Chapter 14
Using the Output Delivery System

Chapter Table of Contents

Chapter 14
Using the Output Delivery System

Overview

In the latest version of SAS software, all SAS/STAT procedures use the Output Delivery System (ODS) to manage their output. This includes managing the form in which the output appears as well as its organization and format. The default for SAS/STAT procedures is to produce the usual SAS listing file. However, by using the features of the Output Delivery System, you can make changes to the format and appearance of your SAS output. In particular, you can

- display your output in hypertext markup language (HTML)
- display your output in Rich-Text-Format (RTF)*
- create SAS data sets directly from output tables
- select or exclude individual output tables
- customize the layout, format, and headers of your output

ODS features can provide you with a powerful tool for managing your output. This chapter provides background material and illustrates typical applications of ODS with SAS/STAT software.

For complete documentation on the Output Delivery System, refer to *The Complete Guide to the SAS Output Delivery System*.

Output Objects and ODS Destinations

All SAS procedures produce *output objects* that the Output Delivery System delivers to various *ODS destinations*, according to the default specifications for the procedure or to your own specifications.

All output objects (for example, a table of parameter estimates) consist of two component parts:

- the data component, which consists of the results computed by a SAS procedure
- the template, which contains rules for formatting and displaying the results

When you invoke a SAS procedure, the procedure sends all output to the Output Delivery System. ODS then routes the output to all open destinations. You define

*experimental in Version 7

the form the output should take when you specify an ODS destination. Supported destinations are as follows:

- Listing destination (the standard SAS listing), which is the default
- HTML destination, hypertext markup language
- Output destination, SAS data set

Future versions of ODS will support the following additional destinations:

- the ODS Output Document for modifying and replaying output without rerunning the procedure that created it
- Rich Text Format (RTF) for inclusion in Microsoft Word
- postscript and PCL for high fidelity printers

You can activate multiple ODS destinations at the same time, so that a single procedure step can route output to multiple destinations. If you do not supply any ODS statements, ODS delivers all output to the SAS listing, which is the default.

Each output object has an associated template that defines its presentation format. You can modify the presentation of the output by using the TEMPLATE procedure to alter these templates or to create new templates. You can also specify stylistic elements for ODS destinations, such as cell formats and headers, column ordering, colors, and fonts. For detailed information, refer to the chapter titled "The Template Procedure" in the *SAS Procedures Guide*.

Using the Output Delivery System

The ODS statement is a global statement that enables you to provide instructions to the Output Delivery System. You can use ODS statements to specify options for different ODS destinations, select templates to format your output, and select and exclude output. You can also display the names of individual output tables as they are generated.

In order to select, exclude, or modify a table, you must first know its name. You can obtain the table names in several ways:

- For any SAS/STAT procedure, you can obtain table names from the individual procedure chapter or from the individual procedure section of the SAS online Help system.
- For any SAS procedure, you can use the SAS Explorer window to view the names of the tables created in your SAS run (see the section "Using ODS with the SAS Explorer" on page 7 for more information).
- For any SAS procedure, you can use the ODS TRACE statement to find the names of tables created in your SAS run. The ODS TRACE statement writes identifying information to the SAS log (or, optionally, to the SAS listing) for each generated output table.

Specify the ODS TRACE ON statement prior to the procedure statements that create the output for which you want information. For example, the following statements write the trace record for the specific tables created in this REG procedure step.

```
ods trace on;
proc reg;
   model y=x;
   model z=x;
run;
```

By default, the trace record is written to the SAS log, as displayed in Figure 14.1. Alternatively, you can specify the LISTING option, which writes the information, interleaved with the procedure output, to the SAS listing (see Example 14.3).

```
   ods trace on;
   proc reg;
      model y=x;
      model z=x;
   run;

    .
    .
    .

Output Added:
-------------
Name:        ParameterEstimates
Label:       Parameter Estimates
Template:    Stat.REG.ParameterEstimates
Path:        Reg.MODEL1.Fit.y.ParameterEstimates
-------------
    .
    .
    .

Output Added:
-------------
Name:        ParameterEstimates
Label:       Parameter Estimates
Template:    Stat.REG.ParameterEstimates
Path:        Reg.MODEL2.Fit.z.ParameterEstimates
-------------
```

Figure 14.1. Partial Contents of the SAS Log: Result of the ODS TRACE Statement

Figure 14.1 displays the trace record, which contains the name of each created table and its associated label, template, and path. The label provides a description of the table. The template name displays the name of the template used to format the table. The path shows the output hierarchy to which the table belongs.

The fully qualified path is given in the trace record. A partially qualified path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For example, the full path for the parameter estimates for the first model in the preceding regression analysis is

```
Reg.Model1.Fit.y.ParameterEstimates
```

Therefore, partially qualified paths for the table are

```
Model1.fit.y.ParameterEstimates
fit.y.ParameterEstimates
y.ParameterEstimates
```

To refer to a table (in order to select or exclude it from display, for example), specify either the table name or use the table's fully or partially qualified path. You may want to use qualified paths when your SAS program creates several tables that have the same name, as in the preceding example. In such a case, you can use a partially qualified path to select a subset of tables, or you can use a fully qualified path to select a particular table.

You specify the tables that ODS selects or excludes with the ODS SELECT or ODS EXCLUDE statement. Suppose that you want to display only the tables of parameter estimates from the preceding regression analysis. You can give any of the following statements (before invoking the REG procedure) to display both tables of parameter estimates. For this example, these statements are equivalent:

```
ods select Reg.Model1.Fit.y.ParameterEstimates
           Reg.Model1.Fit.z.ParameterEstimates;

ods select y.ParameterEstimates z.ParameterEstimates;

ods select ParameterEstimates;
```

The first ODS SELECT statement specifies the full path for both tables. The second statement specifies the partially qualified path for both tables. The third statement specifies the single name "ParameterEstimates," which is shared by both tables.

The Output Delivery System records the specified table names in its internal selection or exclusion list. ODS then processes the output it receives. Note that ODS maintains an overall selection or exclusion list that pertains to all ODS destinations, and it maintains a separate selection or exclusion list for each ODS destination. The list for a specific destination provides the primary filtering step. Restrictions you specify in the overall list are added to the destination-specific lists.

Suppose, for example, that your listing exclusion list (that is, the list of tables you wish to exclude from the SAS listing) contains the "FitStatistics" table, which you specify with the statement

```
ods listing exclude FitStatistics;
```

and your overall selection list (that is, the list of tables you want to select for all destinations) contains the tables "FitStatistics" and "ParameterEstimates," which you specify with the statement

```
ods select ParameterEstimates FitStatistics;
```

The Output Delivery System then sends only the "ParameterEstimates" and "FitStatistics" tables to all open destinations except the SAS listing. It sends only the "ParameterEstimates" table to the SAS listing because the table "FitStatistics" is excluded from that destination.

Some SAS procedures, such as the REG or the GLM procedure, support run-group processing, which means that a RUN statement does not end the procedure. A QUIT statement explicitly ends such procedures; if you omit the QUIT statement, a PROC or a DATA statement implicitly ends such procedures. When you use the Output Delivery System with procedures that support run-group processing, it is good programming practice to specify a QUIT statement at the end of the procedure. This causes ODS to clear the selection or exclusion list, and you are less likely to encounter unexpected results.

Using ODS with the SAS Explorer

The SAS Explorer is a new feature that enables you to examine the various parts of the SAS System. Figure 14.2 displays the Results window from the SAS Explorer. The Results node retains a running record of your output as it is generated during your SAS session. Figure 14.2 displays the output hierarchy when the preceding statements are executed.

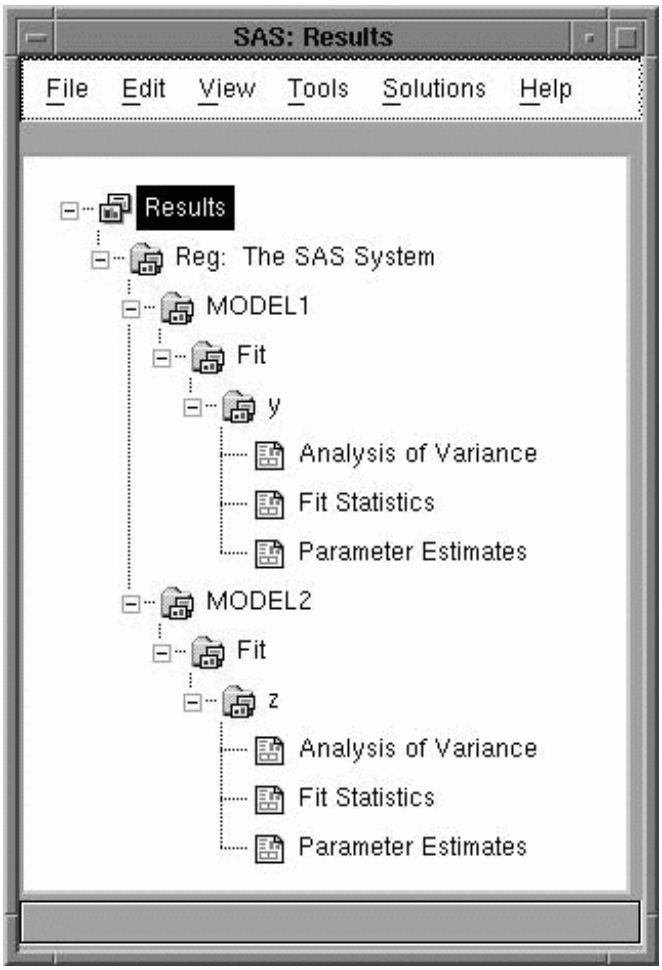


Figure 14.2. The Results Window from the SAS Explorer

When you click on the output table names in the Results window, you link directly to the output in the output window or, if you specify the HTML destination, in an HTML browser. The items on the left-hand side of the Results node are output directories. The items on the right-hand side of the Results node are the names of the actual output objects. You can also use the Explorer to determine names of the templates associated with each output table.

Controlling Output Appearance with Templates

A template is an abstract description of how output should appear when it is formatted. Templates describe several characteristics of the output, including headers, column ordering, style information, justification, and formats. All SAS/STAT procedures have templates, which are stored in the SASHELP library.

You can create or modify a template with the TEMPLATE procedure. For example, you can specify different column headings or different orderings of columns in a table. You can find the template associated with a particular output table by using the ODS TRACE statement or the SAS Explorer.

You can display the contents of a template by executing the following statements:

```
proc template;
   source  templatename;
run;
```

where *templatename* is the name of the template.

Suppose you want to change the way all of the analysis of variance tests are displayed by the GLM procedure. You can redefine the templates that the procedure uses with PROC TEMPLATE. For example, in order to have the SS and MS columns always displayed with more digits, you can redefine the columns used by the procedure to display them:

```
proc template;
   edit Stat.GLM.SS;
      format=Best16.;
   end;
   edit Stat.GLM.MS;
      format=Best16.;
   end;
run;
```

The BEST*w*. format enables you to display the most information about a value, according on the available field width. The BEST16. format specifies a field width of 16. Refer to the chapter on formats in *SAS Language Reference: Dictionary* for detailed information.

When you run PROC TEMPLATE to modify or edit a template, the template is stored in your SASUSER library (see Example 14.11). You can then modify the path that ODS uses to look up templates with the ODS PATH statement in order to access these new templates in a later SAS session. This means that you can create a default set of templates to modify the presentation format for all your SAS output. (Note that you can specify the SHOW option in the ODS PATH statement to determine the current path.)

It is important to note the difference between a style template and a table template. A table template applies only to the specific tables that reference the template. The preceding statements that modify the “Stat.GLM.SS” and “Stat.GLM.MS” templates provide an example of modifying columns within a table template.

A style template applies to an entire SAS job and can be specified only in the ODS HTML statement. You can specify a style as follows:

```
ods html style=Styles.Brown;
```

A style template controls stylistic elements such as colors, fonts, and presentation attributes. When you use a style template, you ensure that all your output shares a consistent presentation style.

You can also reference style information in table templates for individual headers and data cells. You can modify either type of template with the TEMPLATE procedure.

For information on creating your own styles, refer to *The Complete Guide to the SAS Output Delivery System*.

Interaction Between ODS and the NOPRINT Option

Most SAS/STAT procedures support a NOPRINT option that you can use when you want to create an output data set but do not want any displayed output. Typically, you use an OUTPUT statement in addition to the procedure's NOPRINT option to create a data set and suppress displayed output.

You can also use the Output Delivery System to create output data sets by using the ODS OUTPUT statement. However, if you specify the NOPRINT option, the procedure may not send any output to the Output Delivery System. Therefore, when you want to create output data sets through ODS (using the ODS OUTPUT statement), and you want to suppress the display of all output, specify

```
ODS SELECT NONE;
```

or close the active ODS destinations by giving the command

```
ODS  destinationname CLOSE;
```

where *destinationname* is the name of the active ODS destination (for example, ODS HTML CLOSE).

