ODS for Data Analysis: Output As-You-Like-It in Version 7

Christopher R. Olinger
Randall D. Tobias
SAS Institute Inc., Cary NC

Abstract

This paper discusses advances made in the output generated by all Version 7 SAS® Software procedures. New features provided by the Output Delivery System (ODS) include automatic rendering of procedure output in HTML for the web, rich-text, Postscript or PCL format, output data sets from any printed output, and the ability to edit the look and feel of most output. ODS also supports all legacy line-mode printing and supports a document feature to allow post-procedure replaying of output. This talk introduces ODS features to be delivered in Version 7 and provides some concrete examples of how data analysts and statisticians can take advantage of the feature set provided by ODS.

Motivation

Every SAS user knows and loves the listing file, the vehicle for all printed SAS output for more than two decades. While familiar, the listing file has its limitations. For starters, output is restricted to monospace fonts only. On top of that, there isn't any tab information in the listing to let SAS users parse a table themselves. Industrious users have been forced to use all sorts of tricks to get SAS output into a document editor like Microsoft Word.

The limitations of the SAS listing stemmed from the fact that each procedure rendered its output to the listing file one line at a time. This outputting technique made it virtually impossible to add new output destinations (like HTML and RTF) to the SAS System as each procedure was wed to its own internal print routines. Adding new output destinations would require each procedure to support the output destination explicitly. One of the main goals of ODS was to get the statistical procedures out of the output formatting business — and, indeed, one of the nice things about Version 7 ODS is that it completely removes the burden of adding new output destinations from the procedures.

However, the listing file *is* handy for reading as a flat file. In fact, it is too handy. Have you ever had to read a listing file with a PROC PRINTTO and a _NULL_ DATA step just to rearrange some output or to grab some statistic that wasn't on the output data set? Did you ever think to yourself that there should be a better a way to do these sorts of things?

Enter ODS. The following table, generated by the RSREG procedure, was dropped into this document with no intermediate translation. Other ODS features make it possible to create a data set directly from this piece of output.

Factor	DF	Sum of Squares	Mean Square	F Value	Pr > F
Temperature	4	5258.016026	1314.504006	2.60	0.1613
GasLiquidRatio	4	11045	2761.150641	5.46	0.0454
PackingHeight	4	3813.016026	953.254006	1.89	0.2510

Figure 1: RSREG output produced as RTF

The ODS Solution

ODS provides a more flexible mechanism for procedures to produce output. Instead of writing to the listing file directly, the procedure creates an *output object* for each piece of output to be displayed. Every table, plot, and chart is thus a separate entity in the system. Each output object can be further broken down into two component parts:

- a *data* component, the collection of raw numbers and characters comprising the contents of the output
- a *template* component, the description of what the output is supposed to look like when the data is formatted and arranged on a page

Each piece of output exists in a logical hierarchy of output directories that is created as your SAS job executes. For instance, consider the following example.

```
data a;
   do a = 1 to 4;
      do b = 1 to 4;
         do c = 1 to 4;
            n = int(6*ranuni(1));
               do i = 1 to n;
                  y = a + b + rannor(1);
                  x = a * b / rannor(3);
                  output;
                  end;
               end;
            end;
         end;
      end;
   run;

proc glm data=a;
   class a b c;
   model x y = a|b|c / ss1 ;
   run;
```

The output from the GLM procedure is arranged in the following output directory structure. Items in **bold** are directories and items in *italics* are output object names:

GLM
- **Data**
  - *Class Levels*
  - *Number of Observations*
- **Analysis of Variance**
  - **Dependent Variable X**
    - *Overall ANOVA*
    - *Fit Statistics*
    - *Type I Model ANOVA*
  - **Dependent Variable Y**
    - *Overall ANOVA*
    - *Fit Statistics*
    - *Type I Model ANOVA*

Every piece of output that a procedure produces has a name, a label, and a directory path describing where the output resides in the directory structure.
Output objects are created by the procedure, and they are rendered based on the *output destinations* that the user has requested. An output destination describes the type of formatting requested for the output (HTML, RTF, and so forth). You can control which output destinations are in effect by using the **ODS** statement or by selecting output destinations in the SAS Explorer.

Three new features afford users control of ODS output: the ODS statement, the TEMPLATE procedure, and the ODS Output Document.

- Using the ODS statement, you can set output destination attributes, select or exclude individual pieces of output, request that pieces of output be turned into a data set, and control which templates should be used for rendering purposes.

- The TEMPLATE procedure enables you to specify the attributes that describe how a piece of output should be formatted, as well as specify stylistic elements for output destinations. You can specify cell formats and headers, change column ordering, specify colors and fonts, add GIF images to HTML, and specify various other formatting and stylistic characteristics.

- And, finally, there is support for a persistent form of your output. The ODS Output Document enables you to save your output in raw form, modify the appearance of a piece of output, and replay it to any output destination. The Document also enables you to pick and choose individual pieces of output and combine

them into more complex reports. You can do all of this without rerunning the procedure that initially created the output.

Details

All procedures have been converted to produce their output as ODS output objects.[1] You can control these objects and interact with ODS in a variety of ways. The primary ways to interact with the system are through the ODS statement, the SAS Explorer, PROC TEMPLATE, and PROC OUTPUT.

The ODS Statement

The ODS statement is a global statement that enables you to control ODS in batch mode. Using the ODS statement, you can specify options for output destinations, control the templates that are in effect for your SAS job, and select and exclude various pieces of output. You can also use the ODS statement to display the names of individual pieces of output as they are generated.

Output Destinations

Each output destination *generates* a particular type of output. HTML, output data sets, and Postscript are all separate output destinations. ODS is designed to allow multiple output destinations to be active at the same time. A single SAS job can, therefore, create both HTML and the SAS listing concurrently.

The initial release of Version 7 will contain support for the following output destinations:

- the standard SAS listing
- HTML for the Web
- output data sets

Soon after the initial release of Version 7, SAS Institute will be releasing support for the following output destinations:

- the ODS Output Document for modifying and replaying output without rerunning the procedure that created it
- RTF for inclusion in Microsoft Word
- PS and PCL for high fidelity printers
- *possibly* SGML and LaTex

You select an output destination by specifying the appropriate ODS statement options before your SAS job is executed. As an example, consider the following statements that compute a response surface regression analysis and render the results as RTF. The table produced by this job is included at the start of this paper (Figure 1).

```
ods rtf file='example.rtf' ;
ods select factorANOVA;

data a;
   input Odor Temperature GasLiquidRatio PackingHeight @@;
   datalines;
66  -1  -1   0     39  1  -1   0     43  -1   1  0     49 1 1 0
58  -1   0  -1     17  1   0  -1     -5  -1   0  1    -40 1 0 1
65   0  -1  -1      7  0   1  -1     43   0  -1  1    -22 0 1 1
-31  0   0   0    -35  0   0   0    -26   0   0  0
run;

proc rsreg;
   model odor=Temperature GasLiquidRatio PackingHeight;
run;

ods rtf close;
```

See the “Examples” section for some examples of other output destinations.

Tracing and Selecting Output

You may have noticed in the previous example that you can select and exclude individual pieces of output with the ODS statement. Each piece of output has associated with it a name, a label, a directory path, and a

[1] Some procedures are more ODS compliant than others. The PLOT procedure, for instance, does not separate data and template information. It relies on what is known as “batch mode” - the old SAS listing is captured and passed along as the output object’s data.

template path. You can display this information by using the ODS **TRACE** option. Once you know the name or label of a piece of output, you can select it, exclude it, or turn it into a SAS data set. See the "Examples" section for a demonstration.

