Paper AD14

Statistical Table Specifications and Automatic Code Generation using XML

Alan Hopkins and Linda Collins, PharmaStat LLC, Moraga, CA

ABSTRACT

We present a general table model for display of statistical results from a typical clinical trial and describe an application to generate table descriptions in XML. This XML file is then used as input to a SAS code generating application which creates a SAS driver program built with validated macros. The table descriptions and associated software may be saved as standards for future studies. Using an XML-structured approach to defining tables and validated software for producing the tables results in much quicker validation and higher quality of statistical tables produced for clinical study reports.

BACKGROUND

Good statistical practice dictates that a statistical analysis plan (SAP) should be prepared for clinical studies (ICH E6, E9; references 3 and 4). The SAP describes the statistical methods planned in more detail than in the study protocol. In addition, the SAP will usually describe the statistical data displays. The descriptions are usually in the form of “mock tables” which focus on the layout of the rows and columns of each table but do not actually contain data. Mock tables are reasonable visual representations, but are tedious to create (usually in a word processor) and there is no standard format for describing the details of the data table such as the study population, the statistics to display, etc. There is no way to use the mock table as “metadata” for automation of other processes like program creation.

We have built a statistical table designer which creates an XML representation of a statistical table format. This information is used to create SAS code for invoking standard analysis software. Our designer is built to create XML file which describes input to the APT Software Library (reference 1). Using APT Software - a well-defined, well-validated ‘tool’ library of SAS macros - can help satisfy the pressure to produce reliable, customized reports in minimum time.

A macro library separates the complex, generic components of processing from the ones that are specific to a particular analysis. The application program, then, is short and to the point. The code in the application program contains only the details that are relevant to that analysis. The simplicity of the reporting system makes automation of the entire process attractive.

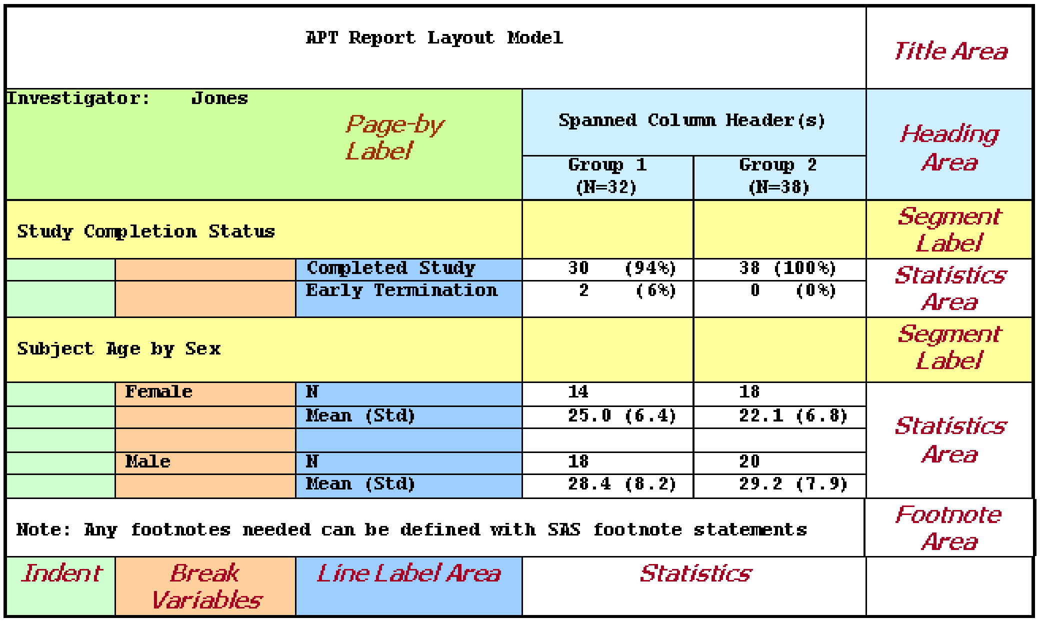
Validating and documenting an application program using the macro library is correspondingly short and to the point. The validation and documentation need only deal with the analysis-specific details.