Note: The ODS statement does not instruct a procedure to generate output: instead, it specifies how the Output Delivery System should manage the table once it is created. You must ensure that the proper options are in effect. For example, the following code does not create the requested data set Parms.

```
proc glm;
   ods output ParameterEstimates=Parms;
   class x;
   model y=x;
   run;
quit;
```

When you execute these statements, the following line is displayed in the log:

```
WARNING: Output 'ParameterEstimates' was not created.
```

The data set Parms is not created because the table of parameter estimates is generated only when the SOLUTION option is specified in the MODEL statement in the GLM procedure.

Compatibility Issues with Version 6 Prototypes

- In Version 6, the MIXED and GENMOD procedures use a prototype of the Output Delivery System. This prototype provides the MAKE statement in order to create data sets from output tables, and this statement remains supported

in these procedures. However, the new mechanism to create SAS data sets from output tables is the ODS OUTPUT statement for all procedures.

- The Version 6 prototype of the ODS output hierarchy is stored in a SAS catalog. The latest version of SAS software has a more flexible item-store file type used to store templates and ODS output.
- The Version 6 prototype ODS uses two macro variables (_DISK_ and _PRINT_) to regulate the saving of an output hierarchy. The latest version of SAS software uses the global ODS statement to accomplish this task.
- The Version 6 PROC TEMPLATE and PROC OUTPUT syntax is not compatible with the latest version of SAS software.

Examples

The following examples display typical uses of the Output Delivery System.

Example 14.1. Creating HTML Output with ODS

This example demonstrates how you can use the ODS HTML statement to display your output in hypertext markup language (HTML).

The following statements create the data set scores, which contains the golf scores for boys and girls in a physical education class. The TTEST procedure is then invoked to compare the scores.

The ODS HTML statement specifies the name of the file to contain body of the HTML output.

```
data scores;
   input Gender $ Score @@;
   datalines;
f 75  f 76  f 80  f 77  f 80  f 77  f 73
m 82  m 80  m 85  m 85  m 78  m 87  m 82
;
run;

ods html body='ttest.htm';

title 'Comparing Group Means';
proc ttest ;
   class Gender;
   var Score;
run;
ods html close;
```

By default, the SAS listing receives all output generated during your SAS run. In this example, the ODS HTML statement opens the HTML destination, and both destinations receive the generated output. Output 14.1.1 displays the results as they are rendered in the SAS listing.

Note that you must specify the following statement before you can view your output in a browser.

```
ods html close;
```

If you do not close the HTML destination, your HTML file may contain no output, or you may experience other unexpected results.

Output 14.1.2 displays the file 'ttest.htm', which is specified in the preceding ODS HTML statement.

Output 14.1.1. Results for PROC TTEST: SAS Listing Procedure Output

Comparing Group Means

The TTEST Procedure

Statistics

Variable	Class	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
Score	f	7	74.504	76.857	79.211	1.6399	2.5448	5.6039	0.9619	73	80
Score	m	7	79.804	82.714	85.625	2.028	3.1472	6.9303	1.1895	78	87
Score	Diff (1-2)		-9.19	-5.857	-2.524	2.0522	2.8619	4.7242	1.5298		

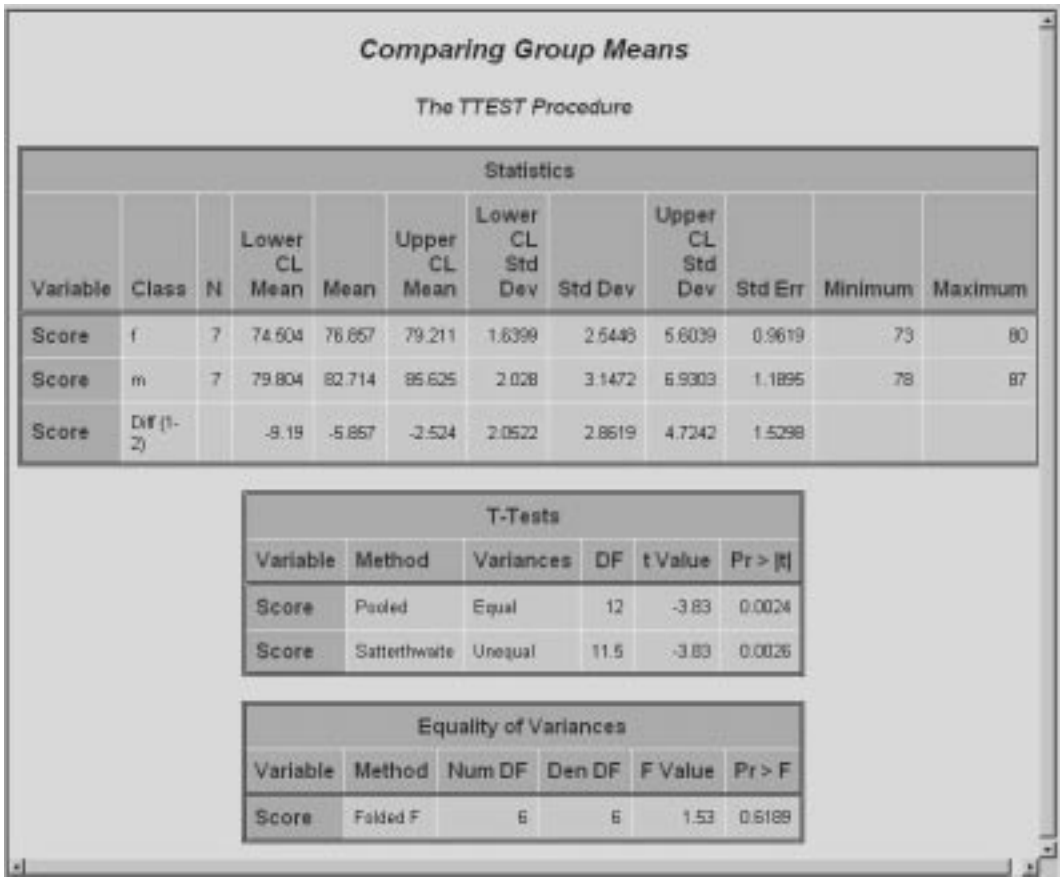
T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
Score	Pooled	Equal	12	-3.83	0.0024
Score	Satterthwaite	Unequal	11.5	-3.83	0.0026

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
Score	Folded F	6	6	1.53	0.6189

Output 14.1.2. Results for PROC TTEST: HTML Procedure Output



Comparing Group Means

The TTEST Procedure

Statistics

Variable	Class	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
Score	f	7	74.504	76.857	79.211	1.6399	2.5448	5.6039	0.9619	73	80
Score	m	7	79.804	82.714	85.625	2.028	3.1472	6.9303	1.1895	78	87
Score	Diff (1-2)		-9.19	-5.857	-2.524	2.0522	2.8619	4.7242	1.5298		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
Score	Pooled	Equal	12	-3.83	0.0024
Score	Satterthwaite	Unequal	11.5	-3.83	0.0026

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
Score	Folded F	6	6	1.53	0.6189

Example 14.2. Creating HTML Output with a Table of Contents

The following example uses ODS to display the output in HTML with a table of contents.

The data are from Pothoff and Roy (1964) and consist of growth measurements for 11 girls and 16 boys at ages 8, 10, 12, and 14.

```
data pr;
   input Person Gender $ y1 y2 y3 y4;
   y=y1; Age=8;  output;
   y=y2; Age=10; output;
   y=y3; Age=12; output;
   y=y4; Age=14; output;
   drop y1-y4;
   datalines;
 1   F   21.0    20.0    21.5    23.0
 2   F   21.0    21.5    24.0    25.5
 3   F   20.5    24.0    24.5    26.0
 4   F   23.5    24.5    25.0    26.5
 5   F   21.5    23.0    22.5    23.5
 6   F   20.0    21.0    21.0    22.5
 7   F   21.5    22.5    23.0    25.0
 8   F   23.0    23.0    23.5    24.0
 9   F   20.0    21.0    22.0    21.5
10   F   16.5    19.0    19.0    19.5
11   F   24.5    25.0    28.0    28.0
12   M   26.0    25.0    29.0    31.0
13   M   21.5    22.5    23.0    26.5
14   M   23.0    22.5    24.0    27.5
15   M   25.5    27.5    26.5    27.0
16   M   20.0    23.5    22.5    26.0
17   M   24.5    25.5    27.0    28.5
18   M   22.0    22.0    24.5    26.5
19   M   24.0    21.5    24.5    25.5
20   M   23.0    20.5    31.0    26.0
21   M   27.5    28.0    31.0    31.5
22   M   23.0    23.0    23.5    25.0
23   M   21.5    23.5    24.0    28.0
24   M   17.0    24.5    26.0    29.5
25   M   22.5    25.5    25.5    26.0
26   M   23.0    24.5    26.0    30.0
27   M   22.0    21.5    23.5    25.0
run;

ods html body='mixed.htm'
         contents='mixedc.htm'
         frame='mixedf.htm';

proc mixed data=pr method=ml covtest asycov;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
```

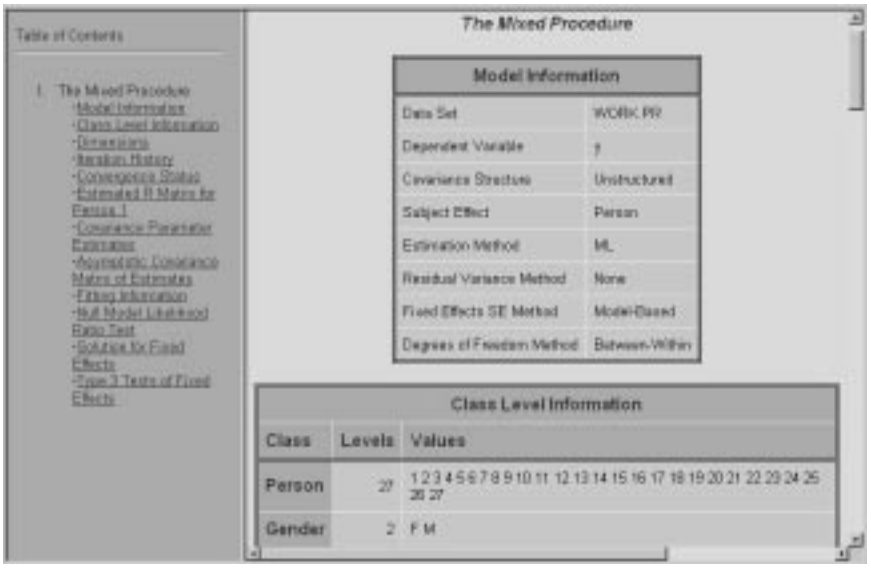
```
   repeated / type=un subject=Person r;
run;
ods html close;
```

The ODS HTML statement specifies three files. The BODY= argument specifies the file to contain the output generated from the statements that follow. The BODY= argument is required.

The CONTENTS= option specifies a file to contain the table of contents. The FRAME= option specifies a file to contain both the table of contents and the output. You open the FRAME= file in your browser to view the table of contents together with the generated output (see Output 14.2.1). Note that, if you specify the ODS HTML statement with only the BODY= argument, no table of contents is created.

The MIXED procedure is invoked to fit the specified model. The resulting output is displayed in Output 14.2.1.

Output 14.2.1. HTML Output from the MIXED Procedure



The table of contents displayed in Output 14.2.1 contains the descriptive label for each output table produced in the MIXED procedure step. You can select any label in the table of contents and the corresponding output will be displayed in the right-hand side of the browser window.

Example 14.3. Determining the Names of ODS Tables

In order to select or exclude a table, or to render it as a SAS data set, you must first know its name. You can obtain the table names in several ways:

- For any SAS/STAT procedure, you can obtain table names from the individual procedure chapter or from the SAS online Help system.
- For any SAS procedure, you can use the SAS Explorer window to view the names of the tables created in your SAS run.
- For any SAS procedure, you can use the ODS TRACE statement to find the names of tables created in your SAS run. The ODS TRACE statement writes identifying information to the SAS log for each generated output table.

This example uses the ODS TRACE statement with the LISTING option to obtain the names of the created output objects. By default, the ODS TRACE statement writes its information to the SAS log. However, you can specify the LISTING option to have the information interleaved with the procedure output in the SAS listing.