The SAS Explorer

The SAS Explorer is a new feature in the Version 7 SAS Display Manager that enables you to explore the various pieces and parts of the SAS System. Part of the SAS Explorer is the **RESULTS** node. The RESULTS node retains a running record of the output directory hierarchy that is produced when a SAS job is executed. Figure 2 shows the Explorer with the results for an analysis of variance. By clicking on the output labels in the Explorer you can link directly to the output in the output window or 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 pieces of output (tables in this case).

The Explorer also enables you to set ODS options and edit templates. All of the options available in the ODS statement can be set from the Explorer. The template editor is a dialogue-based editor that allows you to set values for all of the template options.

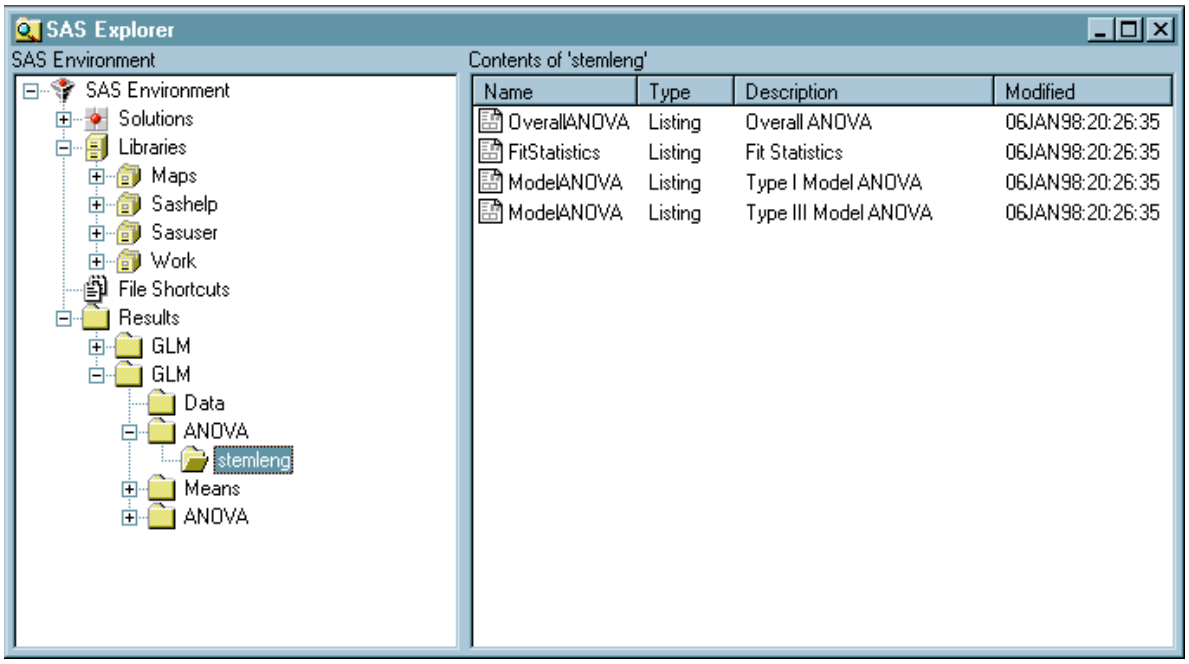


Figure 2: The SAS Explorer

The Template Procedure

A *template* is an abstract description of what a piece of output is supposed to look like when it is formatted. A template contains attributes describing column ordering, style information, justification, cell formats, and other formatting characteristics. Most output objects have a template associated with them. You can find the template associated with a particular piece of output by looking at the ODS **TRACE** record. For an example, please see Figure 8.

The TEMPLATE procedure enables you to create and edit templates in batch mode. Templates are fully editable, so you can change the appearance of most output in the system. You can edit a template in Display Manager mode by using the new SAS Explorer.

SAS System templates are stored in the SASHELP library. When you run PROC TEMPLATE to modify or edit a template, the template is stored in the SASUSER library. You can modify the concatenation path that ODS uses to look up templates with the ODS **PATH** statement. In this manner you can set up a default set of templates for your organization to use that modifies the look and feel of SAS output. Once you have edited a template for a piece of output, the output continues to use that template until the template is deleted.

For an example of editing and using templates, please see the "Examples" section.

Styles

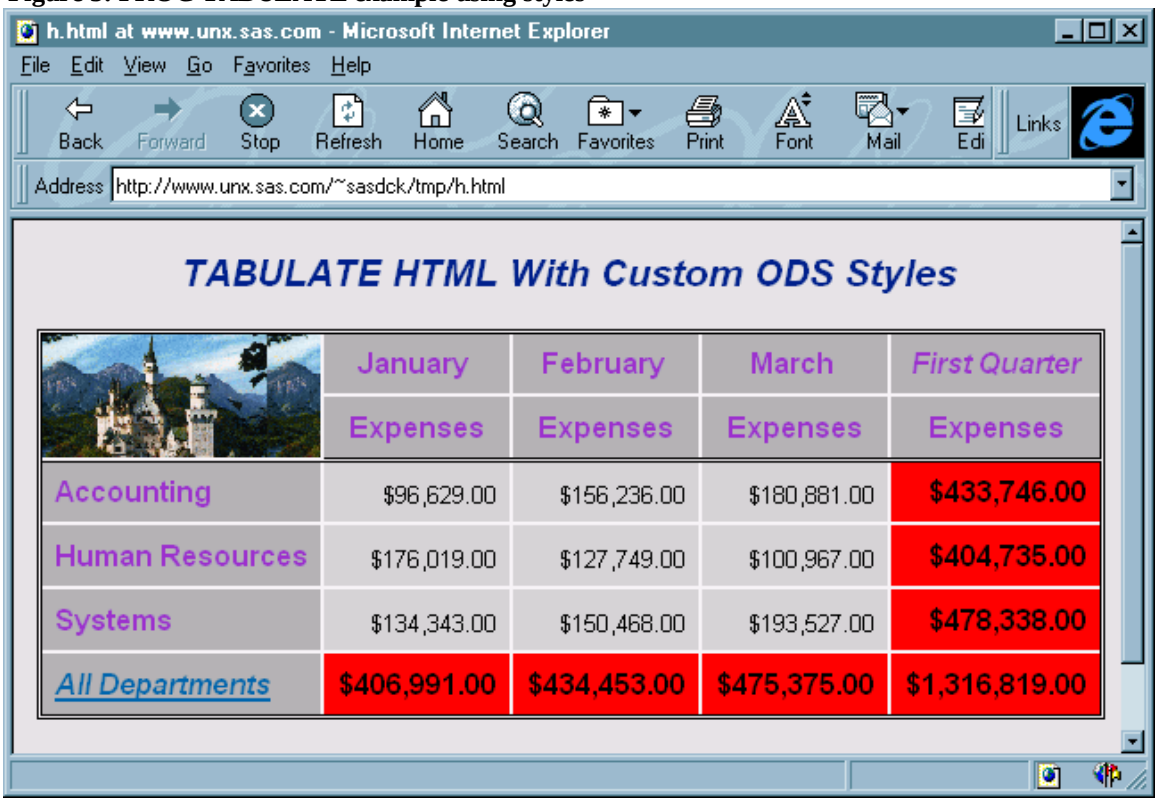
A style template is mostly concerned with stylistic issues such as colors, fonts, and presentation attributes. A style can be specified only for output destinations that support them. At present, the only destinations that

support styles are HTML, RTF, and PS/PCL. The SAS output window does not participate in the style scheme, so if you want to use styles you need to use an output destination that fully supports them.

