Paper SER11_05

Use a SAS Database Search Engine Instead of Writing a Program

Eugene Yeh, PharmaNet Inc., Cary, NC
Donovan Verrill, PharmaNet, Inc., Cary, NC

ABSTRACT

We developed a search engine to scan all the variables from multiple SAS® data sets located in different directories. It produces a summary report of data values and names of data structure elements (e.g. variable names and labels) from multiple data sets that match your search parameter. The report can hyperlink to a subset of the source data sets, consisting only of observations that match your search criteria. You can also restrict the names of variables and data sets scanned and therefore it can filter data just like the SAS Viewer.

This paper describes the use of the SAS database search engine in performing different tasks. These tasks involve scanning many data locations, where the approach of using a search engine is more practical and efficient than writing a SAS program and analyzing its results. A wide range of personnel can use this application since programming knowledge is not required.

INTRODUCTION

A search engine application was created to fill a void in performance left by the current SAS and Windows operating system (OS) query tools. This search engine is capable of scanning all the data values and the database structure from multiple SAS data sets located in multiple directories (1). The GUI (graphic user interface) is an Excel spreadsheet file with point and click functionality (see Figure 1). The user defined search parameter and all the settings are entered in the Excel file. Initiated by clicking on 'Start Search', a SAS program conducts the actual search. The search engine is suitable for use by a wide range of personnel since no programming knowledge is required. Advanced search features allow you to refine your search results. A single search scans either numeric data values, or character values, including the names of data structure elements (e.g. variable names and labels). The standard output produced from conducting a search consists of several HTML files, which displays a table of contents that hyperlinks to a set of tables. Each table displays either a listing of matches found to the search parameter, or a summary of the count of matches. A description of each table is shown in Table 1. These tables provide the basic information about the search results, such as the most common value that matched the search parameter, and which data sets and variables held the most matches.

The advantages of using the search engine include the following capabilities:

1. Easy to use, point and click interface, programming skills are not required
2. Scans all observations and all variables from multiple data sets located in multiple directories with a single search
3. Identify matches to data values and names of data structure elements (e.g. variable name & label, data set name & label etc.) from multiple SAS data sets
4. Recognizes unformatted and formatted data values
5. Produce summary tables and listings describing all matches
6. Save search parameters and all settings to be reused at a later time

When is it practical to use a SAS database search engine, instead of other methods of reviewing data, such as writing a SAS program, or filtering a data set with FSEDIT or the SAS Viewer? The following scenarios will be discussed in which the search engine was used:

1. Scenario #1: The topic of interest has a character value, stored in many different locations. The data values are stored in unformatted and formatted variables, from multiple data sets, and directories

2. Scenario #2: The topic of the data of interest is known, but its location(s) in the database are unknown. The name of the relevant data set(s) and variable(s) are unknown.