The complex details of processing are handled by the macro library. These details are described – and proven correct – in one place. The macro documentation and validation is provided with the macro library and applies to all analyses done under that version of the macro library.

THE SEGMENT CONCEPT: BUILDING A TABLE IN SECTIONS

Consider a system that allows the user to build a statistical table in ‘segments’. Each ‘segment’ corresponds to a variable to be analyzed. Characteristics of a statistical segment include the statistics to generate, how the statistics look, whether they are broken out by one or more categories, and many other options. Once a statistical segment is defined, its characteristics become the default for the rest of the analysis variables for that report. Or, if needed, these characteristics can be different for different analysis variables. The concept is displayed graphically in Figure 1.

Figure 1. Schematic layout of a typical statistical table



APT Report Layout Model					Title Area
Investigator: Jones Page-by Label			Spanned Column Header(s)		Heading Area
			Group 1 (N=32)	Group 2 (N=38)	
Study Completion Status					Segment Label
		Completed Study	30 (94%)	38 (100%)	Statistics Area
		Early Termination	2 (6%)	0 (0%)	
Subject Age by Sex					Segment Label
	Female	N	14	18	Statistics Area
		Mean (Std)	25.0 (6.4)	22.1 (6.8)	
	Male	N	18	20	
		Mean (Std)	28.4 (8.2)	29.2 (7.9)	
Note: Any footnotes needed can be defined with SAS footnote statements					Footnote Area
Indent	Break Variables	Line Label Area	Statistics		

There are two segments in this table. The first is study completion status. These statistics displayed in this segment were created by a single APT macro call for simple subject counts. The second segment displays age by sex. A break-out variable (sex) is used to stratify the age statistics which is generated by the APT descriptive statistics macro. The whole display could also be stratified using a page-by variable which creates the whole display for different values of the page-by variable.

There is a broad range of possible layouts for clinical reports. No macro library can possibly produce all of them. However, the statistical summaries generally used fall into a fairly small number of general types. The types of analysis used in most reports include:

- Numeric univariate statistics
- Categorical summations
- Counts of the number of unique subjects in certain categories such as adverse events
- Comparative statistics: p-values, differences in means or proportions, confidence intervals for differences
- Time-to-Event (survival) analysis

Within these general report types, a few options can provide a surprising degree of flexibility. It's useful to be able to control characteristics such as:

- By-variable categories
- Control of the format of statistics
- Sort order for categories
- Arrangement of statistics and categories on the page

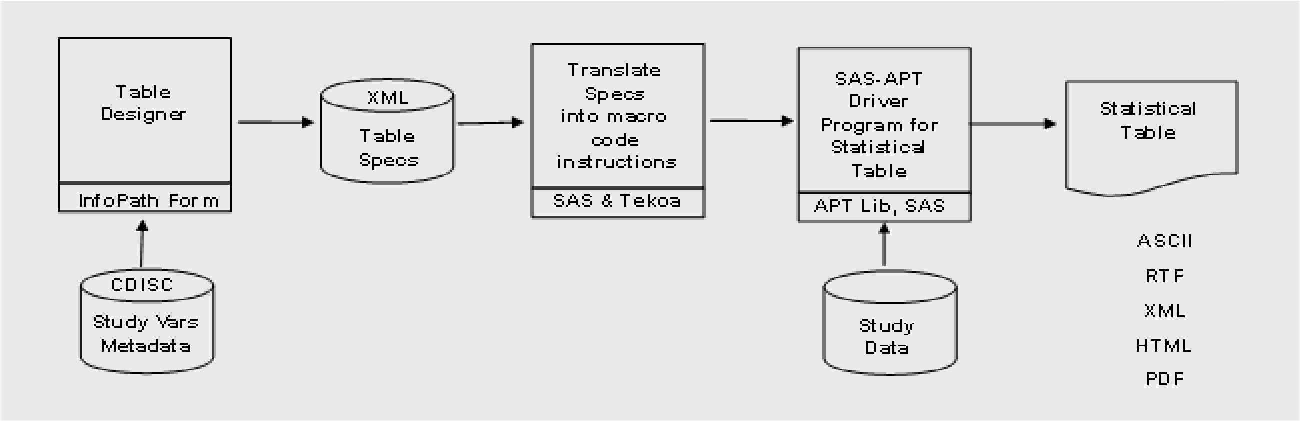
For each table, certain information is of a general nature such as labeling for titles, columns, footnotes, data set name, analysis population, etc. Each segment also needs to be defined:
Segment name
Analysis variable name
Analysis type (continuous, categorical, subject counts, and survival)
Then depending on the analysis type, specific questions about statistics to be reported are necessary.