It is important to note the difference between a style template and a normal table template. A style applies to an entire SAS job, and a table template applies only to the tables that reference the template. Use of styles ensures a consistent look and feel across all output from the SAS System. You can, however, reference style information in templates for individual headers and data cells.

You can specify a style to use for an output destination using the ODS statement. As an example of what you can do with styles, see the PROC TABULATE output in Figure 3.

Figure 3: PROC TABULATE example using styles



The ODS Output Document

Have you ever wanted to reformat output from a procedure without actually having to rerun the procedure? If so then the Document is for you. The Document is a persistent representation of the output that is produced by your SAS job. Output is stored completely in raw form. This enables you to change formatting characteristics (formats, column headers, and so on) without rerunning the procedure that created the output. The Document also enables you to browse and modify SAS output from previous runs, combine output into custom reports, and access your output from client-server applications.

The OUTPUT Procedure

The OUTPUT procedure enables you to manipulate ODS Output Documents in batch mode. There are commands to reorder output in the output directory hierarchy, to move and copy output to new directories, and to replay output to the various ODS output destinations. For Display Manager users, there is support for managing Output Documents in the SAS Explorer. All of the functionality in PROC OUTPUT will be available in the Explorer.

Examples

Using Output Delivery in a Pharmaceutical Trial

Introduction

This example uses a pre-clinical drug experiment to demonstrate how ODS gives you unprecedented control over SAS output. The data is taken from Cole and Grizzle (1966). In order to study the effect of two different

drugs on histamine levels in the blood, the drugs were given to 13 animals, some with depleted histamine levels, and the levels of histamine in the animals' blood was 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.

```
title "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
;
```

The appropriate analysis for this type of experiment is a *repeated measures analysis*, which assumes that responses from different animals are independent but responses at different times for the same animal are correlated. The following statements use PROC GLM 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 the GLM chapter of the SAS/STAT® documentation.)

```
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;
quit;
```

Delivering Output on the Web

Repeated measures analyses produce a *lot* of output, including univariate and multivariate ANOVAs, correlation matrices, within- and between-subject tests, and so on. In this case, the output starts with the class level information and goes on for over 200 lines on five or six pages. So your first question might be: “How can I get all that output to my colleagues without having to put all those pages in the mail?” ODS provides the answer: you can produce the GLM output directly as HTML and put it on your company's intranet. All you have to do is name the file where you want the HTML to be written, as shown in the following statements:

```
ods html body = 'HistGLM.html';
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;
quit;
ods html close;
```

The **`ods html body=`** option specifies the file where the HTML output is to be written, and the **`ods html close`** option directs the system to stop writing HTML. The results use HTML's built-in table formatting facilities to make output that looks right on any browser (Figure 4).

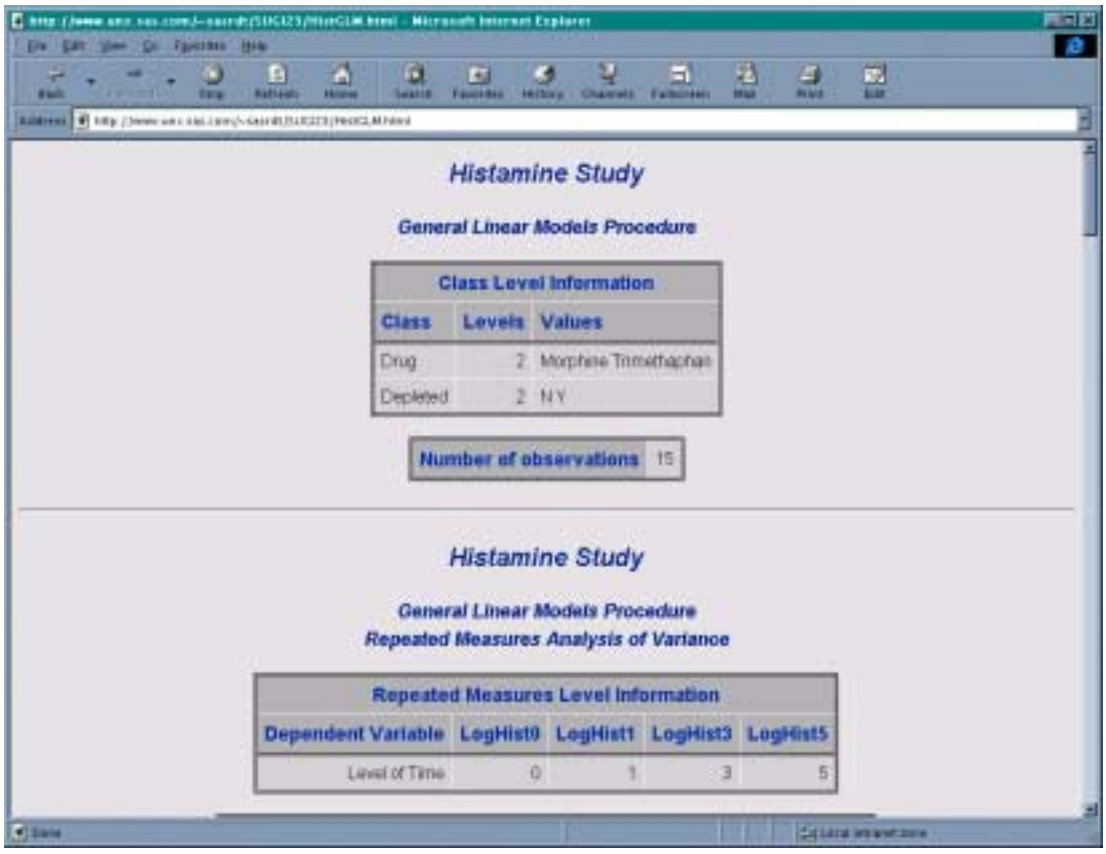


Figure 4: HTML output for GLM repeated measures analysis

Because there are so many tables to make sense of, a table of contents would be helpful. Fortunately, ODS can produce that, too. Just name another file to contain the table of contents, as shown in the following statements:

```
ods html body     = 'HistGLM.html'
         contents = 'HistGLMContents.html';
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;
quit;
ods html close;
```

The resulting Web page (Figure 5) contains a road map for the GLM output.