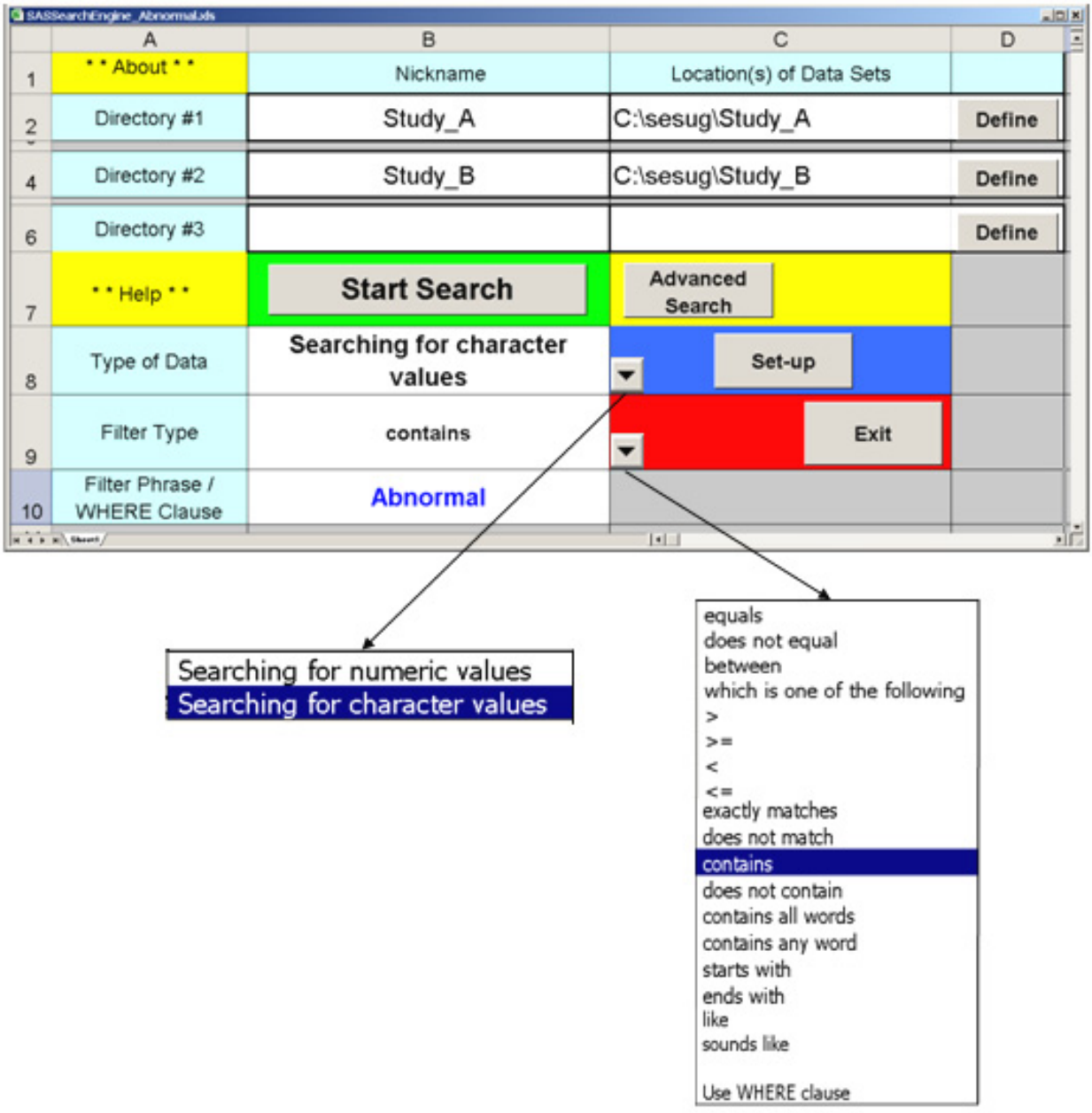


Figure 1 – Main Menu of SAS Database Search Engine. Search is conducted on pathnames defined in cells C2, C4 and C6. Search parameter is defined in cells B8, B9 and B10.

3. Scenario #3: Analyze a single numeric variable from one data set. Mimic the functionality of filtering, and subsetting data using FSVIEW, an interactive SAS session or SAS Viewer.

	Table Topic	Type of Table	Describes Matches to . . .
1	Summary of Matches in Data Structure	Summary of Counts	Data Structure Elements
2	List of Matches in Data Structure	Listing	Data Structure Elements
3	Summary of Matching Directories	Summary of Counts	Data Values
4	Summary of Matching Directories and Data Sets	Summary of Counts	Data Values
5	Summary of Matching Variables	Summary of Counts	Data Values
6	Summary of Matching Directories, Data Sets and Variables	Summary of Counts	Data Values
7	Summary of Matching Data Values, Data Sets	Summary of Counts	Data Values
8	Summary of Matching Data Values, Variables, Data Sets	Summary of Counts	Data Values
9	*Summary of SUBJECT Values	Summary of Counts	Data Values
10	*Summary of SUBJECT Values, Data Sets	Summary of Counts	Data Values
11	*Summary of SUBJECT Values, Variables, Data Sets	Summary of Counts	Data Values
12	**List of Directories, Data Sets, Variables, SUBJECT and Values	Listing	Data Values

Table 1 – List of Standard Search Engine Output Tables

*Table is produced only when a reference variable (e.g. SUBJECT) is chosen by the user.
**If a reference variable (e.g. SUBJECT) is not chosen by the user, the observation number (i.e. row number) is displayed instead.

