OBTAINING P-VALUES USING PROC PRINTTO AND PROC TABULATE

Michael M. Brown, Syntex Research
Karl Konnerth, Syntex Research

ABSTRACT

SAS procedures can produce copious amounts of data. A problem of too much data can develop when information from these data are extracted by hand for a report. Obtaining p-values produced by PROC T-TEST on many treatments and variables is an example of this problem. Thus, it is necessary to find programming techniques that will extract this information. By combining PROC PRINTTO with SAS data step programming and PROC TABULATE, obtaining information such as p-values will require less effort.

By first sending the results of T-TEST to a dataset on disk and then by reading in the data, using an INFILE statement, the programmer can search the data using various programming statements in order to find the desired values. Once these values have been found, output to the current dataset, and printed for verification, PROC TABULATE can then be used to create a table of p-values, thus saving hours of tedious hand work.

INTRODUCTION

Because obtaining p-values from PROC T-TEST output has proven to be a tedious task, a program using SAS to obtain these values became necessary. By combining steps such as PROC PRINTTO with PROC TABULATE and SAS DATA STEP programming, it is not only easier to extract p-values, but other data values as well. This paper will illustrate how the above steps can be used to create a table using p-values obtained from the PROC T-TEST output.

USING PROC PRINTTO TO OUTPUT T-TEST RESULTS

Since this program is concerned with finding the p-values that result from performing a T-test on all possible paired comparisons, a dataset containing all the possible pairings must be prepared. Using the variable GROUP, each pair will be named, stored, and output.

Data Allpairs:
Set RandomData:

Group = '01 Treat1 vs. Treat2';
If TreatMnt = 1 or TreatMnt = 2 Then Output;
Group = '02 Treat1 vs. Treat2';
If TreatMnt = 1 or TreatMnt = 3 Then Output;
Group = '03 Treat2 vs. Treat3';
If TreatMnt = 2 or TreatMnt = 3 Then Output;

Now that a dataset containing all pairs have been prepared, PROC T-TEST will be performed, and the results will be written to disk. E.g.:

Proc Printto Unit=20
Proc T-Test Data= Allpairs;
By Group;
Class Treatmnt;
Var Var1-Var9;
Proc Printto;

The first PROC PRINTTO initializes the procedure to output the data to disk. The UNIT option references the data set contained in the JCL that will be named and created by this procedure. The JCL statement containing //FT2OF001 will thus be matched by the UNIT=20 option. The program will then perform the T-TEST procedure on each pair and subsequently send the results to the disk file. The disk file should look something like Figure 1. The second PROC PRINTTO is used to stop the output procedure.

USING SAS PROGRAMMING TO EXTRACT P-VALUES

Now that T-TEST results have been sent to the disk file, they will need to be brought back into the program so that the p-values can be extracted:

Data Process (Keep = Group Variable T Prob)
Infile FT2OF001;
Do While (Group = '');
Input Line $Char133;
FINDgrp = Index (Line, 'GROUP');
If FINDgrp > 0 Then
Group = Substr (Line, FINDgrp + 8, 19);
End:

The INFILE FT2OF001 statement refers the program to the dataset in the JCL that was previously created by the PROC PRINTTO statements and tells SAS to retrieve that data so that it can be used in the present dataset.

Once the data has been made available to SAS, DO statements and SAS programming are used to find the required information.

The first DO statement is used to input a value for the variable GROUP, which is included on the T-Test output. (See Figure 1). In order to do this, a line of data is input as one variable, and a search continues until the string "GROUP" is found. Once the string has been found, a substring of the following data is taken and that value is
then assigned to the variable GROUP. That value will be like the group values assigned when outputting the paired comparisons, e.g. '01 TREAT1 VS. TREAT2.

The following loops involve finding the variable name, i.e., var1, var2...., and the actual p-value. The following procedure to find the variable name uses the same techniques as previously explained, except that this one is used to find the variable name, e.g., VAR1, VAR2... Since there were nine variables that had T-TESTS done, a 00 statement is added so that the program will get the nine variables names, and eventually, the p-values that correspond to each variable name.

DO 1= 1 to 9;
DO WHILE (VARIABLE = " ");
INPUT LINE $CHAR133;
FINDVAR = INDEX (LINE, 'VARIABLE:');
IF FINDVAR > 0 THEN
VARIABLE = SUBSTR (LINE, FINDVAR+10, 8);
END;

The following statements are used to input various values that can be found on the line containing the value "EQUAL" for the variable VARIANCE, and thus, enables the extraction of p-values. Note that the line is broken up into variables and input separately, rather than all at once (See Figure 1).

Figure 1.
PROC TABULATE CAN BE USED TO CREATE A TABLE

Once the desired variables have been extracted from the T-Test output a variety of tables containing this data may be created using SAS procedures. The following SAS statements using PROC TABULATE were used to create the table in Figure 3. In this set of statements, a table has been created listing the p-values for (VAR1-VAR9) by their groupings (TREAT01 VS TREAT02). The ORDER=DATA keeps the values in the order that they are encountered. The FORMCHAR value causes the table to be printed without delineating lines. The RTS option specifies the number of print positions allotted to row titles.

If the program is intended to be run in batch modes, the JCL must refer to datasets in the following order: SAS dataset, the PROC PRINTTO output dataset, and the SAS dataset again. Additionally, the blocksize must be optimized if the record format of the output dataset is going to be fixed block, e.g., if the maximum blocksize is 19069 and the logical record length is 133, then the optimal blocksize would be 19019 (143 * 133 = 19019).

(Sample JCL)

```sas
PROC TABULATE ORDER=DATA FORMCHAR=''
CLASS GROUP VARIABLE;
TABLE VARIABLE, GROUP*PROB
RTS=20;
```

Table 18

<table>
<thead>
<tr>
<th>GROUP</th>
<th>VARIABLE</th>
<th>PROB</th>
<th>PROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR1</td>
<td>0.4294</td>
<td>0.0318</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR2</td>
<td>0.0324</td>
<td>0.4268</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR3</td>
<td>0.3504</td>
<td>0.4278</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR4</td>
<td>0.3504</td>
<td>0.4278</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR5</td>
<td>0.3504</td>
<td>0.4278</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR6</td>
<td>0.3504</td>
<td>0.4278</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR7</td>
<td>0.3504</td>
<td>0.4278</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR8</td>
<td>0.3504</td>
<td>0.4278</td>
</tr>
<tr>
<td>TREAT01 VS TREAT02</td>
<td>VAR9</td>
<td>0.3504</td>
<td>0.4278</td>
</tr>
</tbody>
</table>

Figure 2.

**Table 19**

<table>
<thead>
<tr>
<th>HOURS AFTER MEDICATION TAKEN</th>
<th>VAR1</th>
<th>VAR2</th>
<th>VAR3</th>
<th>VAR4</th>
<th>VAR5</th>
<th>VAR6</th>
<th>VAR7</th>
<th>VAR8</th>
<th>VAR9</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROB</td>
<td>-0.3912</td>
<td>-0.3912</td>
<td>-0.3912</td>
<td>-0.3912</td>
<td>-0.3912</td>
<td>-0.3912</td>
<td>-0.3912</td>
<td>-0.3912</td>
<td>-0.3912</td>
</tr>
<tr>
<td>SUM</td>
<td>0.3912</td>
<td>0.3912</td>
<td>0.3912</td>
<td>0.3912</td>
<td>0.3912</td>
<td>0.3912</td>
<td>0.3912</td>
<td>0.3912</td>
<td>0.3912</td>
</tr>
</tbody>
</table>

Figure 3.