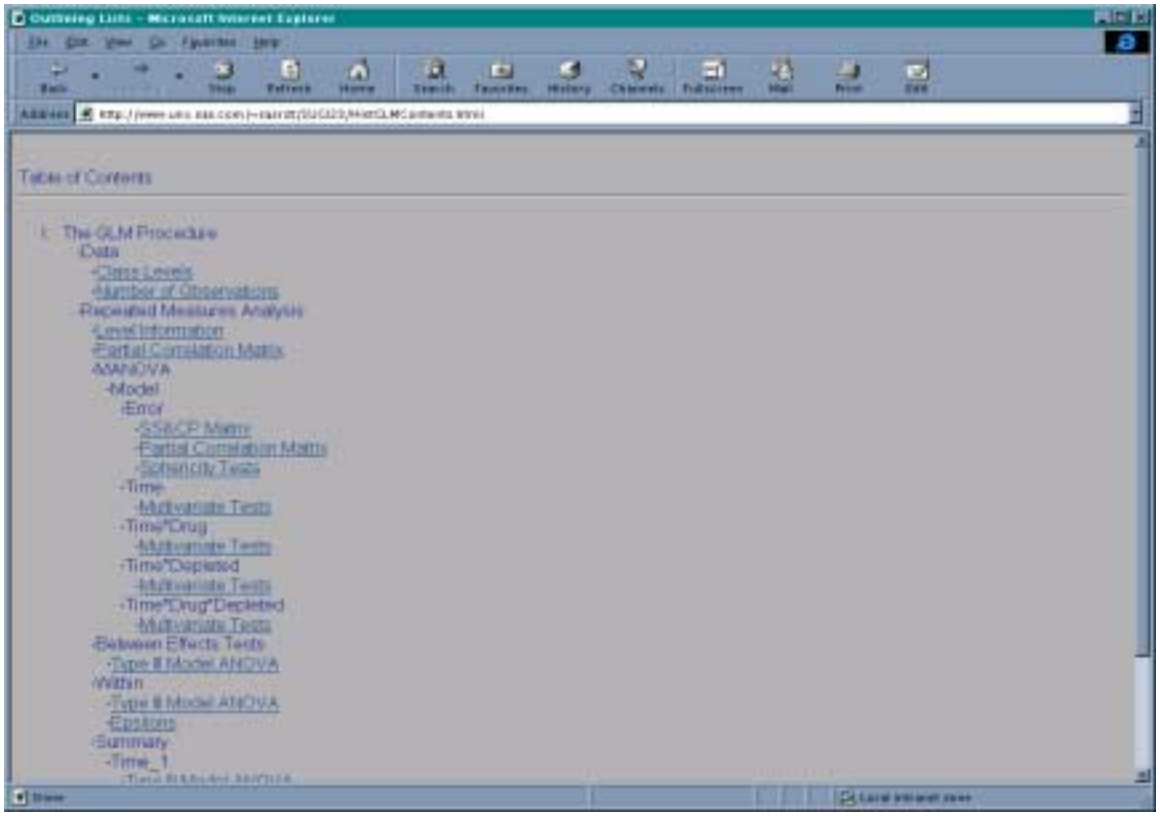
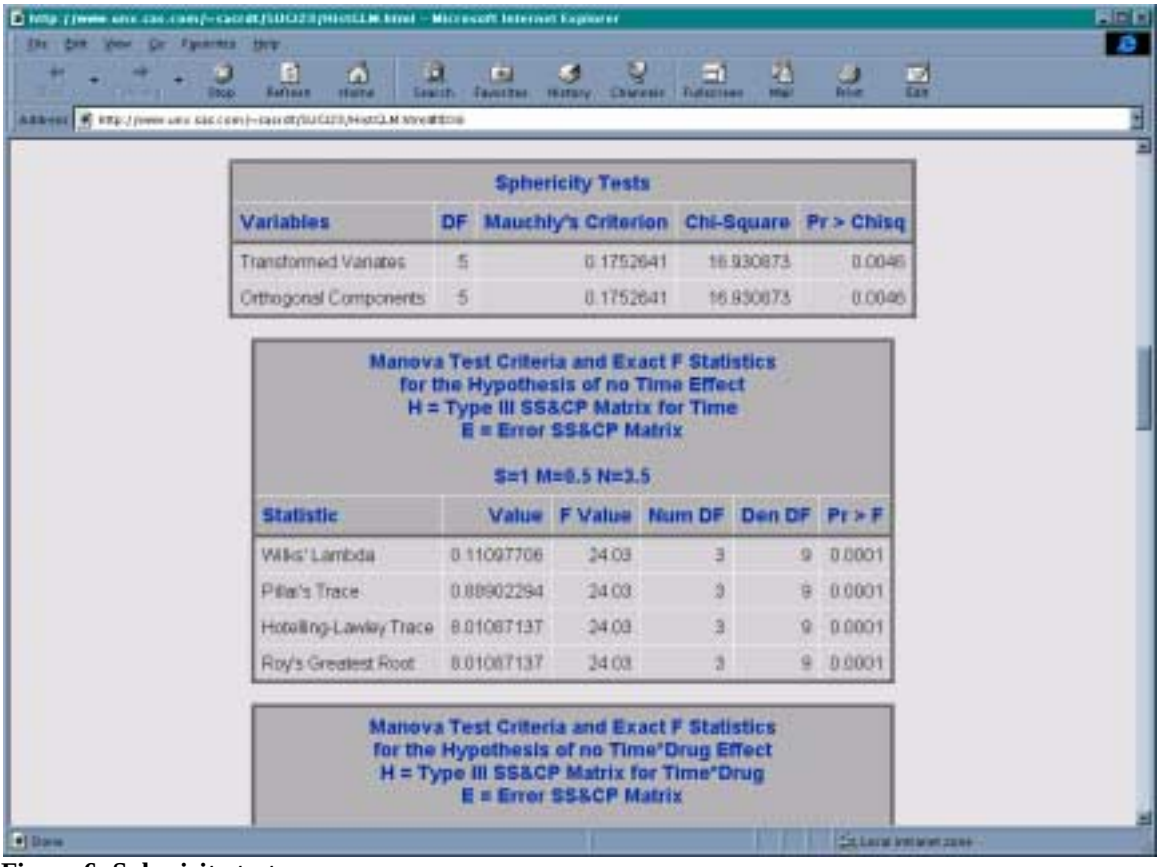


Figure 5: HTML table of contents for GLM repeated measures analysis

Moreover, the contents are linked to the body of the output. That is, if you click on "Sphericity Tests," your browser takes you right to the test for a spherical covariance structure between the within-subject measurements (Figure 6). Note that the sphericity tests are significant, indicating that you should use the multivariate tests for within-subject effects. You can even make a Web page that displays the contents and the body of the output at the same time, using HTML framing, as demonstrated in the following statements.

```
ods html body     = 'HistGLM.html'
         contents = 'HistGLMContents.html'
         frame    = 'HistGLMFrame.html';
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;
quit;
ods html close;
```

The resulting Web page is shown in Figure 7.



Sphericity Tests

Variables	DF	Mauchly's Criterion	Chi-Square	Pr > Chisq
Transformed Variates	5	0.1752641	16.930873	0.0046
Orthogonal Components	5	0.1752641	16.930873	0.0046

Manova Test Criteria and Exact F Statistics for the Hypothesis of no Time Effect
H = Type III SS&CP Matrix for Time
E = Error SS&CP Matrix

S=1 M=0.5 N=3.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.11097706	24.03	3	9	0.0001
Pillai's Trace	0.88902294	24.03	3	9	0.0001
Hotelling-Lawley Trace	8.01067137	24.03	3	9	0.0001
Roy's Greatest Root	8.01067137	24.03	3	9	0.0001

Manova Test Criteria and Exact F Statistics for the Hypothesis of no Time*Drug Effect
H = Type III SS&CP Matrix for Time*Drug
E = Error SS&CP Matrix

