INTRODUCTION

Statistical efficacy reports of clinical trials usually contain descriptive statistics for each drug studied and p-values. Descriptive statistics (n's means, medians, etc.) are readily output into datasets by SAS procedures such as MEANS and UNIVARIATE. P-values, which are used to determine the statistical significance of clinical results, may be listed in a print file without the option of outputing the results into a dataset. In these cases, the programmer must extract the p-value from the SAS procedure output. The process of extraction includes redirecting the procedure output to an external file and reading the required values into a SAS dataset.

ANALYSIS DATASET

Raw efficacy datasets in clinical trials are typically structured with each observation containing evaluations of multiple parameters for one subject at one timepoint. In an allergy drug trial, for example, an observation might contain an investigator number, a subject number, a visit number, and symptom scores for nasal congestion, sneezing, and other symptoms. A separate dataset contains each investigator number, subject number, and the drug code.

The raw efficacy dataset is merged with the drug code and then transposed to get one observation for each parameter per timepoint. A numeric code (par) identifies the parameter and a numeric variable (value) contains the evaluation. The contents of a simplified analysis dataset are as follows.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Length</th>
<th>Format</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investig</td>
<td>Char</td>
<td>2</td>
<td>$2.</td>
<td>Investig</td>
</tr>
<tr>
<td>Subject</td>
<td>Char</td>
<td>3</td>
<td>$3.</td>
<td>Subject</td>
</tr>
<tr>
<td>Par</td>
<td>Num</td>
<td>3</td>
<td>Z3.</td>
<td>Par</td>
</tr>
<tr>
<td>Visit</td>
<td>Num</td>
<td>3</td>
<td>Z3.</td>
<td>Visit</td>
</tr>
<tr>
<td>Value</td>
<td>Num</td>
<td>8</td>
<td>Best8.</td>
<td>Value</td>
</tr>
<tr>
<td>Drug</td>
<td>Char</td>
<td>1</td>
<td>$1.</td>
<td>Drug</td>
</tr>
</tbody>
</table>

Note that the variable labels are the same as the variable names. Although the raw datasets contain more meaningful labels, the analysis dataset labels should be simple because the labels appear in the SAS procedure output which will be extracted. The formats are also important for extraction. Numeric variables which are used as by-variables in the statistical procedure must have uniform lengths in the output. The correct format then is Zw, where w is the variable length. Character variables are similarly formatted as $w.

EXTRACTING P-VALUES FROM PROC GLM

Study Specifications:
- Multicenter
- Subjects randomized to receive 1 of 3 drugs (A, B, or C)

P-values requested:
1. Drug
2. Investigator by Drug Interaction
3. Pairwise Comparisons
   - A vs. B
   - A vs. C
   - B vs. C

The SAS code used for this analysis is:

```
PROC GLM DATA=ANALY;
BY PAR VISIT;
CLASS INVESTIG DRUG;
MODEL VALUE = INVESTIG DRUG
             INVESTIG*DRUG / SS3;
LSMEANS DRUG / STDERR PDIFF;
```

The object of the extraction process is to output a dataset containing the by-variables and the requested p-values. This dataset may then be merged with the dataset containing the descriptive statistics to produce a table summarizing the efficacy results. In order to extract the information, the format of the output must be known.

NESUG '92 Proceedings
The by-variables appear at the top of each page centered within a set of hyphens:

------------- Par=001 Visit=001 -------------

The first page of output for each by-variable contains information on class variables (Investig and Drug in this case). None of this information needs to be extracted. The second page of output for each by-variable contains statistics on the dependent variable (Value), including the overall Drug and interaction p-values. The third page contains the least squares means statistics and a matrix of pairwise comparison p-values. The sections containing the five p-values we will extract appear as follows.

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Type III SS</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investig</td>
<td>4</td>
<td>5.62078394</td>
<td>1.40S19598</td>
<td>0.58</td>
<td>0.6749</td>
</tr>
<tr>
<td>Drug</td>
<td>2</td>
<td>14.39296402</td>
<td>7.19648201</td>
<td>2.99</td>
<td>0.0538</td>
</tr>
<tr>
<td>Investig*Drug</td>
<td>8</td>
<td>14.87794637</td>
<td>1.85974330</td>
<td>0.77</td>
<td>0.6274</td>
</tr>
</tbody>
</table>

The following SAS code creates a dataset containing the by-variables and the p-values.