The user sets up the driver program with as many statistics 'segment' macro calls as needed. After that, a final macro call prints the report to a text file. The output report can be ASCII, RTF, XML, PDF or any other destination supported by SAS ODS.

TABLE CREATION PROCESS OVERVIEW

The process we used for creating statistical tables is illustrated in Figure 2 below. Our statistical table designer program is a graphical form for capturing all the information associated with a statistical table. We used Microsoft InfoPath™ to develop this application. InfoPath is an application that helps create forms and saves the information entered into the form in an XML file. The table designer has a schema that defines all the tags and associated values. InfoPath allows creation of a form with data field validation just as any data entry program would do. Drop down menus of variable names associated with standards can easily be implemented. Pick lists can be driven from a database maintained outside the system. After the table characteristics are specified in the designer form, the data are saved in an XML file.

Figure 2. Process for creating statistical tables

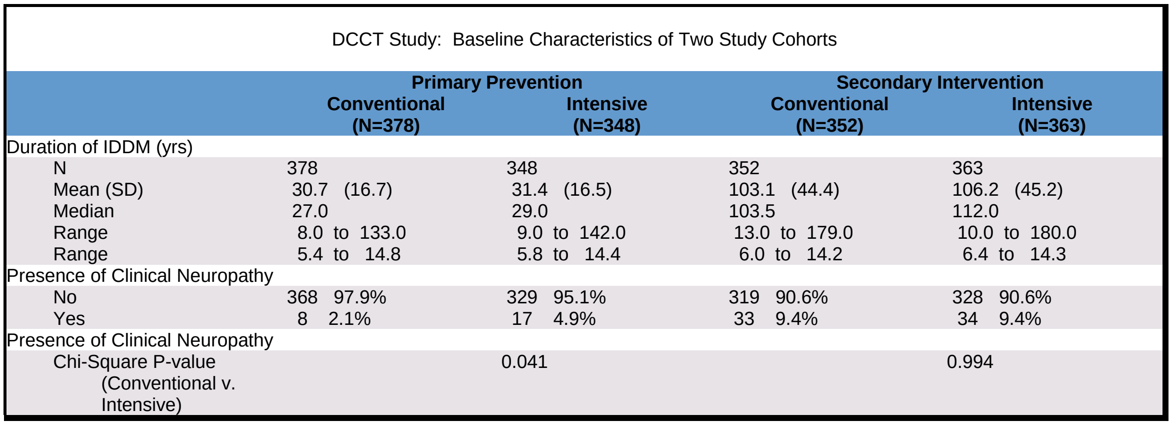


The XML generated in the table designer is necessarily hierarchical. For example each table will likely have multiple segments. We used Zurich Biostatistics' Tekoa™ Technology software to process the hierarchical XML into a normalized indexed SAS data set of tags and associated values. We process the XML tags from the normalized file with a SAS program which creates calls to the APT macro system to produce a driver program which can create the statistical table when run against the appropriate clinical data set.