Figure 6: Sphericity tests

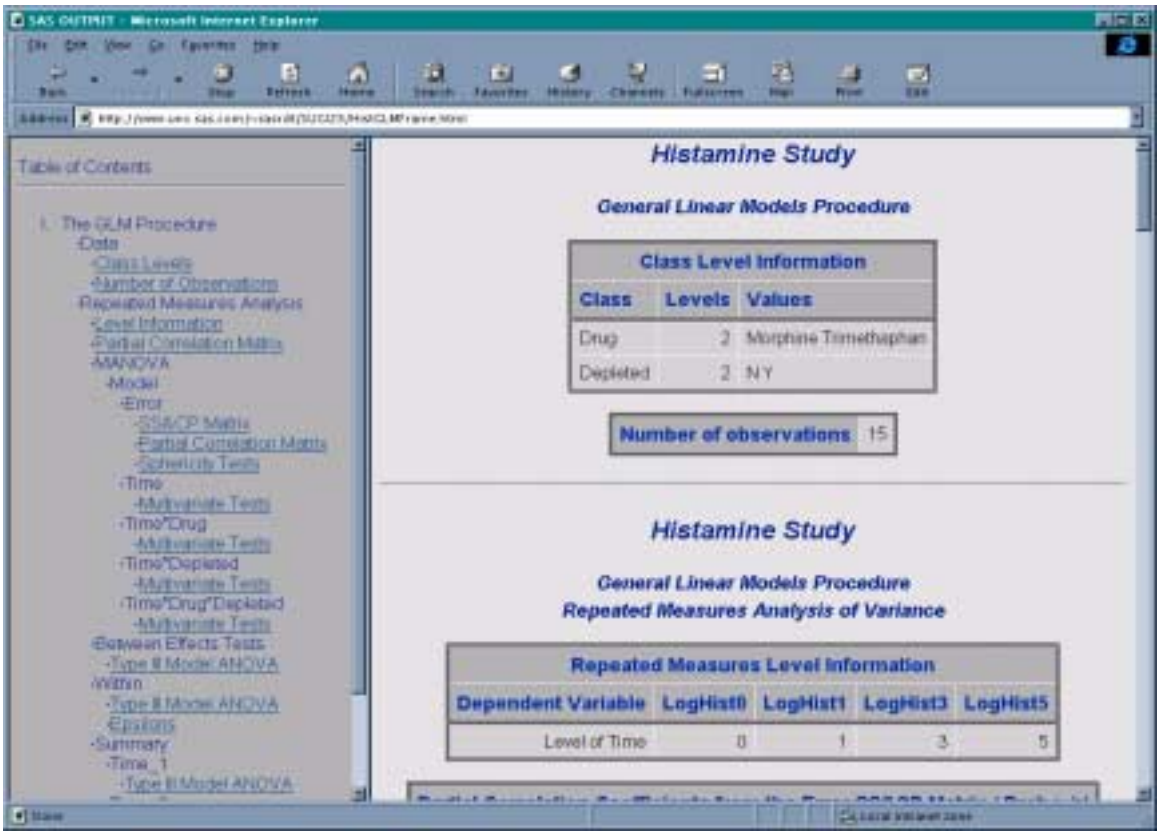


Table of Contents

- The GLM Procedure
  - Data
    - Class Levels
    - Number of Observations
  - Repeated Measures Analysis
    - Level Information
    - Partial Correlation Matrix
    - ANOVA
      - Model
      - Error
        - SS&CP Matrix
        - Partial Correlation Matrix
        - Sphericity Tests
      - Time
        - Multivariate Tests
      - Time*Drug
        - Multivariate Tests
      - Time*Depleted
        - Multivariate Tests
      - Time*Drug*Depleted
        - Multivariate Tests
    - Between Effects Tests
      - Type III Model ANOVA
    - Within
      - Type III Model ANOVA
      - Epsilons
    - Summary
      - Time_1
        - Type III Model ANOVA

Histamine Study

General Linear Models Procedure

Class Level Information

Class	Levels	Values
Drug	2	Morphine Trimethaphan
Depleted	2	N Y

Number of observations 15

Histamine Study

General Linear Models Procedure
Repeated Measures Analysis of Variance

Repeated Measures Level Information

Dependent Variable	LogHist0	LogHist1	LogHist3	LogHist5
Level of Time	0	1	3	5

Figure 7: Combined table of contents and repeated measures analysis

Delivering Just the Output You Want

Even though the HTML output with a table of contents helps to keep the GLM results in order, there might be more information than you need. You can also use ODS to display just the tables you want to see.

Each piece of output has a name and a path, corresponding to where it occurs in the table of contents. To limit which tables are displayed, you simply tell ODS the names of the tables you want to see. The easiest way to find out the names of a procedure's tables is to use the **ods trace output** option. This option displays the name (and other information) associated with each piece of output as it is produced. This is demonstrated in the following statements.

```
ods trace output;
proc glm data=Histamine;
   class Drug Depleted;
   model LogHist0–LogHist5 = Drug Depleted Drug*Depleted;
   repeated Time 4 (0 1 3 5) polynomial / summary printe;
quit;
ods trace off;
```

For example, the information displayed for the between-subjects ANOVA is shown in Figure 8.

Suppose that you want just these tables to be displayed. All you have to do is tell ODS to select only tables with these names:

```
ods select Mtests BetweenSubjects.ModelANOVA;
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;
quit;
```

Only the four tables with the multivariate tests for the within-subject effects and the table for the between-subject tests are displayed. All four multivariate test tables are displayed, since they all have the name "Mtests," even though they occur in different directories in the output directory hierarchy. On the other hand, "BetweenSubjects.ModelANOVA" refers just to the model ANOVA table that occurs in the "BetweenSubjects" directory.

```
Output Added:
-------------
Name:       Mtests
Label:      Multivariate Tests
Template:   stat.GLM.Mtests
Path:       GLM.Repeated.MANOVA.Model.Time_Drug_Depleted.MTests
-------------


              Manova Test Criteria and Exact F Statistics for
               the Hypothesis of no Time*Drug*Depleted Effect
              H = Type III SS&CP Matrix for Time*Drug*Depleted
```

```
                          E = Error SS&CP Matrix

                          S=1    M=0.5    N=3.5

 Statistic                      Value   F Value   Num DF   Den DF   Pr > F

 Wilks' Lambda             0.19383010     12.48        3        9   0.0015
 Pillai's Trace            0.80616990     12.48        3        9   0.0015
 Hotelling-Lawley Trace    4.15915732     12.48        3        9   0.0015
 Roy's Greatest Root       4.15915732     12.48        3        9   0.0015

Output Added:
-------------
Name:       ModelANOVA
Label:      Type III Model ANOVA
Template:   stat.GLM.Tests
Path:       GLM.Repeated.BetweenSubjects.ModelANOVA
-------------


                             Histamine Study

                    General Linear Models Procedure
                Repeated Measures Analysis of Variance
             Tests of Hypotheses for Between Subjects Effects

 Source                    DF   Type III 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
 Error                     11   24.34683348    2.21334850
```

Figure 8: Listing output with interleaved TRACE OUTPUT results

Delivering Output to Data Sets

The displayed output may still be more verbose than you want. You may want to reorganize the information in the five tables selected previously into one table with a line for each effect. If you had the different tables in data sets, you could use the extensive data manipulation facilities of the SAS language to get the job done.

SAS procedures have always made the most important statistics available in auxiliary output data sets, but with the advent of Output Delivery, *every number* in *every table* of *every procedure* can be accessed via a data set. The syntax is very similar to that for selecting specific tables to display. Simply name the tables to be saved and equate them to the names of data sets in which they should be saved.

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

All the multivariate test results appear in the **HISTWITHIN** data set. This is because all the multivariate test tables are named "MTests," although they occur in different directories in the output directory hierarchy. There are also other tables named "ModelANOVA," but the above statement makes sure you get only the between-subject ANOVA in the **HISTBETWEEN** data set by additionally specifying the group in which it occurs.