Suppose that you perform a randomized trial on rats that have been exposed to a carcinogen. You divide them into two groups and give each group a different treatment. In the following example, interest lies in whether the survival distributions differ between the two treatments. The data set Exposed contains four variables: Days (survival time in days from treatment to death), Status (censoring indicator variable: 0 if censored and 1 if not censored), Treatment (treatment indicator), and Sex (gender: F if female and M if male).

```
data Exposed;
  input Days Status Treatment Sex $ @@;
  datalines;
179   1   1   F   378   0   1   M
256   1   1   F   355   1   1   M
262   1   1   M   319   1   1   M
256   1   1   F   256   1   1   M
255   1   1   M   171   1   1   F
224   0   1   F   325   1   1   M
225   1   1   F   325   1   1   M
287   1   1   M   217   1   1   F
319   1   1   M   255   1   1   F
264   1   1   M   256   1   1   F
237   0   2   F   291   1   2   M
156   1   2   F   323   1   2   M
270   1   2   M   253   1   2   M
257   1   2   M   206   1   2   F
242   1   2   M   206   1   2   F
157   1   2   F   237   1   2   M
249   1   2   M   211   1   2   F
180   1   2   F   229   1   2   F
226   1   2   F   234   1   2   F
268   0   2   M   209   1   2   F
;
```

```
ods trace on/listing;

proc lifetest data=Exposed;
   time Days*Status(0);
   strata Treatment;
run;

ods trace off;
```

The purpose of these statements is to obtain the names of the ODS tables produced in this PROC LIFETEST run. The ODS TRACE ON statement writes the trace record of ODS output tables. The LISTING option specifies that the information is interleaved with the output and written to the SAS listing.

The LIFETEST procedure is invoked to perform the analysis, the SAS listing receives the procedure output and the trace record, and the trace is then disabled with the OFF option.

Output 14.3.1. The ODS Trace, Interleaved with LIFETEST Results: Partial Results

The LIFETEST Procedure

Output Added:

Name: ProductLimitEstimates
Label: Product-Limit Estimates
Template: Stat.Lifetest.ProductLimitEstimates
Path: Lifetest.Stratum1.ProductLimitEstimates

Stratum 1: Treatment = 1

Product-Limit Survival Estimates

Days	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.000	1.0000	0	0	0	20
171.000	0.9500	0.0500	0.0487	1	19
179.000	0.9000	0.1000	0.0671	2	18
217.000	0.8500	0.1500	0.0798	3	17
224.000*	.	.	.	3	16
225.000	0.7969	0.2031	0.0908	4	15
255.000	.	.	.	5	14
255.000	0.6906	0.3094	0.1053	6	13
256.000	.	.	.	7	12
256.000	.	.	.	8	11
256.000	.	.	.	9	10
256.000	0.4781	0.5219	0.1146	10	9
262.000	0.4250	0.5750	0.1135	11	8
264.000	0.3719	0.6281	0.1111	12	7
287.000	0.3188	0.6813	0.1071	13	6
319.000	.	.	.	14	5
319.000	0.2125	0.7875	0.0942	15	4
325.000	.	.	.	16	3
325.000	0.1063	0.8938	0.0710	17	2
355.000	0.0531	0.9469	0.0517	18	1
378.000*	.	.	.	18	0

NOTE: The marked survival times are censored observations.

Summary Statistics for Time Variable Days

Output Added:

Name: Quartiles
Label: Quartiles
Template: Stat.Lifetest.Quartiles
Path: Lifetest.Stratum1.TimeSummary.Quartiles

Quartile Estimates

Percent	Point Estimate	95% Confidence Interval [Lower	Upper)
75	319.000	262.000	325.000
50	256.000	255.000	319.000
25	255.000	217.000	256.000

As displayed in Output 14.3.1, the ODS TRACE ON statement writes the name, label, template, and path name of each generated ODS table. For more information

on names, labels, and qualified path names, see the discussion in the section "Using the Output Delivery System" beginning on page 4.

The information obtained with the ODS TRACE ON statement enables you to request output tables by name. The examples that follow demonstrate how you can use this information to select, exclude, or create data sets from particular output tables.

Example 14.4. Selecting ODS Tables for Display

You can use the ODS SELECT statement to deliver only certain tables to open ODS destinations. In the following example, the GLM procedure is used to perform an analysis on an unbalanced two-way experimental design.

```
data twoway;
title "Unbalanced Two-way Design";
   input Treatment Block y @@;
   datalines;
1 1 17   1 1 28   1 1 19   1 1 21   1 1 19
1 2 43   1 2 30   1 2 39   1 2 44   1 2 44
1 3 16
2 1 21   2 1 21   2 1 24   2 1 25
2 2 39   2 2 45   2 2 42   2 2 47
2 3 19   2 3 22   2 3 16
3 1 22   3 1 30   3 1 33   3 1 31
3 2 46
3 3 26   3 3 31   3 3 26   3 3 33   3 3 29   3 3 25
;
ods select ModelANOVA Means;
ods show;
ods trace on;
```

The ODS SELECT statement specifies that only the two tables “ModelANOVA” and “Means” are to be delivered to the ODS destinations. In this example, no ODS destinations are explicitly opened. Therefore, only the default SAS listing receives the procedure output. The ODS SHOW statement displays the current overall selection list in the SAS log. The ODS TRACE statement writes the trace record of the ODS output objects to the SAS log. In the following statements, the GLM procedure is invoked to produce the output.

```
proc glm data=twoway;
   class Treatment Block;
   model y = Treatment|Block;
   means Treatment;
   lsmeans Treatment;
run;
```

Output 14.4.1 displays the results of the ODS SHOW statement, which writes the current overall selection list to the SAS log. As specified in the preceding ODS SELECT statement, only the two ODS tables “ModelANOVA” and “Means” are selected for output.

Output 14.4.1. Results of the ODS SHOW Statement

```
             ods select ModelANOVA Means;
             ods trace on;
             ods show;

Current OVERALL select list is:
1. ModelANOVA
2. Means
```

Partial results of the ODS TRACE statement, which is written to the SAS log, is displayed in Output 14.4.2. Note that there are two tables having the name "ModelANOVA," which are the Type I Model Anova and the Type III Model Anova tables. Similarly, there are two ODS tables having the name "Means," which are the Means and the LS-means tables.

Output 14.4.2. The ODS TRACE: Partial Contents of the SAS Log

```
             proc glm data=twoway;
                class Treatment Block;
                model y = Treatment|Block;
                means Treatment;
                lsmeans Treatment;
         run;


Output Added:
-------------
Name:       ClassLevels
Label:      Class Levels
Template:   STAT.GLM.ClassLevels
Path:       GLM.Data.ClassLevels
-------------
               .
               .
               .
               .
Output Added:
-------------
Name:       ModelANOVA
Label:      Type I Model ANOVA
Template:   stat.GLM.Tests
Path:       GLM.ANOVA.y.ModelANOVA
-------------

Output Added:
-------------
Name:       ModelANOVA
Label:      Type III Model ANOVA
Template:   stat.GLM.Tests
Path:       GLM.ANOVA.y.ModelANOVA
-------------
NOTE: Means from the MEANS statement are not adjusted for other terms in the
      model.  For adjusted means, use the LSMEANS statement.

Output Added:
-------------
Name:       Means
Label:      Means
Template:   stat.GLM.Means
Path:       GLM.Means.Treatment.Means
-------------

Output Added:
-------------
Name:       Means
Label:      Means
Template:   stat.GLM.LSMeans
Path:       GLM.LSMEANS.Treatment.Means
-------------
```

In the following statements, the ODS SHOW statement writes the current overall selection list to the SAS log. The QUIT statement ends the GLM procedure. The second ODS SHOW statement writes the selection list to the log after PROC GLM terminates. The ODS selection list is reset to 'ALL,' by default, when a procedure terminates. For more information on ODS exclusion and selection lists, see the section "Using the Output Delivery System" beginning on page 4.

```
ods show;
quit;
```

```
ods show;
```

The results of the statements are displayed in Output 14.4.3. Before the GLM procedure terminates, the ODS selection list includes only the two tables, "ModelANOVA" and "Means."

Output 14.4.3. The ODS Selection List, Before and After PROC GLM Terminates

```
            ods show;

Current OVERALL select list is:
1. ModelANOVA
2. Means


              quit;

NOTE: The PROCEDURE GLM printed pages 11-13.


              ods show;

Current OVERALL select list is: ALL
```

The GLM procedure supports run-group processing. Before the QUIT statement is executed, PROC GLM is active and the ODS selection list remains at its previous setting before PROC GLM was invoked. After the QUIT statement, the selection list is reset to deliver all output tables.

The entire displayed output consists of the four selected tables (two "ModelANOVA" tables and two "Means" tables), as displayed in Output 14.4.4 and Output 14.4.5.

Output 14.4.4. The ModelANOVA Tables from PROC GLM

```
                                Unbalanced Two-way Design

                                   The GLM Procedure

Dependent Variable: y

        Source                          DF       Type I SS     Mean Square    F Value    Pr > F

        Treatment                        2        8.060606        4.030303       0.24    0.7888
        Block                            2     2621.864124     1310.932062      77.95    <.0001
        Treatment*Block                  4       32.684361        8.171090       0.49    0.7460


        Source                          DF      Type III SS     Mean Square    F Value    Pr > F

        Treatment                        2      266.130682      133.065341       7.91    0.0023
        Block                            2     1883.729465      941.864732      56.00    <.0001
        Treatment*Block                  4       32.684361        8.171090       0.49    0.7460
```

Output 14.4.5. The Means Tables from PROC GLM

```
                    Unbalanced Two-way Design

                        The GLM Procedure

Level of                        ----------y----------
Treatment          N                Mean          Std Dev

1                 11          29.0909091       11.5104695
2                 11          29.1818182       11.5569735
3                 11          30.1818182        6.3058414



                    Unbalanced Two-way Design

                        The GLM Procedure
                       Least Squares Means

                    Treatment           y LSMEAN

                    1                25.6000000
                    2                28.3333333
                    3                34.4444444
```

Example 14.5. Excluding ODS Tables from Display

The following example demonstrates how you can use the ODS EXCLUDE statement to exclude particular tables from ODS destinations. This example also creates a SAS data set from the excluded table.

The data are from Hemmerle and Hartley (1973). The response variable consists of measurements from an oven experiment, and the model contains a fixed effect A and random effects B and A*B.

```
data hh;
   input a b y @@;
   datalines;
1 1 237   1 1 254    1 1 246
1 2 178   1 2 179
2 1 208   2 1 178    2 1 187
2 2 146   2 2 145    2 2 141
3 1 186   3 1 183
3 2 142   3 2 125    3 2 136
;
ods html body='mixed.htm'
         contents='mixedc.htm'
         frame='mixedf.htm';

ods exclude ParmSearch(persist);
ods show;
```

The ODS HTML statement specifies the filenames to contain the output generated from the statements that follow.

The ODS EXCLUDE statement excludes the table "ParmSearch" from display. Although the table is excluded from the displayed output, the information contained in the "ParmSearch" table is graphically summarized in a later step.

The PERSIST option in the ODS EXCLUDE statement excludes the table for the entire SAS session or until you execute an ODS SELECT statement or an ODS EXCLUDE NONE statement. If you omit the PERSIST option, the exclusion list is cleared when the procedure terminates.

The resulting exclusion list is displayed in Output 14.5.1.

Output 14.5.1. Results of the ODS SHOW Statement, Before PROC MIXED

```
          ods exclude ParmSearch(persist);
          ods show;

Current OVERALL exclude list is:
1. ParmSearch(PERSIST)
```

The following ODS OUTPUT statement outputs the table "ParmSearch" to a SAS data set called parms. The MIXED procedure is invoked and the model is fit. All

output from the MIXED procedure, except the "ParmSearch" table, is delivered to the HTML destination and the SAS listing. The ODS SHOW statement again displays the overall current exclusion list.

```
ods output ParmSearch=parms;
proc mixed data=hh asycov mmeq mmeqsol covtest;
   class a b;
   model y = a / outp=predicted;
   random b a*b;
   lsmeans a;
   parms (17 to 20 by 0.1) (.3 to .4 by .005) (1.0);
run;

ods show;
```

The results of the ODS SHOW statement, given after the MIXED procedure has terminated, are displayed in Output 14.5.2.

Output 14.5.2. Results of the ODS SHOW Statement, After PROC MIXED

```
          proc mixed data=hh asycov mmeq mmeqsol covtest;
             class a b;
             model y = a / outp=predicted;
             random b a*b;
             lsmeans a;
             parms (17 to 20 by .1) (.3 to .4 by .005) (1.0);
          run;
          ods show;

Current OVERALL exclude list is:
1. ParmSearch(PERSIST)
```

Normally the ODS exclusion list is cleared at the conclusion of a procedure (for more information on ODS exclusion and selection lists, see the section "Using the Output Delivery System" on page 4). However, the PERSIST option in the preceding ODS EXCLUDE statement specifies that the "ParmSearch" table should remain in the exclusion list until the list is explicitly cleared (that is, when the ODS EXCLUDE NONE statement or an ODS SELECT statement is encountered). Output 14.5.2 shows that the exclusion list remains in effect after PROC MIXED terminates.

The PERSIST option is useful when you want to exclude the same table in further analyses during your SAS session.