*** PROC PRINTTO with no options returns the SAS procedure output to the default destination (print file);
*** PROC PRINTTO;
*** DATA STAT;
*** end in the procedure output. The pad option
*** appends blanks to the end of each record. The
*** missover option prevents SAS from going to a new
*** line when expected variables are missing;
*** FILEOUT OUTSTAT PAD MISSOVER;

*** Input the entire record as one variable and hold the
*** input record using the trailing @;
*** INPUT @1 TEMP SCHAR132. @;
*** Remove case-sensitivity;
*** TEMP = UPCASE(TEMP);

*** Retain the by-variables;
*** RETAIN PAR VISIT;

*** Use the INDEX function to search each record for
*** specific strings and return the beginning column;
*** Read the by-variables;
*** COL = INDEX (TEMP, 'PAR=');
*** IF COL GT 0 THEN
*** PAR = INPUT( SUBSTR( TEMP,COL+4, 3), 3. );
*** COL = INDEX (TEMP, 'VISIT=');
*** IF COL GT 0 THEN
*** VISIT = INPUT( SUBSTR( TEMP,COL+6, 3), 3. );

*** Read the drug and interaction p-values;
*** COL = INDEX (TEMP, 'TYPE III SS');
*** Skip the current line and next 2 lines, read in dummy
*** character variables, and the numeric p-values;
*** IF COL GT 0 THEN
*** INPUT /// D S D $ D $ D S D $ PDRUG
*** / D S D $ D $ D $ D S PINT;

*** Retain these p-values for later input statements;
*** RETAIN PDRUG PINT;

*** Read the pairwise comparison p-values;
*** COL = INDEX (TEMP, 'LEAST SQUARES MEANS');
*** IF COL GT 0 THEN DO;
*** INPUT /// D S D $ D S D S D S D $ P12 P13
*** / D S D $ D S D S D $ D S P23;

*** Set retained variables to missing;
*** PAR=.; VISIT=.; PDRUG=.; PINT=.;
*** END;
*** KEEP PAR VISIT PDRUG PINT P12 P13 P23;
*** RUN;
A SIMPLER METHOD

Prior to release 6.06 of the SAS system, almost all useful p-values had to be read using extraction techniques. Recent enhancements to SAS have eliminated the need to extract some information from procedure output. The following SAS code outputs the desired p-values into a SAS dataset.

```sas
PROC GLM DATA=ANALY OUTSTAT=OUTSTAT;
  BY PAR VISIT;
  CLASS INVESTIG DRUG;
  MODEL VALUE = INVESTIG DRUG
               INVESTIG*DRUG / SS3;
  CONTRAST 'I VS 2' DRUG I -1 0;
  CONTRAST 'I VS 3' DRUG I 0 -1;
  CONTRAST '2 VS 3' DRUG 0 1 -1;
RUN;
```

The OUTSTAT option of the GLM procedure in SAS release 6.06 produces an output dataset containing, among other variables, the by-variables and p-values. The CONTRAST statement is used to get pairwise comparison p-values into the output dataset. These p-values are the same as obtained from the matrix produced by the LSMEANS statement. Printing dataset work.outstat would look like the following.

```
Par  VISIT  PDRUG  PINT  P12  P13  P23
1 1  0.0538 0.6274 0.0195 0.0932 0.4724
```

The following SAS code transposes the dataset to be identical to the dataset created by the extraction method.

```sas
DATA STAT;
  SET OUTSTAT(DROP=_NAME_ DF SS F);
  WHERE _TYPE_='SS3' OR _TYPE_='CONTRAST';
  BY PAR VISIT;
  RETAIN PDRUG PINT P12 P13 P23;
  *** Initialize to missing;
  ARRAY _PVAL_I*/ PDRUG PINT P12 P13 P23;
  IF FIRST.VISIT THEN DO I=1 TO DIM(_PVAL_);
    _PVAL_I[I] = .;
  END;
  IF _TYPE_='SS3' THEN DO;
    IF _SOURCE_='DRUG' THEN PDRUG=PROB;
    ELSE IF _SOURCE_='INVESTIG*DRUG' THEN PINT=PROB;
  END;
  ELSE DO;
    IF _SOURCE_='1 VS 2' THEN P12=PROB;
    ELSE IF _SOURCE_='1 VS 3' THEN P13=PROB;
    ELSE P23=PROB;
  END;
  IF LAST.VISIT THEN OUTPUT;
  KEEP PAR VISIT PDRUG PINT P12 P13 P23;
RUN;
```

In the example above, extraction programming is not required to output the desired p-values. However, there are many situations in which we need the ability to read p-values and other information from procedure output. In the previous example, if we were requested to flag comparisons significant at the 0.05 level using Dunnett’s test, extraction of the flag would be necessary. Next, the extraction technique described above will be used to read p-values from other SAS procedures.

PROC LIFETEST

The LIFETEST procedure is used when the analysis is based on time to an event, with censoring.