Delivering Combined Output

So, now that you have the tests in data sets, how can you display all the information in one over-all summary table? The following statements show how to combine the two data sets into a single data set, dropping the row corresponding to "Error" from the between-tests and renaming a couple of variables in the within-tests data set. Also, you retain the results of just one of the four multivariate tests.

```
data HistTests;
   set HistBetween(where =(Source     ^= "Error")                )
       HistWithin (rename=(Hypothesis = Source NumDF=DF)
                   where =(Statistic  = "Hotelling-Lawley Trace"));
   run;
```

Now, you can use the DATA step to apply a *template* to the combined test data. A template specifies how an abstract rectangle of data should be displayed. In the output from the **ods trace** output statement before, you saw that each table has a template as well as a name. In particular, the template associated with the ANOVA tables is "Stat.GLM.Tests". Looking at Figure 8 you can see that the GLM procedure uses this template to format ANOVA tables, and you can use it, too. Suppose the combined data set is named **HISTTESTS**. The following statements "render" this data set using the "Stat.GLM.Tests" template.

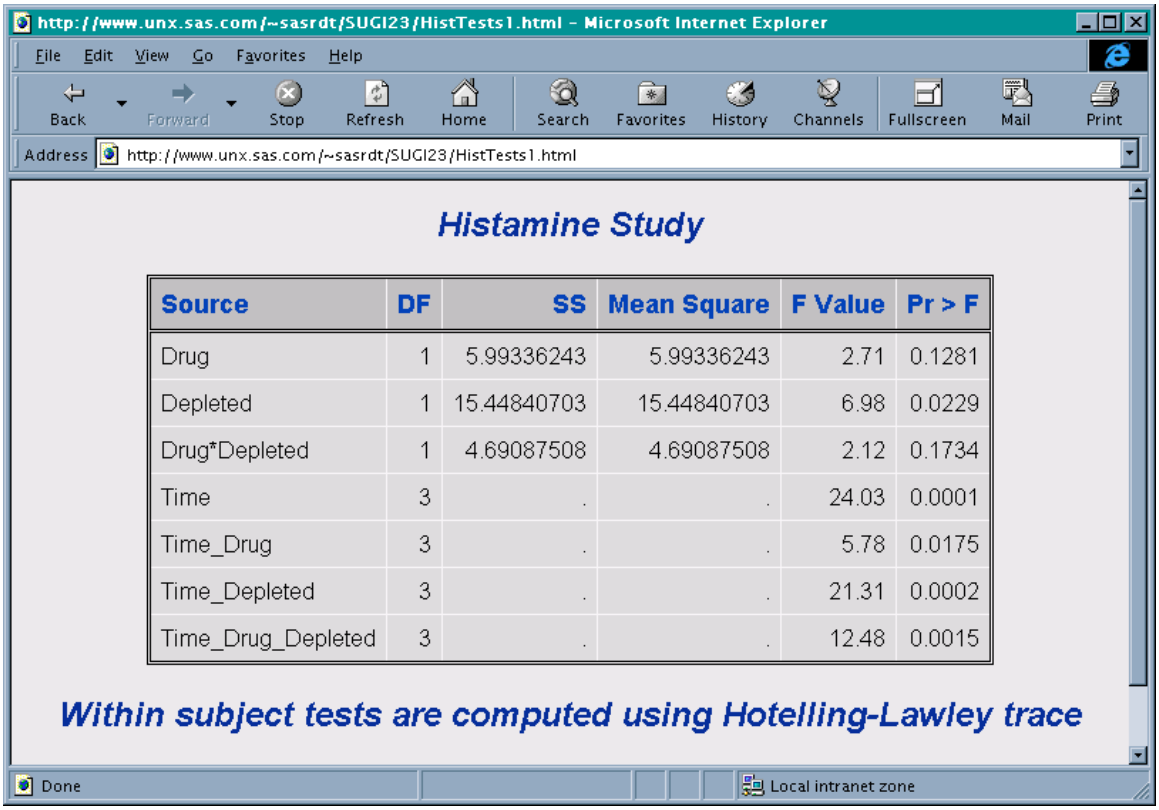
```
data _null_;
   set HistTests;
   file print ods=(template='Stat.GLM.Tests');
   put _ods_;
   run;
```

The results look just like any other ANOVA table displayed by the GLM procedure. You can dress them up a bit by adding TITLE and FOOTNOTE statements and displaying them as HTML, as shown in the following statements.

```
ods html body = 'HistTests1.html';
title "Histamine Study";
footnote1
"Within-subject tests are computed using Hotelling-Lawley trace";
data _null_; set HistTests;
   file print ods=(template='Stat.GLM.Tests');
   put _ods_;
   run;
title;
footnote1;
ods html close;
```

The results are shown in Figure 9. The footnote explains why the within-subject tests have missing SS and MS columns. For more information on how to use the DATA step to interface with ODS, please refer to Heffner (1998).

Figure 9: Combined tests for within- and between-subject effects



Delivering Customized Output

For historical reasons, the GLM procedure displays sums of squares and mean squares with many digits by default. Suppose that you want to display them with fewer digits. It is easy to redefine the format used to display a column in a template. For example, the following statements copy the current "Stat.GLM.Tests" template, changing the format for printing these two columns.

```
proc template;
   edit Stat.GLM.Tests as CombinedTests;
      edit SS; format=D7.3; end;
      edit MS; format=D7.3; end;
      end;
   run;
```

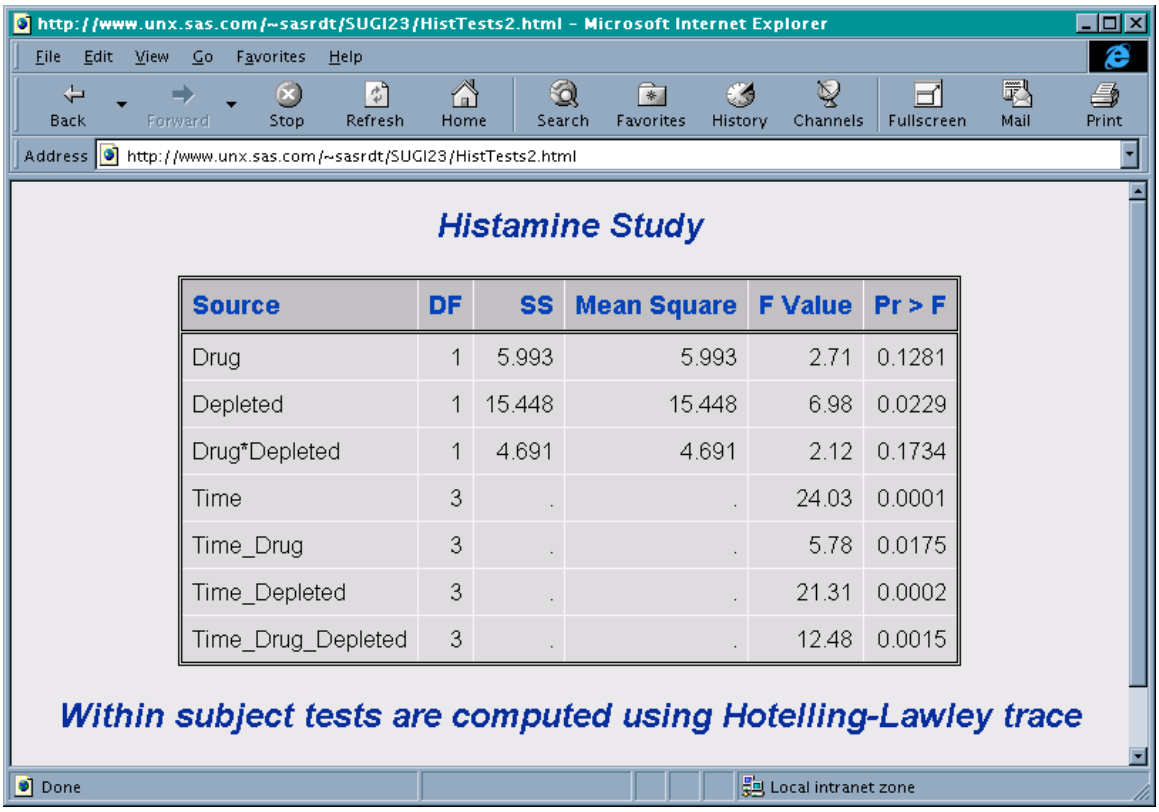
You can apply this new template to the combined test data in the same way that you applied the predefined template.

```
title "Histamine Study";
footnote1
   "Within-subject tests are computed using Hotelling-Lawley trace";
ods html body = 'HistTests2.html';
data _null_;
   set HistTests;
   file print ods=(template='CombinedTests');
   put _ods_;
```

```
   run;
ods html close;
title;
footnote1;
ods html close;
```