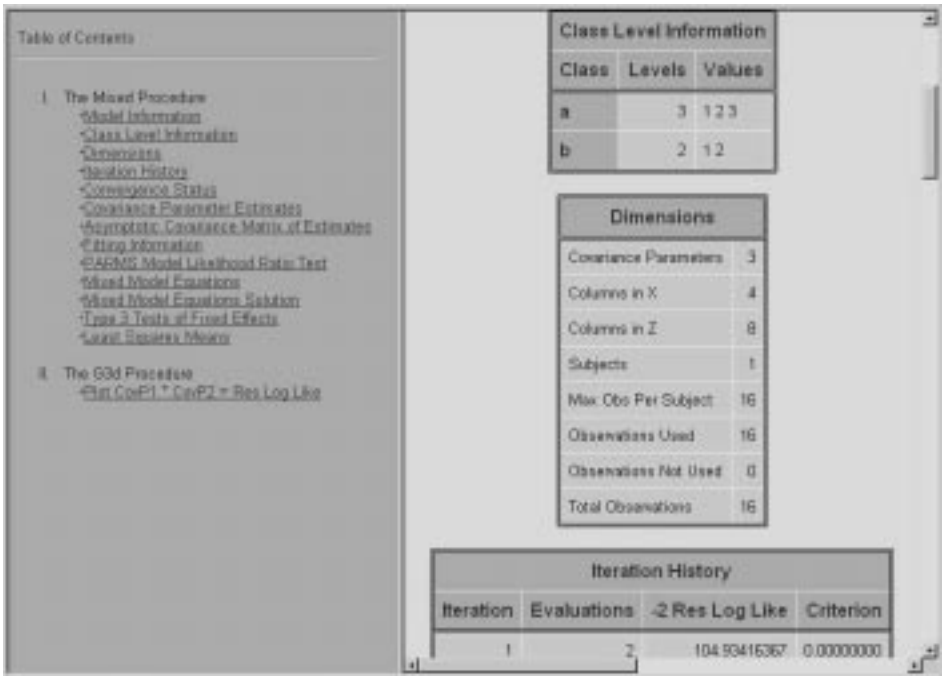
The "ParmSearch" table is contained in the parms data set (as specified in the ODS OUTPUT statement). The information is plotted with the G3D procedure in the following step.

```
proc g3d data=parms;
   plot CovP1*CovP2 = ResLogLike /
                ctop=red cbottom=blue caxis=black;
run;

ods html close;
```

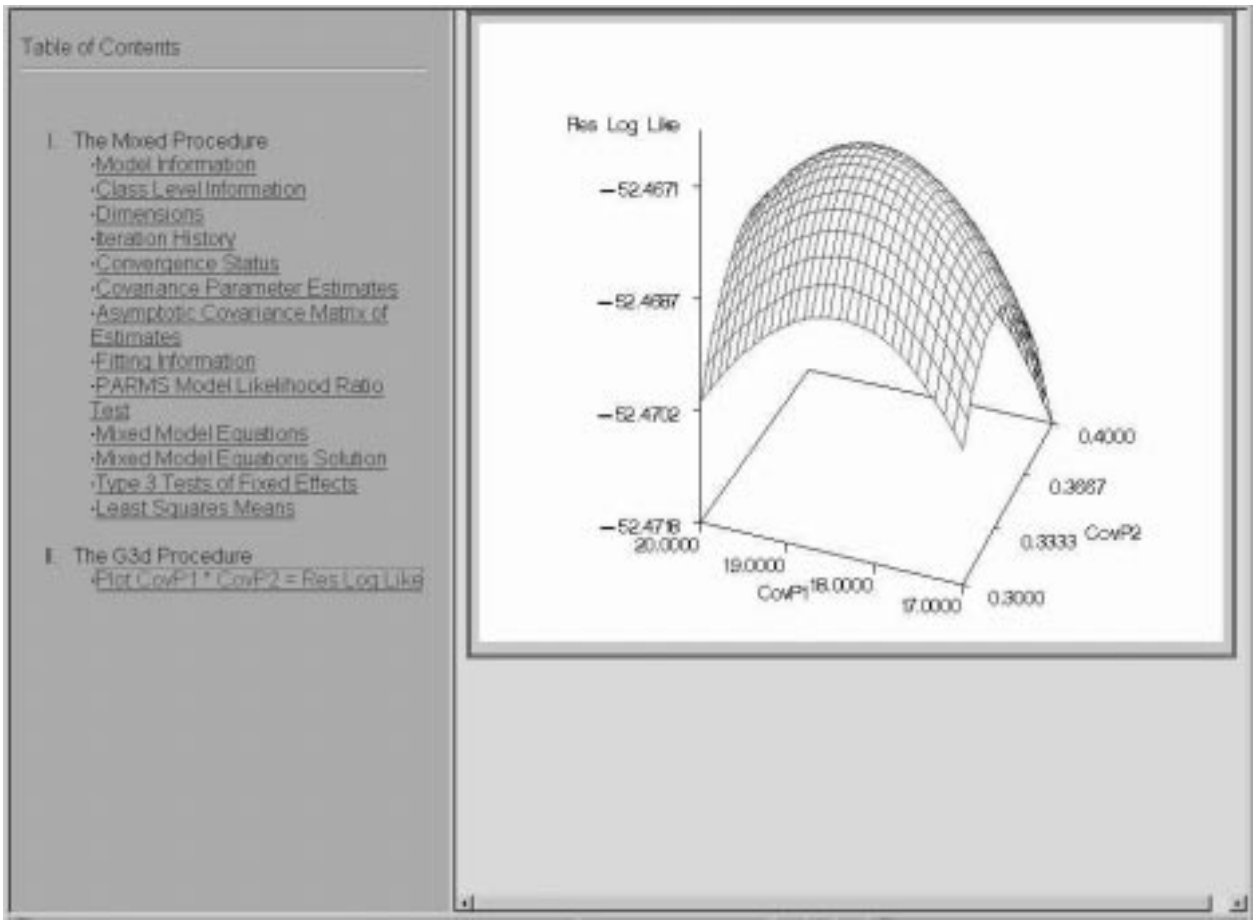
The MIXED procedure output resulting from the preceding statements is displayed in Output 14.5.3. The table of contents shows the names for all of the output tables. The "ParmSearch" table is not listed in the table of contents because of the preceding ODS EXCLUDE statement.

Output 14.5.3. HTML Output from the Mixed Procedure



The results of the G3D procedure is displayed in Output 14.5.4. The large amount of information contained in the table, which is excluded from display, can be summarized with a single plot.

Output 14.5.4. Plot of the ParmSearch Data Set



Example 14.6. Creating an Output Data Set from an ODS Table

The ODS OUTPUT statement creates SAS data sets from ODS tables. In the following example, the GENMOD procedure is invoked to perform Poisson regression and part of the resulting procedure output is output to a SAS data set.

Suppose the following insurance claims data are classified by two factors: age group (with two levels) and car type (with three levels).

```
data insure;
   input n c car$ age;
   ln = log(n);
   datalines;
500   42  small  1
1200  37  medium 1
100    1  large  1
400  101  small  2
500   73  medium 2
300   14  large  2
;
```

In the data set insure, the variable n represents the number of insurance policyholders and the variable c represents the number of insurance claims. The variable car represents the type of car involved (classified into three groups) and the variable age is the age group of a policyholder (classified into two groups).

In the statements that follow, PROC GENMOD performs a Poisson regression analysis of these data with a log link function. Assume that the number of claims c has a Poisson probability distribution and that its mean, μ_i, is related to the factors car and age.

Determining the Names of the ODS Tables

The purpose of the following statements is to obtain the names of the output tables produced in this PROC GENMOD run. The ODS TRACE statement lists the trace record, and the SAS listing destination is closed so that no output is displayed.

```
ods trace on;
ods listing close;

proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln
                       obstats;
run;
ods trace off;
```

Output 14.6.1. The ODS TRACE: Partial Contents of the SAS Log

```
          ods trace on;
          ods listing close;

          proc genmod data=insure;
             class car age;
             model c = car age / dist   = poisson
                                 link   = log
                                 offset = ln
                                 obstats;
          run;

Output Added:
-------------
Name:        ModelInfo
Label:       Model Information
Template:    Stat.Genmod.ModelInfo
Path:        Genmod.ModelInfo
-------------
             .
             .
             .
             .

Output Added:
-------------
Name:        ParameterEstimates
Label:       Analysis Of Parameter Estimates
Template:    stat.genmod.parameterestimates
Path:        Genmod.ParameterEstimates
-------------
NOTE: The scale parameter was held fixed.

Output Added:
-------------
Name:        ObStats
Label:       Observation Statistics
Template:    Stat.Genmod.Obstats
Path:        Genmod.ObStats
-------------
```

By default, the trace record is written to the SAS log, as displayed in Output 14.6.1. Note that you can alternatively specify that the information be interleaved with the procedure output in the SAS listing (see Example 14.3).

Creating the Output Data Set

In the statements that follow, the ODS OUTPUT statement writes the ODS table "ObStats" to a SAS data set called myObStats. All of the usual data set options, such as the KEEP= or RENAME= options, can be used in the ODS OUTPUT statement. Thus, to modify the ObStats data set so that it contains only certain variables, you can use the data set options as follows.

```
ods output ObStats=myObStats
             (keep=Car Age Pred
              rename=(Pred=PredictedValue));

proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
```

```
                        offset = ln
                        obstats;
run;
```

The KEEP= option in the ODS OUTPUT statement specifies that only the variables Car, Age, and Pred are written to the data set, and the Pred variable is renamed to PredictedValue. The GENMOD procedure is again invoked. In order to limit the amount of displayed output, the SAS listing destination remains closed. When a destination is closed, it remains closed until it is explicitly reopened.

In the following statements, the output data set myObStats is sorted, and the SAS listing is reopened for output. The results are displayed in Output 14.6.2.

```
proc sort data=myObStats;
   by descending PredictedValue;
run;

ods listing;
proc print data=myObStats noobs;
title 'Values of Car, Age, and the Predicted Values';
run;
```

Output 14.6.2. The ObStats Table, Created as a SAS Data Set

Values of Car, Age, and the Predicted Values

car	age	Predicted Value
small	2	107.2011
medium	2	67.025444
medium	1	42.974556
small	1	35.798902
large	2	13.773459
large	1	1.2265414

Example 14.7. Creating an Output Data Set: Subsetting the Data

This example demonstrates how you can create an output data set with the ODS OUTPUT statement and also use data set selection keywords to limit the output that ODS writes to a SAS data set.

The following data set, called Color, contains the eye and hair color of children from two different regions of Europe. The data are recorded as cell counts, where the variable Count contains the number of children exhibiting each of the 15 eye and hair color combinations.

```
data Color;
   input Region Eyes $ Hair $ Count @@;
      label Eyes  ='Eye Color'
            Hair  ='Hair Color'
            Region='Geographic Region';
datalines;
1 blue  fair   23  1 blue  red     7  1 blue  medium 24
1 blue  dark   11  1 green fair   19  1 green red     7
1 green medium 18  1 green dark   14  1 brown fair   34
1 brown red     5  1 brown medium 41  1 brown dark   40
1 brown black   3  2 blue  fair   46  2 blue  red    21
2 blue  medium 44  2 blue  dark   40  2 blue  black   6
2 green fair   50  2 green red    31  2 green medium 37
2 green dark   23  2 brown fair   56  2 brown red    42
2 brown medium 53  2 brown dark   54  2 brown black  13
;
```

In the statements that follow, the SAS listing is closed. The ODS OUTPUT statement creates the "ChiSq" table as a SAS data set called myStats. Note that you can obtain the names of the tables created by any SAS/STAT procedure in the individual procedure chapter or from the individual procedure section of the SAS online Help system. You can also determine the names of tables with the ODS TRACE statement (see Example 14.3 and Example 14.6).

The DROP= data set option excludes variables from the new data set. The WHERE= data set option selects particular observations for output to the new data set myStats.

```
ods listing close;

ods output ChiSq=myStats
           (drop=Table
            where=(Statistic='Chi-Square' or
                   Statistic='Likelihood Ratio Chi-Square'));
```

In the following statements, the Color data set is first sorted by the Region variable. The FREQ procedure is invoked to create and analyze a crosstabulation table from the two categorical variables Eyes and Hair, for each value of the variable Region.

No ODS destinations are open until the ODS LISTING statement is encountered just prior to the invocation of the PRINT procedure.

```
proc sort data=Color;
   by Region;
run;

proc freq data=Color order=data;
   weight Count;
   tables Eyes*Hair/testp=(30 12 30 25 3);
   by Region;
   title 'Hair Color of European Children';
run;

ods listing;
proc print data=myStats;
run;
```

Output 14.7.1 displays the output resulting from the previous statements.

Output 14.7.1. Output Data Set from PROC FREQ and ODS

Hair Color of European Children

Obs	Region	Statistic	DF	Value	Prob
1	1	Chi-Square	8	12.6331	0.1251
2	1	Likelihood Ratio Chi-Square	8	14.1503	0.0779
3	2	Chi-Square	8	18.2839	0.0192
4	2	Likelihood Ratio Chi-Square	8	23.3021	0.0030

Example 14.8. Concatenating Output Data Sets: BY-Group Processing

Your data may be structured in such a way that the columns of an output table change over BY groups or from one section of output to another. This can occur in analyses that contain CLASS variables with differing values for some BY groups. This example demonstrates how you can write multiple ODS tables (possibly with different columns) to a single data set by using the MATCH_ALL option in the ODS OUTPUT statement.