P-values requested:
1. Log-Rank
2. Wilcoxon

The by-variables are read in the same way as in the PROC GLM example. PROC LIFETEST prints product-limit survival estimates and quantiles for each drug, a summary of censored and uncensored values, and statistics over strata. The p-values are listed in the section appearing as follows.

```
Par Visit _Name_ _Source_ _Type_ DF SS F Prob
001 001 Value Error Error 128 308.07 . 0.6749
001 001 Value Investig SS3 4 5.62 0.5838 0.6749
001 001 Value Drug SS3 2 14.39 2.9900 0.0538
001 001 Value Investig*Drug SS3 8 14.88 0.7727 0.6274
001 001 Value 1 VS 2 Contrast 1 13.47 5.5964 0.0195
001 001 Value 1 VS 3 Contrast 1 6.89 2.8606 0.0932
001 001 Value 2 VS 3 Contrast 1 1.25 0.5194 0.4724
```

The following SAS code transposes the dataset to be identical to the dataset created by the extraction method.
The following code reads the by-variables and requested p-values into a SAS dataset. The variable CENVAR contains a 1 for censored values, or 0 otherwise.

FILENAME OUTSTAT 'OUTSTAT.DAT';
PROC PRINTTO PRINT=OUTSTAT NEW;
RUN;

PROC LIFETEST DATA=ANALY;
BY PAR;
TIME VALUE*CENVAR(1);
STRATA DRUG;
RUN;

PROC PRINTTO;
RUN;

DATA STAT;
INFILE OUTSTAT PAD MISSOVER;
INPUT @1 TEMP $CHAR132. @;
TEMP = UPCASE(TEMP);
*** Retain the by-variables;
RETAIN PAR;

*** Read the by-variables;
COL = INDEX (TEMP, 'PAR=');
IF COL GT 0 THEN
PAR = INPUT( SUBSTR( TEMP,COL+4, 3), 3. );

*** Read the p-values;
COL=INDEX(TEMP, 'EQUALITY OVER STRATA');
IF COL GT 0 THEN DO;
*** Skip current line and next 4 lines, read dummy
*** character variables and p-values;
INPUT ///// D $ D $ D $ D $ PLOGRANK
/D $ D $ D $ D $ PWILC;
OUTPUT;
*** Set retained variables to missing;
PAR=.;
END;
KEEP PAR PLOGRANK PWILC;
RUN;

Printing work.stat would look like the following.

PAR PLOGRANK PWILC
15 0.0196 0.0425

PROC NPARIWAY

For some nonparametric analyses, the NPARIWAY procedure is used to obtain the Kruskal-Wallis p-value (in the case of more than 2 drugs) or the Wilcoxon rank-sum p-value for the 2-sample case, including pairwise comparisons. Using the analysis dataset described in the GLM example, we will read the Kruskal-Wallis p-value and then subset the dataset to read the 2-sample pairwise comparison p-values from the normal approximation. The SAS code for this analysis follows.

*** Kruskal-Wallis test;
PROC NPARIWAY WILCOXON DATA=ANALY;
BY PAR VISIT;
CLASS DRUG;
VAR VALUE;
RUN;

*** For Pairwise comparisons, we run NPARIWAY 3
*** more times with drug subsetting;
PROC NPARIWAY WILCOXON DATA=ANALY;
WHERE DRUG='A' OR DRUG='B';
BY PAR VISIT;
CLASS DRUG;
VAR VALUE;
RUN;

*** Repeat for AC and BC subsets;

The Kruskal-Wallis p-value is listed as follows.

Kruskal Wallis Test (Chi-Square Approximation)
CHISQ= 5.3939 DF= 2 Prob > CHISQ= 0.0674

The 2-Sample tests contain additional information as
shown below for the subset of drugs A and B.

Wilcoxon 2-Sample Test (Normal Approximation)
(with Continuity Correction of .5)
S= 2632.00 Z= 2.32572 Prob > |Z| = 0.0200

T-Test approx. Significance = 0.0221

Kruskal Wallis Test (Chi-Square Approximation)
CHISQ= 5.4253 DF= 1 Prob > CHISQ= 0.0198
To avoid typing the same code four times with slight modifications, one macro is called 4 times.

FILENAME OUTSTAT 'OUTSTAT.DAT';
%MACRO NPARP(SUBSET=, PVAR=);
PROC PRINTTO PRINT=OUTSTAT NEW;
RUN;
*** Kruskal-Wallis Test (more than 2 drugs);
%IF %INDEX(&PVAR, KW) GT 0
%THEN %LET STRING=PROB > CHISQ=;
*** 2-Sample test -- use normal approximation;
ELSE %LET STRING=PROB > |Z| =;
%LET LENSTR=%LENGTH(&STRING);
PROC NPARIWAY WILCOXON DATA=ANALY;
%IF &SUBSET NE
%THEN %STR(WHERE &SUBSET);
BY PAR VISIT;
CLASS DRUG;
VAR VALUE;
RUN;
PROC PRINTTO;
RUN;
DATA &PVAR;
INFILE OUTSTAT PAD MISSOVER;
INPUT @1 TEMP SCHAR132. @;
TEMP=UPCASE(TEMP);
RETAIN PAR VISIT;
COL = INDEX (TEMP, 'PAR=');
IF COL GT 0 THEN
PAR = INPUT( SUBSTR( TEMP, COL+4, 3), 3. );
COL = INDEX (TEMP, 'VISIT=');
IF COL GT 0 THEN
VISIT = INPUT( SUBSTR( TEMP, COL+6, 3), 3. );
COL = INDEX(TEMP, "&STRING");
IF COL GT 0 THEN DO;
INPUT @COL+&LENSTR &PVAR;
OUTPUT;
PAR=.; VISIT=.;
END;
KEEP PAR &PVAR;
RUN;
%MEND NPARP;