An Example

We used the table designer to create a statistical table using baseline data from the Diabetes Control and Complications Trial (DCCT, 1993). Subjects in two study cohorts were randomized to either conventional or intensive insulin therapy. The table (Figure 3) we created was a subset of the baseline data table presented in the original paper. This table has three segments. The first segment specifies basic descriptive statistics for the duration of insulin-dependent diabetes. The second segment is frequency of clinical neuropathy. The third segment displays p-values for comparing conventional therapy with intensive therapy for each of the two cohorts – the primary prevention cohort and also the secondary intervention cohort.

Figure 3. DCCT Statistical Table – Selected baseline characteristics



DCCT Study: Baseline Characteristics of Two Study Cohorts

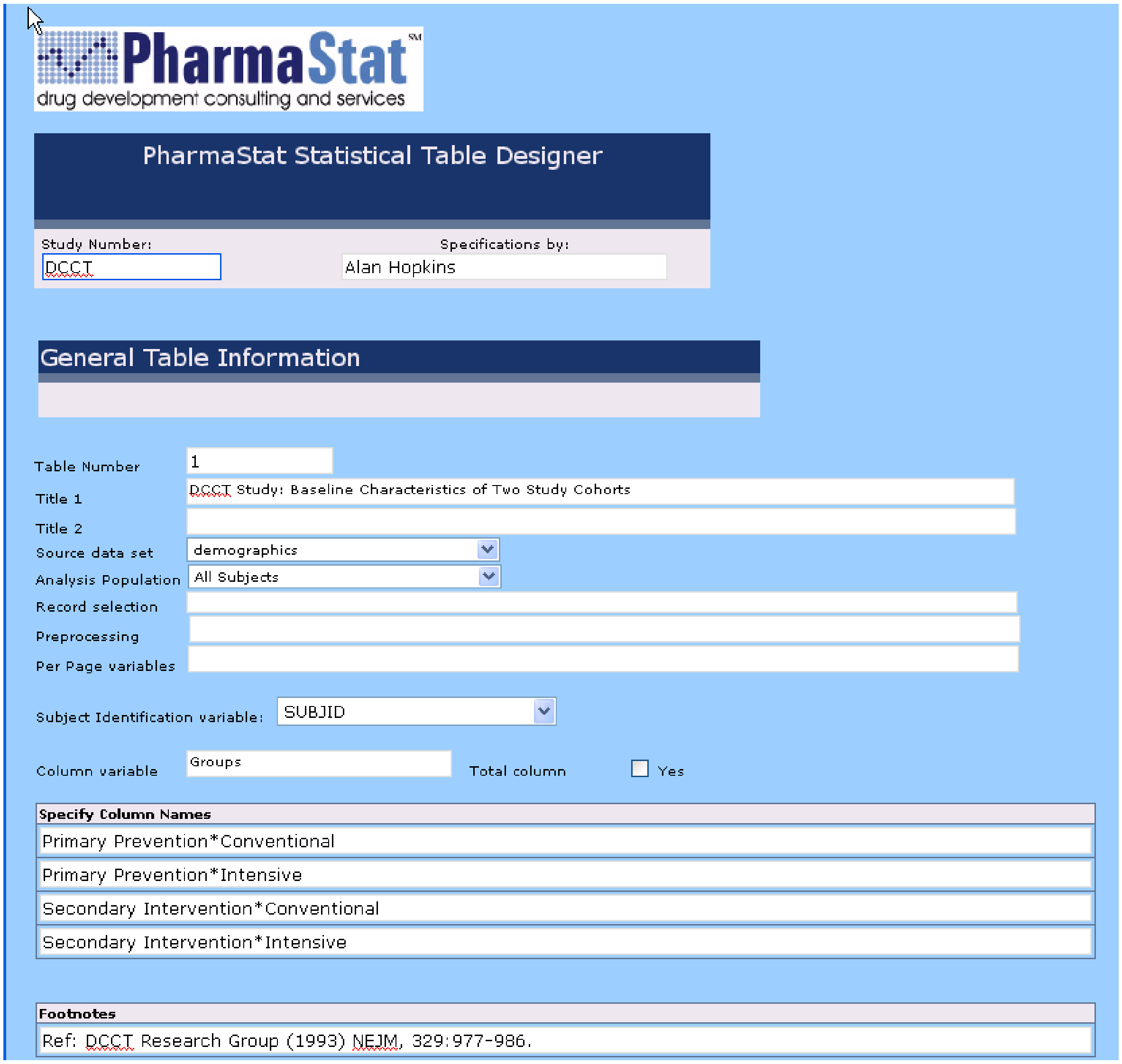
	Primary Prevention		Secondary Intervention	
	Conventional (N=378)	Intensive (N=348)	Conventional (N=352)	Intensive (N=363)
Duration of IDDM (yrs)				
N	378	348	352	363
Mean (SD)	30.7 (16.7)	31.4 (16.5)	103.1 (44.4)	106.2 (45.2)
Median	27.0	29.0	103.5	112.0
Range	8.0 to 133.0	9.0 to 142.0	13.0 to 179.0	10.0 to 180.0
Range	5.4 to 14.8	5.8 to 14.4	6.0 to 14.2	6.4 to 14.3
Presence of Clinical Neuropathy				
No	368 97.9%	329 95.1%	319 90.6%	328 90.6%
Yes	8 2.1%	17 4.9%	33 9.4%	34 9.4%
Presence of Clinical Neuropathy				
Chi-Square P-value (Conventional v. Intensive)		0.041		0.994