Suppose that you wish to analyze the following hypothetical data, which record the cell counts resulting from six types of blood tests administered at three laboratories. The values of method are not the same for each of the three laboratories. In the first laboratory, method = 1, 2, 3, 4; in the second laboratory, method = 2, 3, 4, 5; and in the third laboratory, method = 3, 4, 5, 6.

```
data Tests;
   input lab method count @@;
   datalines;
1 1 3  1 1 8  1 1 8  1 2 9  1 2 5  1 2 8
1 3 9  1 3 8  1 3 4  1 4 5  1 4 8  1 4 7
2 2 0  2 2 5  2 2 7  2 3 1  2 3 0  2 3 9
2 4 9  2 4 6  2 4 9  2 5 9  2 5 0  2 5 1
3 3 4  3 3 0  3 3 5  3 4 2  3 4 1  3 4 7
3 5 1  3 5 6  3 5 6  3 6 6  3 6 2  3 6 1
;
proc sort data=Tests;
   by lab;
run;
```

In the following analysis, the MATCH_ALL option is omitted and the $\mathbf{X}'\mathbf{X}$ matrix for each of the BY groups is output to a SAS data set. The columns of $\mathbf{X}'\mathbf{X}$ depend on the levels of the CLASS variable. Therefore, the structure of the $\mathbf{X}'\mathbf{X}$ matrix differs according to the level of the CLASS variable (method).

```
ods output XpX=my1XpX;    /* Incorrect for this example */

proc glm data=Tests ;
   class method;
   model count = method / xpx;
   by lab;
run;
quit;

proc print data=my1XpX;
title 'X''X Data Set, Omitting the MATCH_ALL Option';
run;
```

The GLM procedure produces the following warning:

```
WARNING: Output object 'XpX' contains 1 column(s) that cannot
         be mapped to data set WORK.MY1XPX (there is no
         corresponding variable on the output data set).
NOTE: The above message was for the following by-group:
   lab=2
```

The PRINT procedure results, displayed in Output 14.8.1, show that the data set is missing columns for method=5 and method=6, which are in the second and third BY groups (methods performed in laboratories 2 and 3).

Output 14.8.1. PRINT Procedure: Omitting the MATCH_ALL Option

X'X Data Set, Omitting the MATCH_ALL Option

Obs	lab	Parameter	Intercept	method_1	method_2	method_3	method_4	count
1	1	Intercept	12	3	3	3	3	82
2	1	method 1	3	3	0	0	0	19
3	1	method 2	3	0	3	0	0	22
4	1	method 3	3	0	0	3	0	21
5	1	method 4	3	0	0	0	3	20
6	1	count	82	19	22	21	20	606
7	2	Intercept	12	.	3	3	3	56
8	2	method 2	3	.	3	0	0	12
9	2	method 3	3	.	0	3	0	10
10	2	method 4	3	.	0	0	3	24
11	2	method 5	3	.	0	0	0	10
12	2	count	56	.	12	10	24	436
13	3	Intercept	12	.	.	3	3	41
14	3	method 3	3	.	.	3	0	9
15	3	method 4	3	.	.	0	3	10
16	3	method 5	3	.	.	0	0	13
17	3	method 6	3	.	.	0	0	9
18	3	count	41	.	.	9	10	209

When multiple tables with different columns have the same name, as in this case, you can use the MATCH_ALL option to obtain correct results. The MATCH_ALL option creates a separate data set for each table. The initial data set name is specified after the equal sign, outside the parentheses, as follows:

```
ods output XpX(match_all=list)=matrix;   /* Correct */

proc glm data=Tests;
   class method;
   model count = method / xpx;
   by lab;
run;
quit;
```

In the ODS OUTPUT statement, the specified data set name is matrix. When a second output table is generated, the corresponding data set name is created by appending a '1' to the specified data set name. In this case, the second data set name is matrix1. Subsequent data sets are named matrix2 and so on.

The macro variable list (specified as MATCH_ALL=list) contains the list of data set names. Thus, ODS creates the macro variable &list = matrix matrix1 matrix2. Note that the use of a macro variable name with MATCH_ALL is optional.

The set of values contained in the macro variable list is used in the following DATA step to combine all of the individual data sets into one data set. Note that, when referring to a macro variable, an ampersand (&) always precedes a macro variable name, but is not part of the name.

```
data my2XpX;
   set &list;
run;

proc print data=my2XpX;
title 'X''X Data Set, Specifying the MATCH_ALL Option';
run;
```

The final data set contains all of the results, including the results for method=5 and method=6 from the second and third laboratories. Output 14.8.2 displays the new data set.

Output 14.8.2. Print Procedure: Specifying the MATCH_ALL Option

X'X Data Set, Specifying the MATCH_ALL Option

Obs	Alter	Parameter	Intercept	method_1	method_2	method_3	method_4	Count	method_5	method_6
1	1	Intercept	12	3	3	3	3	82	.	.
2	1	method 1	3	3	0	0	0	19	.	.
3	1	method 2	3	0	3	0	0	22	.	.
4	1	method 3	3	0	0	3	0	21	.	.
5	1	method 4	3	0	0	0	3	20	.	.
6	1	count	82	19	22	21	20	606	.	.
7	2	Intercept	12	.	3	3	3	56	3	.
8	2	method 2	3	.	3	0	0	12	0	.
9	2	method 3	3	.	0	3	0	10	0	.
10	2	method 4	3	.	0	0	3	24	0	.
11	2	method 5	3	.	0	0	0	10	3	.
12	2	count	56	.	12	10	24	436	10	.
13	3	Intercept	12	.	.	3	3	41	3	3
14	3	method 3	3	.	.	3	0	9	0	0
15	3	method 4	3	.	.	0	3	10	0	0
16	3	method 5	3	.	.	0	0	13	3	0
17	3	method 6	3	.	.	0	0	9	0	3
18	3	count	41	.	.	9	10	209	13	9

Example 14.9. Concatenating Output Data Sets: RUN Group Processing

This example demonstrates how you can write multiple tables to a single data set using the MATCH_ALL= and PERSIST= options in the ODS OUTPUT statement. The PERSIST= option maintains ODS settings across RUN statements for procedures that support run-group processing. In the following analysis, the REG procedure is invoked and the covariance matrix of the estimates is output for two different models. The flexibility of the ODS OUTPUT statement enables you to create a single data set that contains both of the resulting covariance matrices.

Consider the following population growth trends. The population of the United States from 1790 to 1970 is fit to linear and quadratic functions of time. Note that the quadratic term, YearSq, is created in the DATA step; this is done since polynomial effects such as Year*Year cannot be specified in the MODEL statement in PROC REG. The data are as follows:

```
title1 'Concatenating Two Tables into One Data Set';
title2 'US Population Study';
data USPopulation;
   input Population @@;
   retain Year 1780;
   Year=Year+10;
   YearSq=Year*Year;
   Population=Population/1000;
   datalines;
3929 5308 7239 9638 12866 17069 23191 31443 39818 50155
62947 75994 91972 105710 122775 131669 151325 179323 203211
;
```

In the following statements, the REG procedure is invoked and the ODS statement requests that a data set be created to contain the COVB matrix (the covariance matrix of the estimates).

The ODS statement uses the MATCH_ALL= and PERSIST= options. The effect of these options is to create a separate data set for each COVB matrix encountered in the entire procedure step.

```
proc reg data=USPopulation;
   ods output covb(match_all=Bname          /* correct */
                   persist=run)=Bmatrix;
   var YearSq;
   model Population=Year / covb ;
run;
```

The MODEL statement defines the regression model, and the COVB matrix is requested. The RUN statement executes the REG procedure and the model is fit, producing a covariance matrix of the estimates with 2 rows and 2 columns.

Output 14.9.1. Regression Results for the Model Population=Year

Concatenating Two Output Tables into One Data Set
US Population Study

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	66336	66336	201.87	<.0001
Error	17	5586.29253	328.60544		
Corrected Total	18	71923			

Root MSE	18.12748	R-Square	0.9223
Dependent Mean	69.76747	Adj R-Sq	0.9178
Coeff Var	25.98271		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-1958.36630	142.80455	-13.71	<.0001
Year	1	1.07879	0.07593	14.21	<.0001

Output 14.9.2. CovB Matrix for the Model Population=Year

Concatenating Two Output Tables into One Data Set
US Population Study

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Covariance of Estimates

Variable	Intercept	Year
Intercept	20393.138485	-10.83821461
Year	-10.83821461	0.0057650078

In the next step, the YearSq variable is added to the model and the model is again fit, producing a covariance matrix of the estimates with three rows and three columns.

```
add YearSq;
print;
run;
```

The results of the regression are displayed in Output 14.9.3.

Output 14.9.3. Regression Results for the Model Population=Year YearSq

```
            Concatenating Two Output Tables into One Data Set
                          US Population Study

                            The REG Procedure
                             Model: MODEL1.1
                        Dependent Variable: Population

                            Analysis of Variance

                                    Sum of           Mean
Source                    DF       Squares         Square    F Value    Pr > F

Model                      2         71799          35900    4641.72    <.0001
Error                     16     123.74557        7.73410
Corrected Total           18         71923


              Root MSE              2.78102    R-Square     0.9983
              Dependent Mean       69.76747    Adj R-Sq     0.9981
              Coeff Var             3.98613


                            Parameter Estimates

                         Parameter       Standard
    Variable      DF      Estimate          Error    t Value    Pr > |t|

    Intercept      1         20450      843.47533      24.25      <.0001
    Year           1      -22.78061        0.89785     -25.37      <.0001
    YearSq         1        0.00635     0.00023877      26.58      <.0001
```

Output 14.9.4. CovB Matrix for the Model Population=Year YearSq

```
            Concatenating Two Output Tables into One Data Set
                          US Population Study

                            The REG Procedure
                             Model: MODEL1.1
                        Dependent Variable: Population

                          Covariance of Estimates

      Variable          Intercept              Year            YearSq

      Intercept      711450.62602      -757.2493826      0.2013282694
      Year           -757.2493826      0.8061328943      -0.000214361
      YearSq         0.2013282694      -0.000214361      5.7010894E-8
```

In the preceding analysis, two COVB matrices are generated, corresponding to the two fitted models. When you select two output tables that have the same name but different structures, specify the MATCH_ALL option to create a new data set for each table.

When you specify MATCH_ALL=*name*, a macro variable called *name* is created that contains the names of all the generated data sets. Thus, in this example, ODS creates two data sets (one for each model fit in the PROC REG run) and the macro variable Bname is created to contain the names of the two data sets.

The PERSIST=RUN option maintains the ODS selection list across RUN statements for procedures that support run-group processing. If the PERSIST=RUN option is

omitted, the selection list is cleared when the RUN statement is encountered and only the first COVB matrix is selected. Because the PERSIST=RUN option is specified, the selection list remains in effect throughout the REG procedure step. This ensures that each of the COVB matrices is selected and output.

The first output data set has the specified name BMatrix. Subsequent data set names are automatically created by appending the numerals $1, 2, 3, \ldots$, as needed. In this case the names of the data sets are BMatrix and BMatrix1. The names are contained in the macro variable Bname.

The result of the ODS OUTPUT statement is displayed with the following statements. The SET &BName statement reads observations from all data sets listed by the macro variable Bname. The variable Bname contains the two values (BMatrix and BMatrix1). Thus, the SET statement reads observations from these two data sets. Note that, when you refer to a macro variable, an ampersand (&) always precedes a macro variable name, but is not part of the name.

```
data new2;
   title 'The COVB Matrix Data Set, Using the PERSIST option';
   title2 'Concatenating Two Tables into One Data Set';
   set &Bname;
run;
proc print;
run;
```

The data set new2 contains the two data sets created from the two COVB matrices.

Output 14.9.5. Results of the ODS OUTPUT Statement, Specifying the PERSIST Option

The COVB Matrix Data Set, Using the PERSIST option
Concatenating Two Output Tables into One Data Set

Obs	_Run_	Model	Dependent	Variable	Intercept	Year	YearSq
1	1	MODEL1	Population	Intercept	20393.138485	-10.83821461	.
2	1	MODEL1	Population	Year	-10.83821461	0.0057650078	.
3	2	MODEL1.1	Population	Intercept	711450.62602	-757.2493826	0.2013282694
4	2	MODEL1.1	Population	Year	-757.2493826	0.8061328943	-0.000214361
5	2	MODEL1.1	Population	YearSq	0.2013282694	-0.000214361	5.7010894E-8

Example 14.10. Using the TEMPLATE Procedure to Customize Output

You can use the TEMPLATE procedure to modify the appearance of your displayed ODS tables. The following example, similar to that given in Olinger and Tobias (1998), creates output data sets using the ODS OUTPUT statement, modifies a template using PROC TEMPLATE, and displays the output data sets using the modified template.