%MNPAP(PVAR=PKW);

%MNPAP(PVAR=P12,
SUBSET=DRUG='A' OR DRUG='B');
%MNPAP(PVAR=P13,
SUBSET=DRUG='A' OR DRUG='C');

Merging the four datasets by PAR and VISIT would look like the following.

<table>
<thead>
<tr>
<th>PAR</th>
<th>VISIT</th>
<th>PKW</th>
<th>PRS12</th>
<th>PRS13</th>
<th>PRS23</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.0674</td>
<td>0.0200</td>
<td>0.1322</td>
<td>0.5943</td>
</tr>
</tbody>
</table>

CONCLUSION

The following steps are used to extract p-values:

1. FILENAME statement assigns external file for output.
2. Redirect procedure output using PROC PRINTTO with the options NEW and PRINT=.
3. Run the SAS procedure.
4. PROC PRINTTO with no options redirects output back to the default destination.
5. Create a dataset by reading the external file with an INFILE statement referencing the external file.
6. INPUT statement with MISSOVER option reads each line as a variable and holds line with the trailing @.
7. UPCASE function removes case sensitivity.
8. RETAIN the by-variables.
9. Use the INDEX function to read by-variables and search for a unique string to locate and input p-values.
10. Output an observation and then set all retained variables to missing; keep by-variables and p-values.

Extracting information is dangerous. The format of the output and the attributes of the variables going into the procedure should be clearly understood. The programs have to be validated and then revalidated with each new release of the SAS system because the format of procedure output often changes with the enhancements of a new release. Extraction programs should only be used to read information for which there are no SAS options or statements available to output them.

Recent releases of the SAS system have eliminated the need to extract some p-values. The table on the next page summarizes the methods of reading some useful p-values in SAS releases 5.18, 6.06 and 6.07.

SAS® is a registered trademark of SAS Institute Inc., Cary, NC, USA.
### P-VALUES COMMONLY REPORTED IN CLINICAL RESEARCH

<table>
<thead>
<tr>
<th>SAS® Procedure</th>
<th>P-Value</th>
<th>Version 5.18</th>
<th>Version 6.06</th>
<th>Version 6.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM</td>
<td>SS1, SS2, SS3, SS4, Contrast</td>
<td>Extract</td>
<td>OUTSTAT option of PROC GLM</td>
<td>OUTSTAT option of PROC GLM</td>
</tr>
<tr>
<td>FREQ</td>
<td>Chi-Square</td>
<td>Extract</td>
<td>Extract</td>
<td>OUTPUT* statement</td>
</tr>
<tr>
<td></td>
<td>CMH General Association</td>
<td>Extract</td>
<td>Extract</td>
<td>OUTPUT* statement</td>
</tr>
<tr>
<td></td>
<td>Fisher's Exact Test</td>
<td>Extract</td>
<td>Extract</td>
<td>OUTPUT* statement</td>
</tr>
<tr>
<td>Lifetest</td>
<td>Log-Rank</td>
<td>Extract</td>
<td>Extract</td>
<td>Extract</td>
</tr>
<tr>
<td></td>
<td>Wilcoxon</td>
<td>Extract</td>
<td>Extract</td>
<td>Extract</td>
</tr>
<tr>
<td>Npar1way</td>
<td>Wilcoxon</td>
<td>Extract</td>
<td>Extract</td>
<td>Extract</td>
</tr>
<tr>
<td></td>
<td>Rank-Sum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kruskal-Wallis</td>
<td>Extract</td>
<td>Extract</td>
<td>Extract</td>
</tr>
<tr>
<td>Univariate</td>
<td>Wilcoxon</td>
<td>OUTPUT</td>
<td>OUTPUT statement</td>
<td>OUTPUT Statement</td>
</tr>
<tr>
<td></td>
<td>Signed-Rank</td>
<td>statement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Means</td>
<td>Paired-t</td>
<td>OUTPUT</td>
<td>OUTPUT Statement</td>
<td>OUTPUT Statement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>statement</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*See SAS® Technical Report P-222