Table Designer

A portion of the table designer form for the DCCT table is shown in the Figure 4 below. This part of the form captures general information about the table. What is the data set name? What variable designates a column variable? The Designer asks to user to select data from drop-down pick lists which contain CDISC standard compliant variable names. In this way, the designer helps enforce organizational data standards.

Certain information on the form is required and it is evident on the blank form by red asterisks in the required field. Examples of required fields are the source dataset and the subject identification variable. Other information is optional such as preprocessing code (e.g. SAS where clauses) or page-break variables.

Figure 4. Statistical table designer – general information



PharmaStat Statistical Table Designer

Study Number:	DCCT
Specifications by:	Alan Hopkins

General Table Information

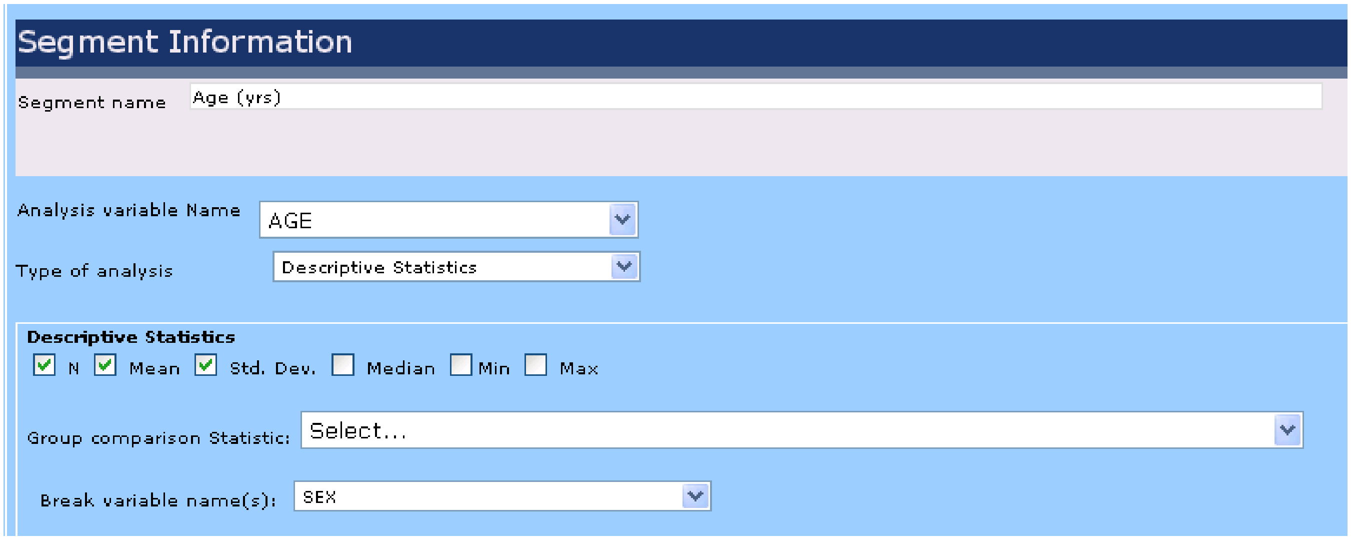
Table Number	1
Title 1	DCCT Study: Baseline Characteristics of Two Study Cohorts
Title 2	
Source data set	demographics
Analysis Population	All Subjects
Record selection	
Preprocessing	
Per Page variables	
Subject Identification variable:	SUBJID
Column variable	Groups
Total column	☐ Yes