SYSTEM

This application is running Microsoft® Windows® 2000, Win2000 servers, Microsoft V5.00 Terminal Server Client, Microsoft® Office® 2000 and SAS version 8.2 in production. can appear as a false positive match to any character search because of a match to a format name, and not to any formatted value.

SCENARIO #1

You are given two sets of data sets representing information collected from two clinical studies. The task is to identify all the data values containing the word “abnormal”.

Table of Contents

Study_A: C:\sesug\Study_A
Study_B: C:\sesug\Study_B
Summary of Data Sets with the condition:
contains "abnormal"

4.

Directory Name	Data Set Name	Data Set Label	Total Matches	Cumulative Matches
STUDY_B	PE	Derived Physical Exam data	207	207
STUDY_A	AE	Derived Adverse Events Data	11	218

6.

Directory Name	Data Set Name	Variable Name	Variable Label	Total Matches	Cumulative Matches
STUDY_B	PE	PEABNCD, fmt=PNNORA.	Physical exam status at Visit 1	205	205
STUDY_A	AE	MDPREF	MedDRA Preferred Term	7	212
STUDY_A	AE	AETXT	Adverse experience	4	216
STUDY_B	PE	PECOM	Physical exam description	2	218

8.

Data Value	Variable Name	Variable Label	Data Set Name	Directory Name	Total Matches	Cumulative Matches
2, fmt=**Abnormal**	PEABNCD, fmt=PNNORA.	Physical exam status at Visit 1	PE	STUDY_B	205	205
ABNORMAL PAP SMEAR PRECANCEROUS CELLS PRESENT	AETXT	Adverse experience	AE	STUDY_A	1	206
ABNORMAL SENSATION UP TO THE MIDDLE 1/3RD OF LEGS BI-LATERAL AND UP TO WRIST BI-LATERAL	PECOM	Physical exam description	PE	STUDY_B	1	207

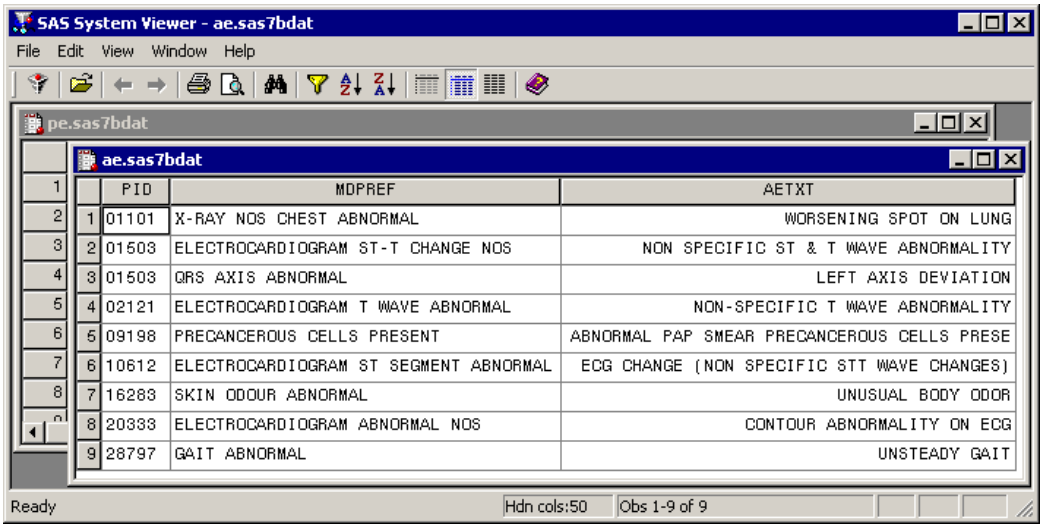
9.

PID	Directory Name	Total Matches	Cumulative Matches
28798	STUDY_B	6	6
29405	STUDY_B	6	12
17295	STUDY_B	5	17
12233	STUDY_B	4	21

12.

