ;

Example 2: TABLE REGION,CITYSIZE*POP *SUM ;

Example 3: TABLE REGION*CITYSIZE,POP*SUM ;

Example 4: TABLE REGION*CITYSIZE*(SUM MEAN),POP ;
Next, add four additional variables to the example:

PRODUCT and SALETYPE are class variables:
QUANTITY and AMOUNT are analysis variables.

Modified proc step:

PROC TABULATE;
   TITLE “Sales by Region and City Size”;
   CLASS REGION PRODUCT SALETYPE;
   VAR POP QUANTITY AMOUNT;

The report produced by the following TABLE statement contains a separate table for each PRODUCT and provides an analysis of wholesale and retail sales by REGION and by CITYSIZE.

Example 5:

TABLE PRODUCT, REGION CITYSIZE, SALETYPE*(QUANTITY AMOUNT);

PRODUCT A100

<table>
<thead>
<tr>
<th>SALETYPE</th>
<th>R</th>
<th>W</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUANTITY</td>
<td>AMOUNT</td>
<td>QUANTITY</td>
<td>AMOUNT</td>
</tr>
<tr>
<td>REGION</td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
</tr>
<tr>
<td>NC</td>
<td>1250</td>
<td>31250</td>
<td>1250</td>
<td>25000</td>
</tr>
<tr>
<td>NE</td>
<td>1600</td>
<td>40000</td>
<td>1600</td>
<td>32000</td>
</tr>
<tr>
<td>SO</td>
<td>1880</td>
<td>47000</td>
<td>1880</td>
<td>37600</td>
</tr>
<tr>
<td>WE</td>
<td>1840</td>
<td>46000</td>
<td>1840</td>
<td>36800</td>
</tr>
<tr>
<td>CITYSIZE</td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
</tr>
<tr>
<td>L</td>
<td>3190</td>
<td>79750</td>
<td>3190</td>
<td>63800</td>
</tr>
<tr>
<td>M</td>
<td>2440</td>
<td>61000</td>
<td>2440</td>
<td>48800</td>
</tr>
<tr>
<td>S</td>
<td>940</td>
<td>23500</td>
<td>940</td>
<td>18800</td>
</tr>
</tbody>
</table>

PRODUCT A200

<table>
<thead>
<tr>
<th>SALETYPE</th>
<th>R</th>
<th>W</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QUANTITY</td>
<td>AMOUNT</td>
<td>QUANTITY</td>
<td>AMOUNT</td>
</tr>
<tr>
<td>REGION</td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
</tr>
<tr>
<td>NC</td>
<td>1295</td>
<td>32375</td>
<td>1295</td>
<td>25900</td>
</tr>
<tr>
<td></td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
<td>SUM</td>
</tr>
</tbody>
</table>