The results are shown in Figure 10.

Figure 10: Combined tests with reformatted SS and MS columns



To make the analysis really easy to interpret, you'd like to emphasize effects that are significant at the 0.02 level. This you can also do with the template used to display the results. The following statements use the template to

- redefine the format for the SS and MS columns, as shown previously
- include the table title and footnote in the body of the table
- translate the missing values for SS and MS in the rows corresponding to multivariate tests to asterisks, referring to the footnote
- emphasize significant *p*-values with a bold font
- add a column depicting the level of significance

```
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;
```

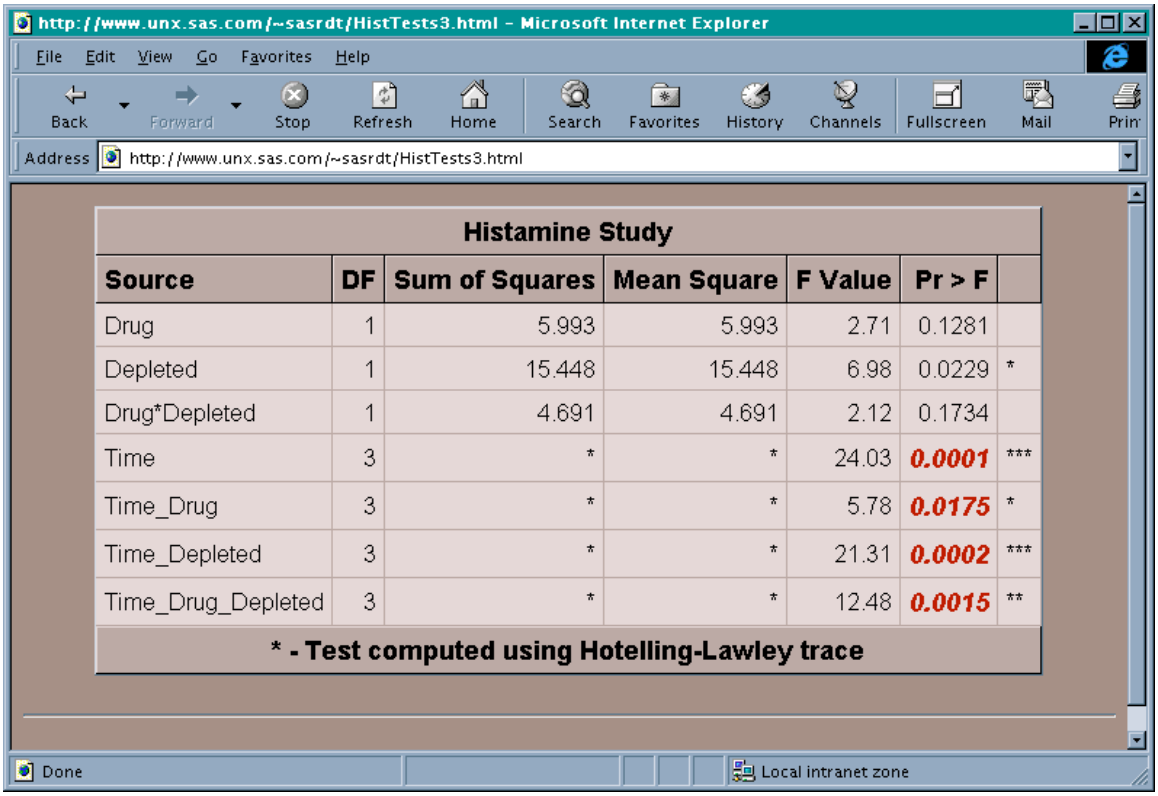
```
            translate _val_ = . into '   *';
            end;
         define MS;
            parent = Stat.GLM.MS;
            format = D7.3;
            translate _val_ = . into '   *';
            end;
         define ProbF;
            parent = Stat.GLM.ProbF;
            cellstyle _val_ < 0.02 as DataEmphasis;
            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;

   ods html body = 'HistTests3.html' style=Brown;
   data _null_;
      set HistTests;
      file print ods=(template='CombinedTests');
      put _ods_;
      run;
   ods html close;
```

The results are shown in Figure 11.

If you want to change the way all ANOVA tests are displayed by PROC GLM, you can redefine the templates that the procedure uses. For example, in order to have the SS and MS columns always displayed with fewer digits, simply redefine the root columns used by the procedure to display them:

```
   proc template;
      edit Stat.GLM.SS; format=D7.3; end;
      edit Stat.GLM.MS; format=D7.3; end;
   run;
```



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**

Figure 11: Combined tests with special formatting

Using Output Delivery in an Industrial Experiment

Introduction

The preceding sections demonstrate how you can use ODS to customize, manipulate, and deliver just the SAS procedure output you want directly to your colleagues. So far the examples focus on results that can be summarized in a single table. When statistical results are necessarily more complicated, you can use ODS to link different parts together. You can even create HTML for your intranet that mixes and links text and graphics.

Suppose you are analyzing a 4×4 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.

```
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
;
```

If the F-test for a factor is significant, you would like to follow up with a multiple comparisons procedure. Since this is a balanced experiment, the ANOVA procedure computes the appropriate analysis, as shown in the following statements:

```
proc anova data=Experiment;
```

```
   class Supplier Machine;
   model Defects = Supplier Machine;
   means Supplier Machine / tukey;
quit;
```