Specify Column Names
Primary Prevention*Conventional
Primary Prevention*Intensive
Secondary Intervention*Conventional
Secondary Intervention*Intensive

Footnotes
Ref: DCCT Research Group (1993) NEJM, 329:977-986.

After the general information is complete, the user will specify the details of the segments of the table. There may be any number of segments. Figure 5 shows the segment-specific information required for the first segment in the DCCT example table. Each segment has a name, an analysis type, a variable name and a “break variable” to identify any SAS by-variable processing. In this example we have chosen to present descriptive statistics for AGE. We have indicated that the statistics will be stratified by SEX and that we will not calculate statistics for group comparisons. This is all that is needed to identify the first segment in the table. Segments two and three would be created in a similar way but using different options for the type of analysis.

Figure 5. Statistical table designer – segment specification



Segment Information

Segment name: Age (yrs)

Analysis variable Name: AGE

Type of analysis: Descriptive Statistics

Descriptive Statistics

☑ N ☑ Mean ☑ Std. Dev. □ Median □ Min □ Max

Group comparison Statistic: Select...

Break variable name(s): SEX

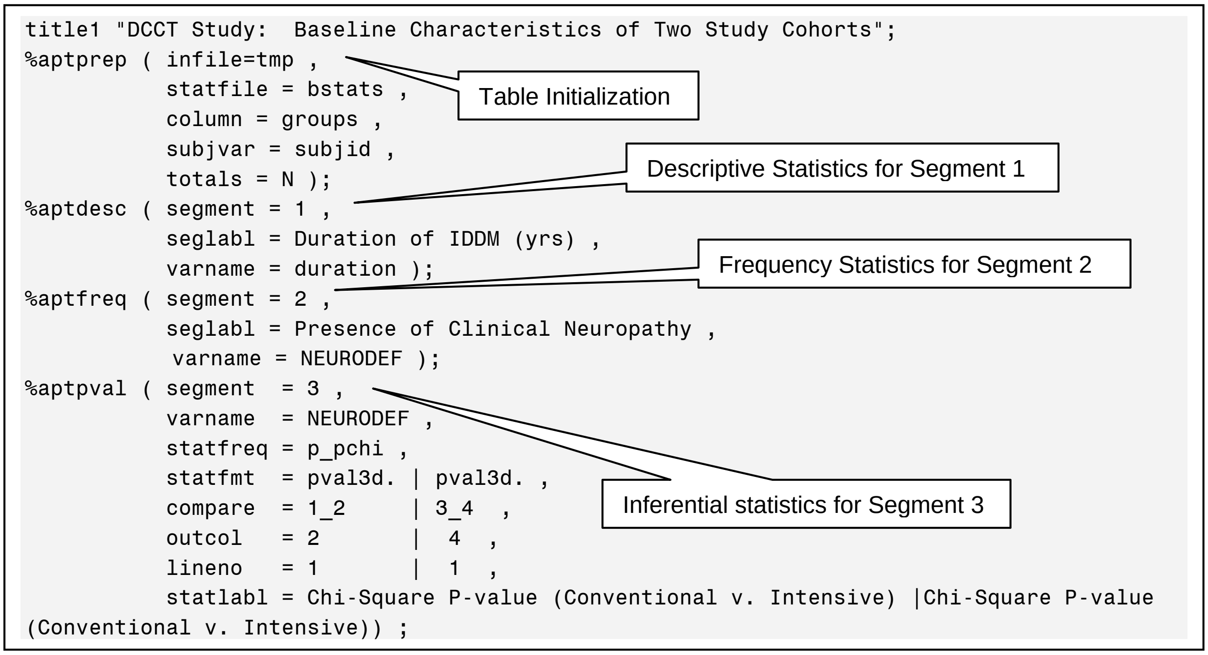
In addition to multiple segments, the user may specify multiple tables in a single session. After the tables are described, the user will save the specification file which is an XML file. In the Appendix we display what the XML output from the designer program looks like.