The data set comes from a preclinical drug experiment (Cole and Grizzle 1966). In order to study the effect of two different drugs on histamine levels in the blood, the drugs are given to 13 animals and the levels of histamine in the animals' blood is measured after 0, 1, 3, and 5 minutes. The response variable is the logarithm of the histamine level. The following statements create a SAS data set named Histamine that contains the experimental data:

```
title1 "Histamine Study";
data Histamine;
   input Drug $12. Depleted $ hist0 hist1 hist3 hist5;
   logHist0 = log(hist0); logHist1 = log(Hist1);
   logHist3 = log(hist3); logHist5 = log(Hist5);
   datalines;
Morphine      N  .04  .20  .10  .08
Morphine      N  .02  .06  .02  .02
Morphine      N  .07 1.40  .48  .24
Morphine      N  .17  .57  .35  .24
Morphine      Y  .10  .09  .13  .14
Morphine      Y  .07  .07  .06  .07
Morphine      Y  .05  .07  .06  .07
Trimethaphan  N  .03  .62  .31  .22
Trimethaphan  N  .03 1.05  .73  .60
Trimethaphan  N  .07  .83 1.07  .80
Trimethaphan  N  .09 3.13 2.06 1.23
Trimethaphan  Y  .10  .09  .09  .08
Trimethaphan  Y  .08  .09  .09  .10
Trimethaphan  Y  .13  .10  .12  .12
Trimethaphan  Y  .06  .05  .05  .05
;
```

In the analysis that follows, the GLM procedure is invoked to perform a repeated measures analysis, naming the drug and depletion status as between-subject factors in the MODEL statement, and naming post-administration measurement time as the within-subject factor (for more information on this study and its analysis, see Example 7 in Chapter 28, "The GLM Procedure").

The following ODS statement requests that two ODS tables be written to SAS data sets called HistWithin and HistBetween. The SAS listing is closed so that no output is displayed. The GLM procedure is invoked and the model is fit.

```
ods output Mtests                      = HistWithin
           BetweenSubjects.ModelANOVA = HistBetween;
```

```
ods listing close;
proc glm data=Histamine;
   class Drug Depleted;

   model LogHist0--LogHist5 = Drug Depleted Drug*Depleted / nouni;
   repeated Time 4 (0 1 3 5) polynomial / summary printe;
run;
quit;
```

All of the multivariate test results appear in the HistWithin data set. This is because all multivariate test tables are named "MTests," although they occur in different directories in the output directory hierarchy.

Note that, even though there are also other tables named "ModelANOVA," the preceding ODS OUTPUT statement ensures that only the between-subject ANOVA appears in the HistBetween data set. The specific table is selected because of the additional specification of the partial path ("BetweenSubjects") in which it occurs. For more information on names and qualified path names, see the discussion in the section "Using the Output Delivery System" beginning on page 4.

In the following statements, a new data set, temp1, is created to contain the two data sets output in the preceding GLM run. They are displayed with no further processing.

```
ods listing;
title2 'Listing of Raw Data Sets';
data temp1;
   set HistBetween HistWithin;
run;
proc print;
run;
```

Output 14.10.1. Listing of the Raw Data Sets: Histamine Study

Histamine Study
Listing of Raw Data Sets

Obs	Dependent	Hypothesis Type	Source	DF	SS	MS	FValue	ProbF
1	BetweenSubjects	3	Drug	1	5.99336243	5.99336243	2.71	0.1281
2	BetweenSubjects	3	Depleted	1	15.44840703	15.44840703	6.98	0.0229
3	BetweenSubjects	3	Drug*Depleted	1	4.69087508	4.69087508	2.12	0.1734
4	BetweenSubjects	3	Error	11	24.34683348	2.21334850	_	_
5		.		.	.	.	24.03	0.0001
6		.		.	.	.	24.03	0.0001
7		.		.	.	.	24.03	0.0001
8		.		.	.	.	24.03	0.0001
9		.		.	.	.	5.78	0.0175
10		.		.	.	.	5.78	0.0175
11		.		.	.	.	5.78	0.0175
12		.		.	.	.	5.78	0.0175
13		.		.	.	.	21.31	0.0002
14		.		.	.	.	21.31	0.0002
15		.		.	.	.	21.31	0.0002
16		.		.	.	.	21.31	0.0002
17		.		.	.	.	12.48	0.0015
18		.		.	.	.	12.48	0.0015
19		.		.	.	.	12.48	0.0015
20		.		.	.	.	12.48	0.0015

Obs	Hypothesis	Error	Statistic	Value	NumDF	DenDF
1				.	.	.
2				.	.	.
3				.	.	.
4				.	.	.
5	Time	Error SSCP Matrix	Wilks' Lambda	0.11097706	3	9
6	Time	Error SSCP Matrix	Pillai's Trace	0.88902294	3	9
7	Time	Error SSCP Matrix	Hotelling-Lawley Trace	8.01087137	3	9
8	Time	Error SSCP Matrix	Roy's Greatest Root	8.01087137	3	9
9	Time_Drug	Error SSCP Matrix	Wilks' Lambda	0.34155984	3	9
10	Time_Drug	Error SSCP Matrix	Pillai's Trace	0.65844016	3	9
11	Time_Drug	Error SSCP Matrix	Hotelling-Lawley Trace	1.92774470	3	9
12	Time_Drug	Error SSCP Matrix	Roy's Greatest Root	1.92774470	3	9
13	Time_Depleted	Error SSCP Matrix	Wilks' Lambda	0.12339988	3	9
14	Time_Depleted	Error SSCP Matrix	Pillai's Trace	0.87660012	3	9
15	Time_Depleted	Error SSCP Matrix	Hotelling-Lawley Trace	7.10373567	3	9
16	Time_Depleted	Error SSCP Matrix	Roy's Greatest Root	7.10373567	3	9
17	Time_Drug_Depleted	Error SSCP Matrix	Wilks' Lambda	0.19383010	3	9
18	Time_Drug_Depleted	Error SSCP Matrix	Pillai's Trace	0.80616990	3	9
19	Time_Drug_Depleted	Error SSCP Matrix	Hotelling-Lawley Trace	4.15915732	3	9
20	Time_Drug_Depleted	Error SSCP Matrix	Roy's Greatest Root	4.15915732	3	9

In order to reduce the amount of information displayed in Output 14.10.1, the following data set, HistTests, is created. Only the observations from the raw data sets that are needed for interpretation are included. The variable Hypothesis in the HistWithin data set is renamed to Source, and the NumDF variable is renamed DF. The renamed variables correspond to the variable names found in the HistBetween data set.

```
data HistTests;
   set HistBetween(where =(Source     ^= "Error"))
       HistWithin (rename=(Hypothesis = Source NumDF=DF)
                   where =(Statistic  = "Hotelling-Lawley Trace"));
run;
proc print ;
title2 'Listing of Selections from the Raw Data Sets';
run;
```

Output 14.10.2. Listing of Selections from the Raw Data Sets: Histamine Study

Listing of Selections from the Raw Data Sets

Obs	Dependent	Hypothesis Type	Source	DF	SS	MS
1	BetweenSubjects	3	Drug	1	5.99336243	5.99336243
2	BetweenSubjects	3	Depleted	1	15.44840703	15.44840703
3	BetweenSubjects	3	Drug*Depleted	1	4.69087508	4.69087508
4		.	Time	3	.	.
5		.	Time_Drug	3	.	.
6		.	Time_Depleted	3	.	.
7		.	Time_Drug_Depleted	3	.	.

Obs	FValue	ProbF	Error	Statistic	Value	DenDF
1	2.71	0.1281			.	.
2	6.98	0.0229			.	.
3	2.12	0.1734			.	.
4	24.03	0.0001	Error SSCP Matrix	Hotelling-Lawley Trace	8.01087137	9
5	5.78	0.0175	Error SSCP Matrix	Hotelling-Lawley Trace	1.92774470	9
6	21.31	0.0002	Error SSCP Matrix	Hotelling-Lawley Trace	7.10373567	9
7	12.48	0.0015	Error SSCP Matrix	Hotelling-Lawley Trace	4.15915732	9

The amount of information contained in the HistTests is appropriate for interpreting the analysis (Output 14.10.2). However, you can further modify the presentation of the data by applying a template to this combined test data. A template specifies how data should be displayed. The output from previous ODS TRACE ON statements (for example, Output 14.4.2) shows that each table has an associated template as well as a name. In particular, the template associated with PROC GLM's ANOVA table is called 'Stat.GLM.Tests'.

You can use the 'Stat.GLM.Tests' template to display the SAS data set HistTests, as follows:

```
data _null_ ;
title2 'Listing of the Selections, Using a Standard Template';
   set HistTests;
   file print ods=(template='Stat.GLM.Tests');
   put _ods_;
run;
```

The ODS= option in the FILE statement enables you to use the DATA step to display a data set as a table. You do this by specifying data columns and associated attributes, such as the template specification.

The PUT statement contains the _ODS_ keyword. The keyword instructs the PUT statement to send the data values for all columns (as defined in the ODS= option in the FILE statement) to the open ODS destinations. For more information on using ODS in the DATA step, refer to *The Complete Guide to the SAS Output Delivery System*.

Output 14.10.3. Listing of the Data Sets Using a Standard Template

```
                          Histamine Study

Source                    DF            SS    Mean Square   F Value   Pr > F

Drug                       1    5.99336243     5.99336243      2.71   0.1281
Depleted                   1   15.44840703    15.44840703      6.98   0.0229
Drug*Depleted              1    4.69087508     4.69087508      2.12   0.1734
Time                       3             .              .     24.03   0.0001
Time_Drug                  3             .              .      5.78   0.0175
Time_Depleted              3             .              .     21.31   0.0002
Time_Drug_Depleted         3             .              .     12.48   0.0015
```

The data set contains the appropriate information, and it is presented in an easily understandable format, using the 'Stat.GLM.Tests' template.

Customizing Your Output

Suppose that you now want to modify the template used to format the ANOVA tables in order to emphasize significant effects. The following statements provide an example of how you can use the TEMPLATE procedure to

- redefine the format for the SS and Mean Square columns
- include the table title and footnote in the body of the table
- translate the missing values for SS and Mean Square in the rows corresponding to multivariate tests to asterisks (to refer to the footnote)
- add a column depicting the level of significance

For detailed information on using the TEMPLATE procedure, refer to the chapter titled "The Template Procedure" in the *SAS Procedures Guide*.

```
proc template;
   define table CombinedTests;
      parent=Stat.GLM.Tests;

      header "#Histamine Study##";
      footer "#* - Test computed using Hotelling-Lawley trace";

      column Source DF SS MS FValue ProbF Star;

      define SS;
         parent = Stat.GLM.SS;
         format = D7.3;
         translate _val_ = . into '   *';
      end;
      define MS;
         parent = Stat.GLM.MS;
         format = D7.3;
         translate _val_ = . into '   *';
      end;
      define Star;
         compute as ProbF;
         translate _val_ >  0.05  into "",
                   _val_ >  0.01  into "*",
```

```
                    _val_ >  0.001 into "**",
                    _val_ <= 0.001 into "***";
            pre_space=1 width=3 just=l;
        end;
     end;
  run;
```

The D*w.s* format, used in the preceding statements to redefine the SS and Mean Square columns, writes numbers in similar ranges with the same number of decimal places. In the format specification, *w* represents the width of the field and *s* represents the number of significant digits. Refer to the chapter on formats in *SAS Language Reference: Dictionary* for detailed information.

The following statements display the HistTests data set using the customized template. The results are displayed in Output 14.10.4.

```
data _null_;
   set HistTests;
   file print ods=(template='CombinedTests');
   put _ods_;
run;
```

Output 14.10.4. Listing of the Data Sets Using a Customized Template: Histamine Study

Histamine Study

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Drug	1	5.993	5.993	2.71	0.1281
Depleted	1	15.448	15.448	6.98	0.0229 *
Drug*Depleted	1	4.691	4.691	2.12	0.1734
Time	3	*	*	24.03	0.0001 ***
Time_Drug	3	*	*	5.78	0.0175 *
Time_Depleted	3	*	*	21.31	0.0002 ***
Time_Drug_Depleted	3	*	*	12.48	0.0015 **

* - Test computed using Hotelling-Lawley trace

Example 14.11. Creating HTML Output, Linked Within a Single Analysis

This example demonstrates how you can use ODS to provide links between different parts of your HTML procedure output.