Directory Name	Data Set Name	Variable Name	Variable Label	PID	Data Value	Total Matches	Cumulative Matches
STUDY_B	PE	PEABNCD, fmt=PNNORA.	Physical exam status at Visit 1	28798	2, fmt=**Abnormal**	6	6
STUDY_B	PE	PEABNCD, fmt=PNNORA.	Physical exam status at Visit 1	29405	2, fmt=**Abnormal**	6	12
STUDY_B	PE	PEABNCD, fmt=PNNORA.	Physical exam status at Visit 1	17295	2, fmt=**Abnormal**	5	17

Figure 2a – Search results from Scenario #1, all data values containing "abnormal". Table of contents link to standard output tables (see corresponding numbers). Note: some tables created are not completely displayed or are not shown.



	PID	MDPREF	AETXT
1	01101	X-RAY NOS CHEST ABNORMAL	WORSENING SPOT ON LUNG
2	01503	ELECTROCARDIOGRAM ST-T CHANGE NOS	NON SPECIFIC ST & T WAVE ABNORMALITY
3	01503	QRS AXIS ABNORMAL	LEFT AXIS DEVIATION
4	02121	ELECTROCARDIOGRAM T WAVE ABNORMAL	NON-SPECIFIC T WAVE ABNORMALITY
5	09198	PRECANCEROUS CELLS PRESENT	ABNORMAL PAP SMEAR PRECANCEROUS CELLS PRESE
6	10612	ELECTROCARDIOGRAM ST SEGMENT ABNORMAL	ECG CHANGE (NON SPECIFIC STT WAVE CHANGES)
7	16283	SKIN ODOUR ABNORMAL	UNUSUAL BODY ODOR
8	20333	ELECTROCARDIOGRAM ABNORMAL NOS	CONTOUR ABNORMALITY ON ECG
9	28797	GAIT ABNORMAL	UNSTEADY GAIT

Figure 2b – Subset of all source data sets containing only observations that match the search parameter "abnormal" created and displayed after search is completed

Study_A: C:\sesug\Study_A
Study_B: C:\sesug\Study_B
Summary of Data Sets with the condition:
contains "death"
Summary of Matches found in Data Structure

Data Structure Element	Total Matches
Data set Name	1
Variable Name	6
Variable Label	4
Formatted Value	2

List of Matches found in Data Structure

Data Structure Element	Directory Name	Data set Name	Variable Name	Value
Data set Name	STUDY_B	DEATH		**DEATH**
Variable Name	STUDY_B	DEATH	DEATHDT	**DEATH**DT
			DEATHDY	**DEATH**DY
			DEATHMO	**DEATH**MO
Variable Label	STUDY_B	DEATH	DEATHDT	Date of subject's **death**
			DEATHDY	(day part) Date of subject's **death**
Formatted Value	STUDY_A	AE	AEOUTCD, fmt=AEOUT.	4, fmt=**Death**
		DEMO	EOSPRCD, fmt=EOSPR.	4, fmt=**Death**

Figure 3 - Search results from Scenario #2, all data structure elements containing "death". Summary table shown on the left side, and listing (incomplete) shown on the right side. Clicking on data set name (blue underlined text) hyperlinks to the source data sets

Using the search engine, the directory pathname of the two studies were entered into cells C2 and C4, see Figure 1. The corresponding "Nickname" of these two studies is entered into cells A2 and A4 and will appear in the summary output files as an alias for the pathnames. A character search "containing" the word "abnormal" was requested by defining the values in cells B8, B9, and B10. In the advanced search menu (not shown), a reference variable PID, the subject number, is defined. Also from this menu, we requested that a subset of each data set with at least one match be created. These data sets contain only observations with at least one occurrence of our search parameter, "abnormal". These data sets will be automatically displayed by the SAS Viewer when the search is completed.

The standard set of summary tables as outlined in Table 1 is created upon completion of the search. Of all these tables the most pertinent ones are shown in Figures 2a. This figure displays the number of matches per data set per directory, and the number of matches per variable, per data set, per directory. The variable PEABNCD from data set PE, has the majority of matches, 205 out of 218.