Delivering Linked Output

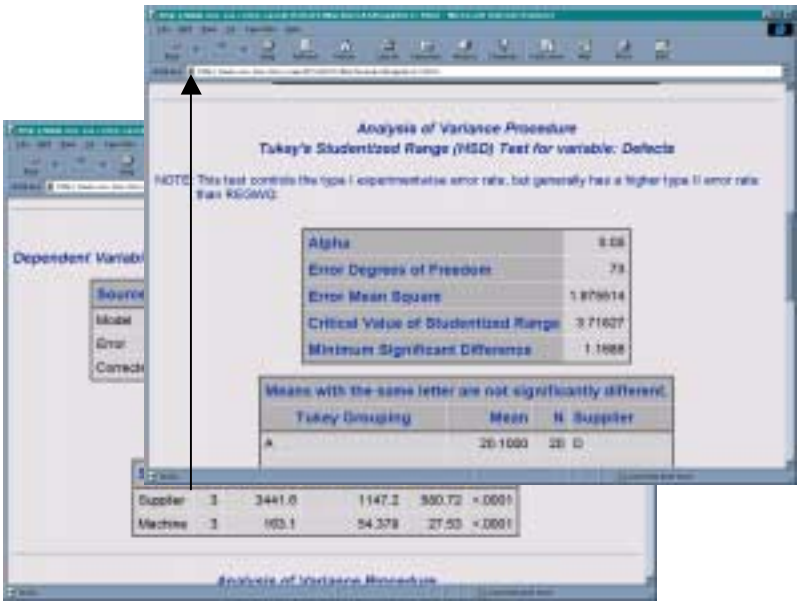
The tables of interest are the model ANOVA and the multiple comparisons output. You can link a row of the ANOVA table to the corresponding multiple comparisons table with the following template:

```
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;
ods html body="MachinesAndSuppliers1.html";
proc anova data=Experiment;
   class Supplier Machine;
   model Defects = Supplier Machine;
   means Supplier Machine / tukey;
quit;
ods html close;
```

The results are shown in Figure 12. The arrow indicates a link from the row for Supplier in the ANOVA table to the multiple comparisons analysis for Supplier.

Figure 12: Linked output

Alternatively, you can use Gabriel's procedure to display the multiple comparisons results as intervals around the estimated means. The following statements link the rows of the ANOVA table to graphics that enable you to judge whether two means compare equal on the basis of whether their corresponding intervals overlap. Note that this example not only uses table reformatting to insert links, but also outputs the tables from Gabriel's analysis in a graphical form using the SHEWHART procedure. The results are shown in Figure 13.



```
proc template;
   edit Stat.GLM.Tests;
      edit Source;
         translate
           _val_ = "Supplier" into
              '<a href="#IDX13"> '||_val_||" </a>",
           _val_ = "Machine"  into
              '<a href="#IDX14"> '||_val_||" </a>";
         end;
      end;
   run;

ods html body="MachinesAndSuppliers2.html";

ods output CLMeans = CLMeans;            /* Produce analysis of variance, */

proc anova data=Experiment;              /* saving Gabriel's analysis in  */
   class Supplier Machine;               /* a data set.                   */
   model Defects = Supplier Machine;
   means Supplier Machine / gabriel clm;
quit;

goptions target=gif hsize=6in vsize=6in  /* Use PROC SHEWHART to plot     */
         ftext=swissb htext=0.25in;      /* Gabriel's analysis for        */

data M_CLMeans; set CLMeans;             /* Machine.                      */
   drop Dependent Effect Method;
   where   (Effect='Machine')
         & (Method='Gabriel');
   rename Level  =Machine
          LowerCL=DefectsL
          Mean   =DefectsX
          UpperCL=DefectsH
          N      =DefectsN;
   label Level = 'Machine';
   DefectsS = (UpperCL - LowerCL)/10;
   DefectsM = Mean;
   Defects1 = LowerCL;
   Defects3 = UpperCL;

proc sort data=M_CLMeans out=M_CLMeans;
   by Machine;

title "Comparison of Machine Means";

proc shewhart history=M_CLMeans;
   boxchart Defects*Machine /
      nolimits
      nolegend
      cframe = ligr
      cboxes = black
      cboxfill = orange;
   run;

data S_CLMeans; set CLMeans;             /* Use PROC SHEWHART to plot     */
   drop Dependent Effect Method;         /* Gabriel's analysis for        */
   where   (Effect='Supplier')           /* Supplier.                     */
         & (Method='Gabriel' );
   rename Level  =Supplier
          LowerCL=DefectsL
          Mean   =DefectsX
          UpperCL=DefectsH
          N      =DefectsN;
   label Level = 'Supplier';
   DefectsS = (UpperCL - LowerCL)/10;
   DefectsM = Mean;
   Defects1 = LowerCL;
```

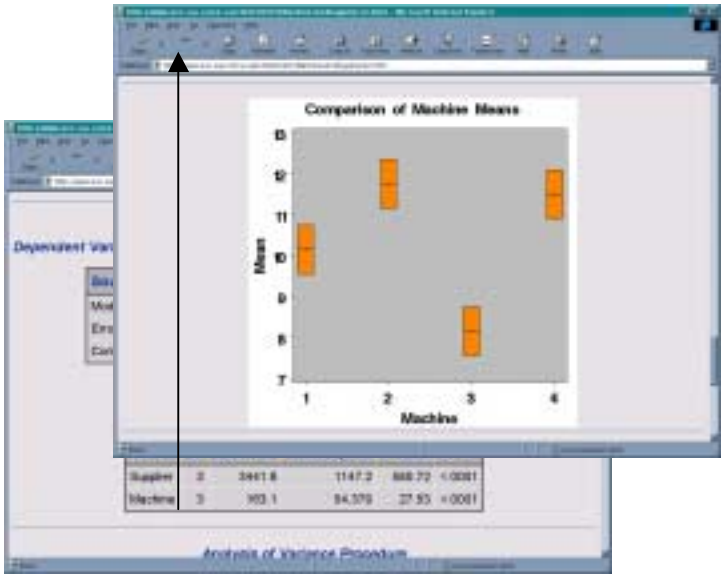
```
   Defects3 = UpperCL;

proc sort data=S_CLMeans out=S_CLMeans;
   by Supplier;

title "Comparison of Supplier Means";

proc shewhart history=S_CLMeans;
   boxchart Defects*Supplier /
      nolimits
      nolegend
      cframe = ligr
      cboxes = black
      cboxfill = orange;
run;

ods html close;
```



Compatibility Issues with Version 6 ODS Prototype

- In Version 6, a few procedures such as the MIXED procedure and the GENMOD procedure use a prototype ODS and use the MAKE statement to create data sets from output objects. This statement remains supported in these procedures, but all other procedures achieve this objective with the global ODS OUTPUT statement.

- The Version 6 prototype of the ODS output hierarchy is stored in a SAS catalog. Version 7 SAS Software has a more flexible *item-store* file type used to store templates and ODS Output Documents. There are currently no plans for a conversion procedure.

- The Version 6 prototype ODS uses two macro variables (_DISK_ and _PRINT_) to regulate the saving of an output hierarchy. Version 7 uses the global ODS statement to indicate this.

- The Version 6 and Version 7 PROC TEMPLATE and PROC OUTPUT syntax are not compatible.

Your Turn

The SAS Institute staff that brings you ODS is very excited about the potential of these new capabilities of the SAS System. You can send electronic mail with your comments or to request a copy of example HTML files to ods@unx.sas.com.

We will be adding more information about ODS capabilities to the Research and Development section of SAS Institute's Web Page. See http://www.sas.com/rnd

You can find more information about various aspects of ODS by referring to Heffner (1998) and Patel (1998).

Acknowledgements

ODS is the work of many individuals in the Applications Division, IPD Department, and Publications Division.

References

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

Heffner, W. F. (1998), "ODS: The Data Step Knows," in the *Proceedings for the Twenty-Third Annual SAS Users Group International Conference,* Cary, NC: SAS Institute Inc.

Patel, H. (1998), "Using SAS/GRAPH® Software to Create Graphs on the Web," in the *Proceedings for the Twenty-Third Annual SAS Users Group International Conference,* Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1989), *SAS/STAT User's Guide, Version 6, Fourth Edition, Volume 2*, Cary, NC: SAS Institute Inc.