Suppose you are analyzing a 4x4 factorial experiment for an industrial process, testing for differences in the number of defective products manufactured by different machines using different sources of raw material. The data set Experiment is created as follows:

```
data Experiment;
   do Supplier = 'A','B','C','D';
      do Machine = 1 to 4;
         do rep = 1 to 5;
            input Defects @@;
            output;
            end;
         end;
      end;
   datalines;
 2  6  3  3  6  8  6  6  4  4  4  2  4  0  4  5  5  7  8  5
13 12 12 11 12 16 15 14 14 13 11 10 12 12 10 13 13 14 15 12
 2  6  3  6  6  6  4  4  6  6  0  3  2  0  2  4  6  7  6  4
20 19 18 21 22 22 24 23 20 20 17 19 18 16 17 23 20 20 22 21
;
```

Suppose that you are interested in fitting a model to determine the effect that the supplier of raw material and machine type have on the number of defects in the products. If the *F*-test for a factor is significant, you would like to follow up with a multiple comparisons procedure. Thus, the tables of interest are the model ANOVA and the multiple comparisons output.

The following statements demonstrate how you can link a row of the ANOVA table to the corresponding multiple comparisons table. This is done by altering the display of values (inserting links) in the Source column of the ANOVA table. The links are inserted by using the TEMPLATE procedure.

```
proc template;
   edit Stat.GLM.Tests;
      edit Source;
         translate _val_ = "Supplier" into
              ('<a href="#IDX6">' || _val_ || '</a>'),
                   _val_ = "Machine"  into
              ('<a href="#IDX8">' || _val_ || '</a>');
         end;
      end;
run;
```

In order to determine the value to use in the HTML anchor link (<a href="">), you can run the analysis once and view information on your output in the Results node of

the SAS Explorer. The anchor name '#IDX6' is given to the table "ANOVA.Means.Supplier.Defects.MCLines.Tukey.MCLines" (the anchor name is automatically generated in the SAS run). The statements create the Supplier label as a link that, when clicked, opens the table of means from the "Tukey's Studentized Range Test for Defects" associated with the Supplier variable.

The '#IDX8' anchor name is given to the table "ANOVA.Means.Machine.Defects.MCLines.Tukey.MCLines". The statements create the Machine label as a link that, when clicked, opens the table of means from the "Tukey's Studentized Range Test for Defects" associated with the Machine variable.

The following statements specify that ODS close the SAS listing destination and open the HTML destination. ODS writes the HTML output to the file 'anovab.htm'.

```
ods listing close;
ods html body='anovab.htm' ;
```

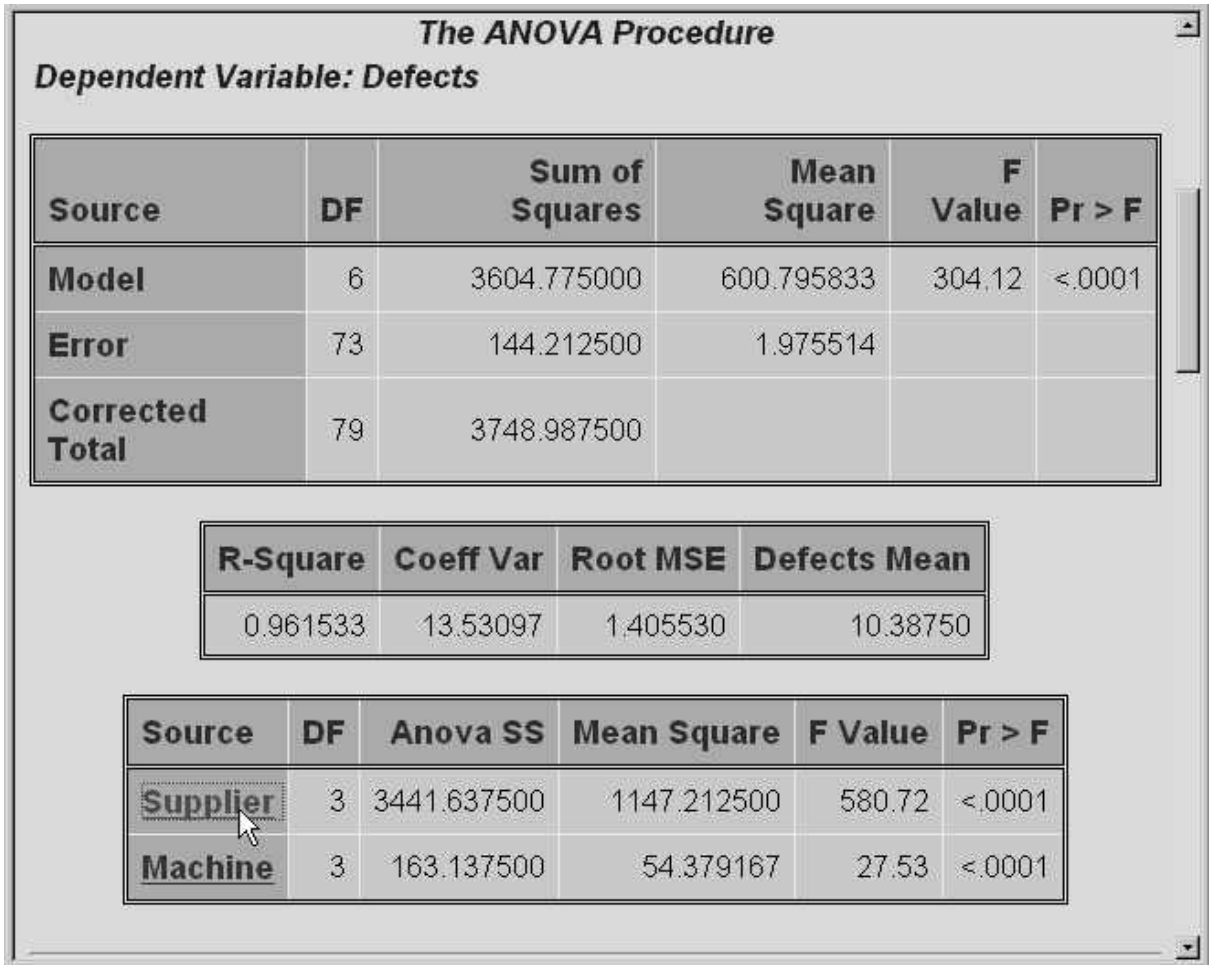
Since this is a balanced experiment, the ANOVA procedure computes the appropriate analysis, performed with the following statements:

```
proc anova data=Experiment;
   class Supplier Machine;
   model Defects = Supplier Machine;
   means Supplier Machine / tukey;
quit;

ods html close;
```

The output from the ANOVA procedure is displayed in Output 14.11.1.

Output 14.11.1. HTML Output from the ANOVA Procedure: Linked Output



The ANOVA Procedure

Dependent Variable: Defects

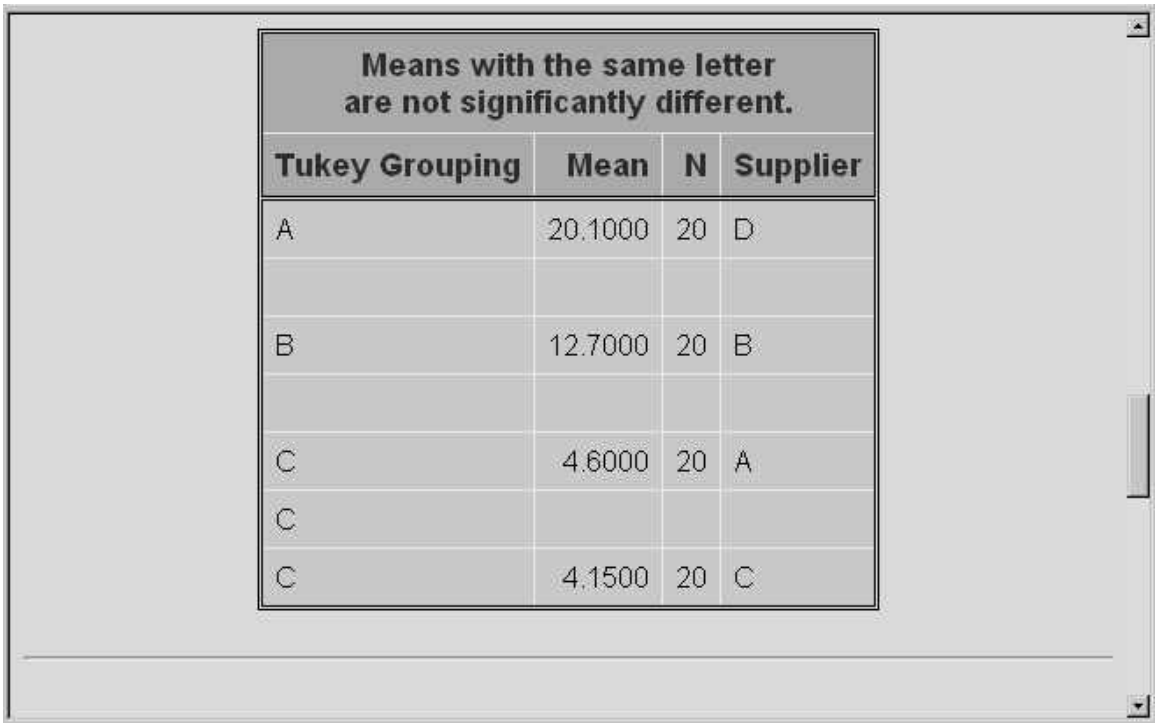
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	3604.775000	600.795833	304.12	<.0001
Error	73	144.212500	1.975514		
Corrected Total	79	3748.987500			

R-Square	Coeff Var	Root MSE	Defects Mean
0.961533	13.53097	1.405530	10.38750

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Supplier	3	3441.637500	1147.212500	580.72	<.0001
Machine	3	163.137500	54.379167	27.53	<.0001

The ANOVA procedure uses the "Stat.GLM.Tests" template to format the ANOVA table. The underlined text displayed in Output 14.11.1 shows the links in the table cells labeled as 'Supplier' and 'Machine.' Because of the modifications in the preceding statements, the Supplier table listing contains the HTML anchor reference to the tag 'IDX6.' When you click on the 'Supplier' link, the resulting multiple comparison table opens in your browser (Output 14.11.2). The links corresponding to the Machine variable operate similarly.

Output 14.11.2. Linked Output: Multiple Comparison Table from PROC ANOVA



Means with the same letter are not significantly different.			
Tukey Grouping	**Mean**	**N**	**Supplier**
A	20.1000	20	D
B	12.7000	20	B
C	4.6000	20	A
C			
C	4.1500	20	C

Example 14.12. Creating HTML Output, Linked Between Analyses

The following example demonstrates how you can use ODS to create links between different types of analyses.

The data in the following example are selected from a larger experiment on the use of drugs in the treatment of leprosy (Snedecor and Cochran 1967, p. 422).

Variables in the study are

Drug	- two antibiotics (A and D) and a control (F)
PreTreatment	- a pretreatment score of leprosy bacilli
PostTreatment	- a posttreatment score of leprosy bacilli

The data set is created as follows:

```
data drugtest;
   input drug $ PreTreatment PostTreatment @@;
   datalines;
a 11  6  a  8  0  a  5  2  a 14  8  a 19 11
a  6  4  a 10 13  a  6  1  a 11  8  a  3  0
d  6  0  d  6  2  d  7  3  d  8  1  d 18 18
d  8  4  d 19 14  d  8  9  d  5  1  d 15  9
f 16 13  f 13 10  f 11 18  f  9  5  f 21 23
f 16 12  f 12  5  f 12 16  f  7  1  f 12 20
;
```

The ODS HTML statement opens the HTML destination, specifies the body file name, requests that a table of contents be generated for the output, and specifies the file name of the frame to contain the body and table of contents. The NOGTITLE option in the ODS HTML statement specifies that titles are not to be included as an integral part of any generated graphics. For all graphics contained in the specified body file, titles appear in the body file and are external to graphics.

```
ods html body='glmb.htm'
         contents='glmc.htm'
         frame='glmf.htm'
         nogtitle;

ods output LSMeans=lsmeans;
```

The ODS OUTPUT statement writes the table of LS-means to the data set called lsmeans.

The GLM procedure is invoked to perform an analysis of covariance and compute LS-means for the variable Drug.

```
proc glm;
   class drug;
```

```
   model PostTreatment = Drug|PreTreatment / solution;
   lsmeans drug / stderr pdiff;
quit;
```

The following steps demonstrate how you can create links to connect the results of different analyses. In this example, the table of LS-means is graphically summarized with the GCHART procedure. In the steps that follow, each part of the resulting chart is linked to a plot that displays the relationship between the PostTreatment response variable and the PreTreatment variable.