USING XML TO CREATE SAS ANALYSIS PROGRAMS

The XML created using the table designer is hierarchical. There is the general table information and the information on each segment which is repeated. We used freely available SAS macros (Palmer, 2002) for converting the XML created in the table designer to indexed SAS datasets (Palmer, 2002). The XML keywords can be used to create APT software macro calls for table reporting software to create each segment. The type of analysis is defined by the segment type (descriptive, frequency, subject counts, survival or p-values). The remaining tags within the segment are options for the specific display type selected. This is all straightforward because there is only a single macro call for each segment.

The code used to create the table in Figure 3 from the specifications in the XML file from the table designer is shown in the Figure 6 below.

Figure 6. Computer generated code to create Figure 3.



```
title1 "DCCT Study:  Baseline Characteristics of Two Study Cohorts";
%aptprep ( infile=tmp ,
           statfile = bstats ,
           column = groups ,
           subjvar = subjid ,
           totals = N );
%aptdesc ( segment = 1 ,
           seglabl = Duration of IDDM (yrs) ,
           varname = duration );
%aptfreq ( segment = 2 ,
           seglabl = Presence of Clinical Neuropathy ,
            varname = NEURODEF );
%aptpval ( segment  = 3 ,
           varname  = NEURODEF ,
           statfreq = p_pchi ,
           statfmt  = pval3d. | pval3d. ,
           compare  = 1_2     | 3_4   ,
           outcol   = 2       |  4   ,
           lineno   = 1       |  1   ,
           statlabl = Chi-Square P-value (Conventional v. Intensive) |Chi-Square P-value
(Conventional v. Intensive)) ;
```

Table Initialization

Descriptive Statistics for Segment 1

Frequency Statistics for Segment 2

Inferential statistics for Segment 3

SUMMARY

A table model and a comprehensive macro library for creating statistical tables is a prerequisite for automating the table making process. The table model allows one to create an analysis specification for each part of the table. Given the analysis specification, the driver programs can be created from the table specifications. Using a validated application to specify the table setup and validated macros to create and print the table statistics, much time can be saved in validating the final tables. Collectively, these tools allow the user to focus on the content of the table rather than the mechanics of programming.

The XML-based table descriptions are reusable if the underlying data structures are the same between trials. A clear specification for a table promotes good communication between statistician and programmer. The XML-based specifications could be included as part of an electronic statistical analysis plan. We would like to extend our tools set to create mock table displays which are familiar to members of the clinical review team.

Use of standard software for table generation is an example of defined business processes which, when combined with good statistical methodology can help ensure credibility of results.

REFERENCES

APT Software Library™ (2004). Copyright © Hygia Biostat Inc., Oakland, CA.

Diabetes Control and Complications Trial Research Group. (1993) The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. *New England Journal of Medicine* 329:977-986.

FDA Guidance for Industry. E3 Guideline for Industry: Structure and Content of Clinical Study Reports, July 1996. www.fda.gov/cder/guidance/iche3.pdf

FDA Guidance for Industry: E9 Statistical Principles for Clinical Trials, September 1998. http://www.fda.gov/cder/guidance/ICH_E9-fnl.PDF

Palmer, Michael. (2002) Tekoa Technology(SM). Copyright © Zurich Biostatistics Inc., Morristown, New Jersey.

ACKNOWLEDGMENTS

We are indebted to Michael Palmer at Zurich Biostatistics, Inc. for making the Tekoa Technology software available to us. Michael is also responsible for raising awareness of the power of XML for generating statistical tables through a series of papers he has presented at PharmaSUG in the past.

Data from the Diabetes Control and Complications Trial was obtained from the University of Minnesota General Clinical Research Center.

CONTACT INFORMATION

We are actively developing tools to automate various analytical processes. Contact:

Alan Hopkins, Ph.D.
PharmaStat LLC
Phone: 925-631-9400
Email: ahopkins@pharmastat.com
Web: www.pharmastat.com

APPENDIX: Partial XML file from the table designer.

```
<?xml version="1.0" encoding="UTF-8"?>
<my:Designer>
        <my:GeneralInfo>
                <my:StudyID>DCCT</my:StudyID>
                <my:Author>Alan Hopkins</my:Author>
                <my:Table_Number>1</my:Table_Number>
                <my:Title1>DCCT Study: Baseline Characteristics of Two Study Cohorts</my:Title1>
                <my:Title2></my:Title2>
                <my:cName></my:cName>
                <my:Population>All Subjects</my:Population>
                <my:columnvar>Groups</my:columnvar>
                <my:DSName>demographics</my:DSName>
                <my:PerPageVars></my:PerPageVars>
                <my:PreProcessing></my:PreProcessing>
                <my:TotalCol>true</my:TotalCol>
                <my:Num_Cols></my:Num_Cols>
                <my:WhereClause></my:WhereClause>
                <my:SubjectIDVar>SUBJID</my:SubjectIDVar>
                <my:ColumnNames>
                        <my:SpecifyColNames>Primary Prevention*Conventional</my:SpecifyColNames>
                      <my:SpecifyColNames>Primary Prevention*Intensive</my:SpecifyColNames>
                      <my:SpecifyColNames>Secondary Intervention*Conventional</my:SpecifyColNames>
                       <my:SpecifyColNames>Secondary Intervention*Intensive</my:SpecifyColNames>
               </my:ColumnNames>
                <my:Footnotes>
                        <my:FNote>Ref: DCCT Research Group (1993)  NEJM, 329:977-986.</my:FNote>
                </my:Footnotes>
        </my:GeneralInfo>
        <my:Segment>
                <my:SegmentName>Duration of IDDM (yrs) </my:SegmentName>
                <my:Analysis_Type>Descriptive Statistics</my:Analysis_Type>
                <my:varname>AGE</my:varname>
                <my:BreakVar></my:BreakVar>
                <my:DescriptiveStats>
                        <my:Count>true</my:Count>
                        <my:Mean>true</my:Mean>
                        <my:StandardDeviation>true</my:StandardDeviation>
                        <my:Median>false</my:Median>
                        <my:Min>false</my:Min>
                        <my:Max>false</my:Max>
                        <my:dstat></my:dstat>
                        <my:dbreak>SEX</my:dbreak>
                </my:DescriptiveStats>
        </my:Segment>
       <my:Segment>
               <my:SegmentName>Presence of Clinical Neuropathy</my:SegmentName>
                <my:Analysis_Type>Frequency Analysis</my:Analysis_Type>
                <my:varname></my:varname>
                <my:BreakVar></my:BreakVar>
                <my:FreqStats>
                        <my:TotalN>true</my:TotalN>
                        <my:fpercent>true</my:fpercent>
                        <my:ftotal>false</my:ftotal>
                        <my:fstats></my:fstats>
                        <my:SortOrder></my:SortOrder>
                        <my:brkvar></my:brkvar>
                        <my:Denom></my:Denom>
                        <my:PrintMissing></my:PrintMissing>
                </my:FreqStats>
        </my:Segment>
         ...
</my:Designer>
```