The following DATA step creates a new variable called drugclick that matches each drug value with an HTML file. The variable drugclick is used in the subsequent GCHART procedure run. The variable provides the connection information for linking the two parts of the analysis together. The files referred to in these statements are created in a later step.

```
data lsmeans;
   set lsmeans;
   if Drug='a' then drugclick='HREF=drug1.htm';
   if Drug='d' then drugclick='HREF=drug2.htm';
   if Drug='f' then drugclick='HREF=drug3.htm';
run;
```

The following GOPTIONS and AXIS statements specify settings for the GCHART procedure. PROC GCHART is invoked, and the HBAR statement requests a horizontal bar chart for the variable drug. The length of the bars represent the value of the lsmean variable. The HTML option specifies the variable drugclick as the html linking variable to use. The FOOTNOTE1 and FOOTNOTE2 statements provide text that indicates how to use the links on the graph.

```
goptions ftext=swissb hsize=5.5in vsize=3.5in
        border cback=white;
axis1 minor=none label=(angle=90 rotate=0);
axis2 minor=none;

title f=swiss 'Chart of LS-means for Drug Type';
proc gchart data=lsmeans;
   hbar Drug/sumvar=lsmean type=mean
            frame cframe=ligr
            gaxis=axis1 raxis=axis2
            html=drugclick;
footnote1 j=l 'click on the bar to see a plot of PostTreatment';
footnote2 j=l 'versus PreTreatment for the corresponding drug';
format lsmean 6.3;
run;

ods html close;
run;
```

The preceding statements create a chart that summarizes the information from PROC GLM and that contains links to a second graphic analysis (using the variable drugclick and the HTML option in PROC GCHART).

The following statements provide that second analysis. The three files referred to by the drugclick variable are created as follows:

```
ods html body='drug1.htm' newfile=page;
symbol1 c=white v=dot i=r ;
title 'Plot of PostTreatment versus PreTreatment';
proc gplot data=drugtest uniform;
   plot PostTreatment*PreTreatment/frame cframe=ligr;
by Drug notsorted;
footnote;
run;
ods html close;
```

The NEWFILE option in the ODS HTML statement specifies that a new HTML file be created for each page of output. Note that page breaks occur only when a procedure explicitly starts a new page. The NEWFILE option also increments the filename for each new HTML file created, with the first filename corresponding to that given in the BODY= option, 'drug1.htm'.

The GPLOT procedure is invoked, producing a plot of the variable PostTreatment versus the variable PreTreatment for each value of the Drug variable. Thus, three plots are created, and each plot is contained in a separate html file. The files are named 'drug1.htm', 'drug2.htm' and 'drug3.htm'. The filenames match those filenames specified as values of the drugclick variable.

Output 14.12.1. Output from PROC GLM

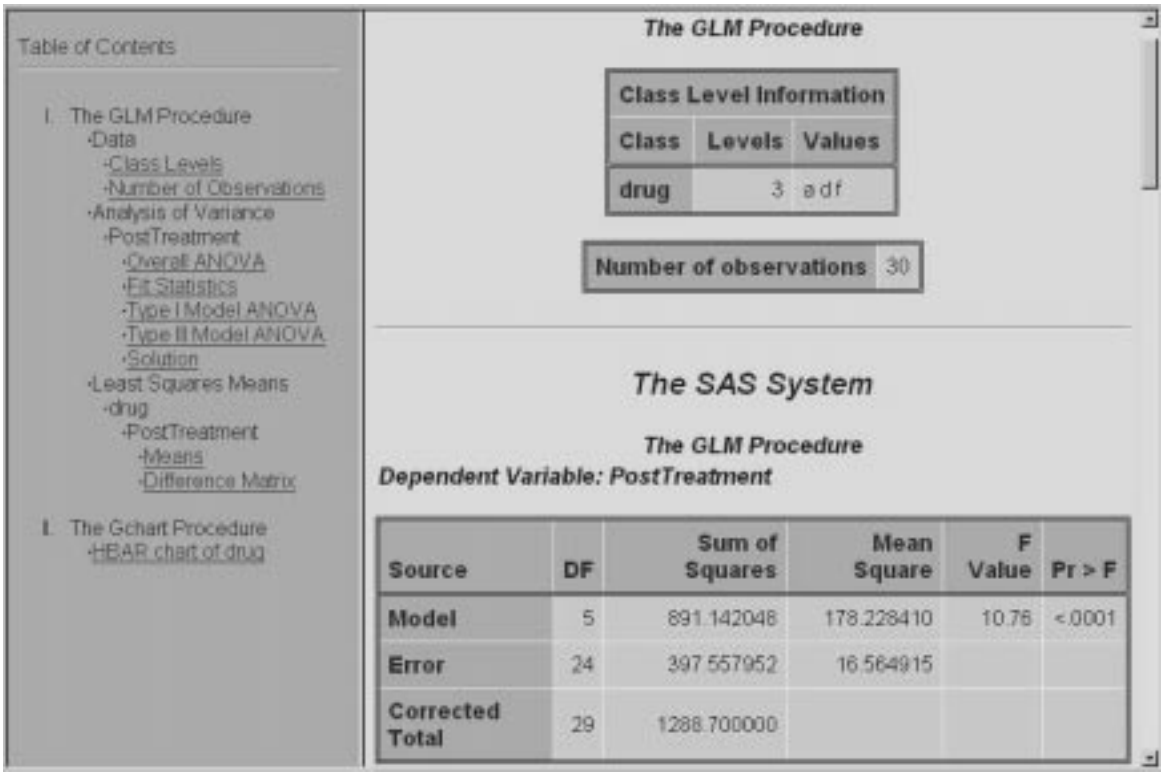


Table of Contents

I. The GLM Procedure
- Data
  - Class Levels
  - Number of Observations
- Analysis of Variance
  - PostTreatment
    - Overall ANOVA
    - Fit Statistics
    - Type I Model ANOVA
    - Type III Model ANOVA
    - Solution
- Least Squares Means
  - drug
  - PostTreatment
    - Means
    - Difference Matrix

II. The Gchart Procedure
- HBAR chart of drug

The GLM Procedure

Class Level Information		
Class	Levels	Values
drug	3	a d f

Number of observations	30

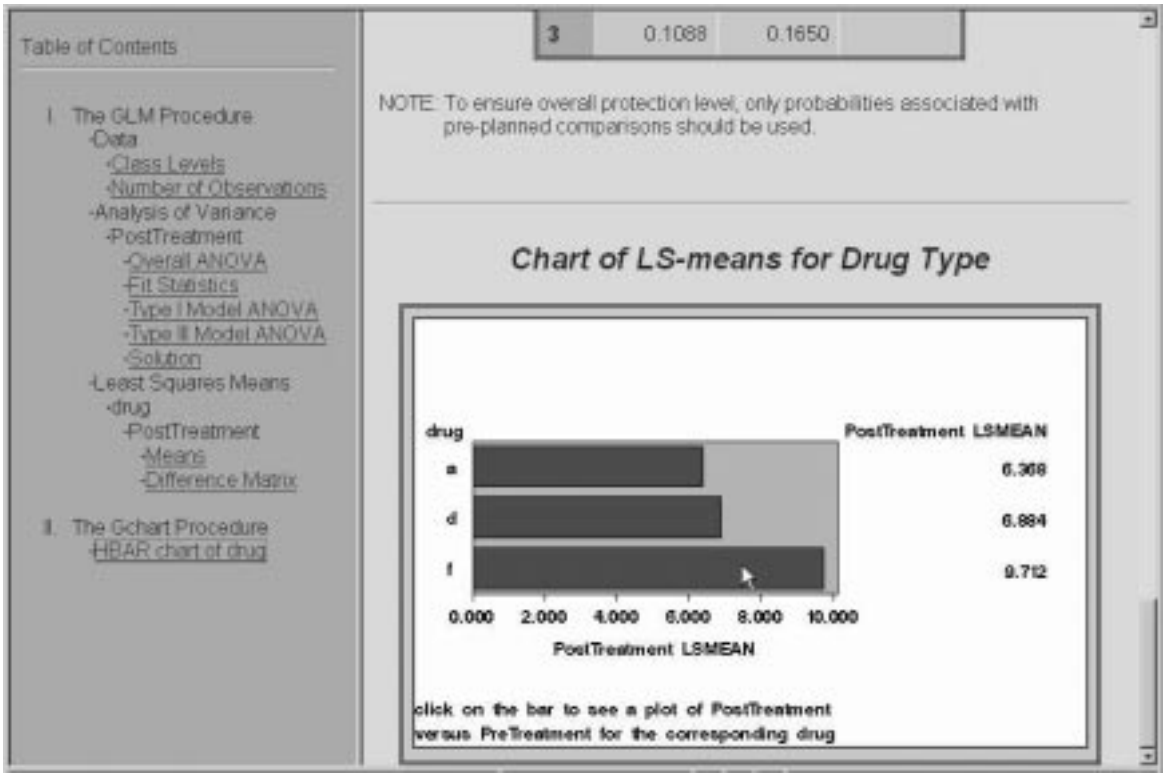
The SAS System

The GLM Procedure

Dependent Variable: PostTreatment

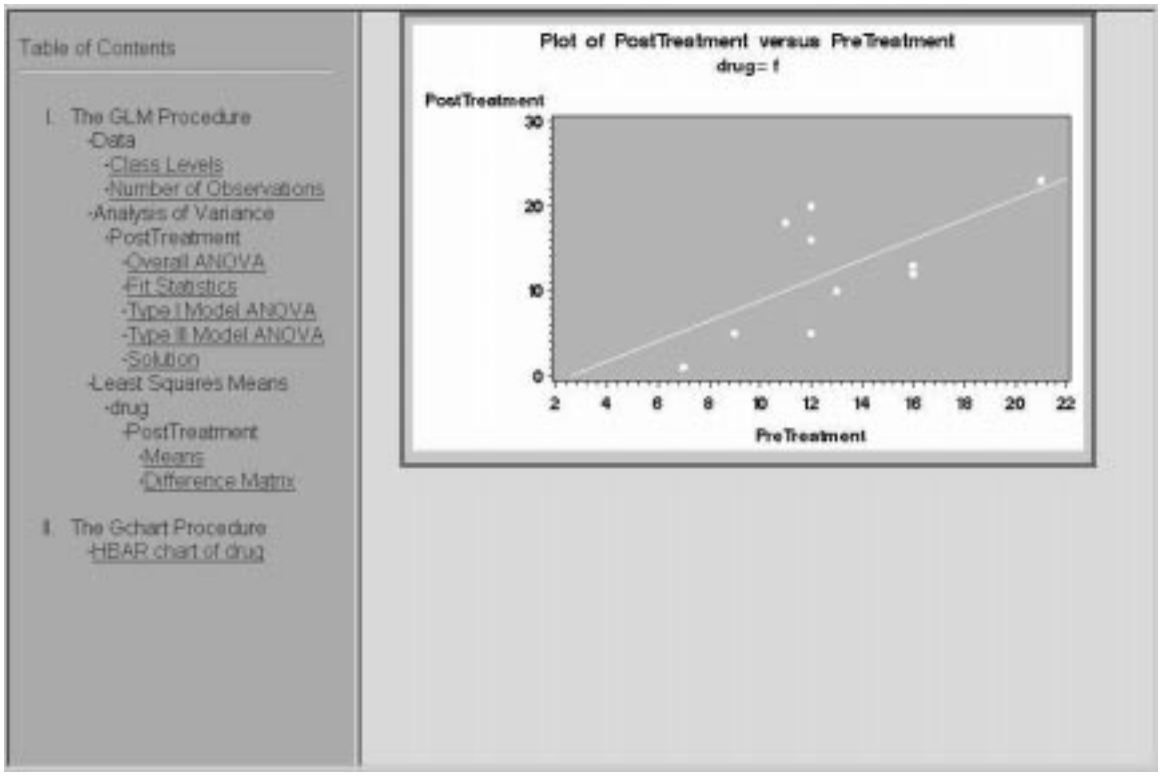
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	891.142048	178.228410	10.76	<.0001
Error	24	397.557952	16.564915		
Corrected Total	29	1288.700000			

Output 14.12.2. Bar Chart of LS-means by Drug Type: Linked Output



The graphic in Output 14.12.2 displays the difference in lsmeans for each drug type. When you click on a bar that represents a value of the variable drug, the browser opens the plot of PostTreatment versus PostTreatment that corresponds to that value of the variable Drug. Output 14.12.3 displays the plot corresponding to the drug type 'f'. You can view this graphic by clicking on the bottom bar in the bar chart in Output 14.12.2.

Output 14.12.3. Plot of PostTreatment versus PreTreatment for Drug Type 'f': Linked Output



References

Cole, J.W.L. and Grizzle, J.E. (1966), "Applications of Multivariate Analysis of Variance to Repeated Measures Experiments," *Biometrics*, 22, 810–828.

Hemmerle, W.J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Olinger, C. R. and Tobias, R. D. (1998), "It Chops, It Dices, It Makes Julienne Slices! ODS for Data Analysis Output As-You-Like-It in Version 7," *SAS Users Group International Conference Proceedings: SUGI 23*, Cary, NC: SAS Institute Inc.

Pothoff, R.F. and Roy, S.N. (1964), "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems," *Biometrika*, 51, 313–326.

Snedecor, G.W. and Cochran, W.G. (1967), *Statistical Methods*, Ames, IA: Iowa State University Press.