A SAS Viewer display is shown in Figure 2b. Two data sets, AE and PE are shown as the subsets of the original data sets, each containing at least one match to "abnormal" in each observation. The AE data set has nine observations and displays seven matches for variable MDPREF and four matches for variable AETXT. Note that this summary count of matches is the same as that shown in table 6 in Figure 2a. The next table in Figure 2a reveals that PEABNCD is a numeric variable, where an unformatted value of 2 corresponds to a formatted value of "Abnormal" (format name is PNNORA.). This is in the first row since it has the greatest number of matches. The next table shows the subject number (variable PID), which has the greatest number of matches; regardless of the variable or data set in which it occurred. The bottom table of Figure 2a lists the number of matches per PID, variable name, data set name and directory. If a match were to occur in a data set which lacked the PID variable, the observation number would be displayed instead of the value of PID.

ALTERNATIVE METHODS

If you use FSEDIT, an interactive SAS session, to perform this same task, you would have to open each of the data sets from the two directories and apply the same data filter, contains 'abnormal', to each variable, one variable at a time. Furthermore, numeric data, whose formatted value contains 'abnormal' is tedious to identify. You can sort the data set by each formatted numeric variable, one variable of the time. Then, all similar formatted values for the sorted variable are adjacent to each other and the 'abnormal' can be identified by browsing the data set. Another method to use is to apply the format name to the numeric variable it's assigned to, and determine which formatted values contain "abnormal".

The SAS Viewer handles character variables more easily than the FSEDIT or a SAS session, since using the FIND feature scans all the variables for a given data set, not just one variable. Unfortunately, if you wish to scan the entire database, you need to scan each data set one at a time. Also, if any matches are present, the SAS Viewer will move the cursor from the location of one match to the next match, but not provide you a summary of all matches.

Another approach is to use the "search" feature in Microsoft Explorer to scan all the data sets in each directory and the format catalog for files "containing" the word "abnormal". A list of data sets containing "abnormal" is created by the search. The search results yield matches to data sets AE and PE and the format catalog. The third table in Figure 2a reveals that most of the matches come from the formatted value of PEABNCD, while the data set PE only has two matches from unformatted values. The formatted value of PEABNCD is why the format catalog is part of the Explorer search results. But using Explorer does not reveal which variables and data sets use this formatted value. Explorer identified data sets AE and PE as matches based only on matches to character values, not on the format labels associated with numeric values. To determine the latter, you must do more work by manually matching the format name using that formatted value with the format name assigned to data set variable(s).

SCENARIO #2

You are given two sets of data sets representing information collected from two clinical studies. The case report forms used for these studies are unavailable. The task is to identify the location(s) where "death" is collected in each study. That is, which data structure elements (e.g. Data set, variable names, labels,formatted values) contain the word "death".

A character search containing the word “death” was requested by defining the values in cells B8, B9, and B10. The “Start Search” button is clicked to initiate the search.

The two summary tables produced from the search are shown in Figure 3. Study_A has no matching data set or variable names or labels, but two variables have formatted values which contain “death”. That is for both variables, the unformatted value is 4, but applying the format attribute associated with variables AEOUTCD and EOSPRCD, the formatted value is “Death”. For Study_B, one data set name, six variable names and four variable labels contain the word “death” (in lower, upper and mixed case). Figure 3 (right side) displays some of these data set, variable and label names.

ALTERNATIVE METHODS

If you use FSEDIT, an interactive SAS session, or SAS Viewer to perform this same task, you would have to open each of the data sets from the two directories and scan through the data set name and all variables and their labels for the word “death”. A summary of all matches is unavailable. Although you can view formatted data values in a SAS session, you cannot scan through all the possible formatted values defined in the data structure.

You can also approach this task by writing a short SAS program to print out the entire data structure from both studies and all formatted values (ie. format labels).

```
Proc Contents data=Study_A._all_;
Proc Contents data=Study_B._all_;
Proc Formats library=library fmtlib;
run;
```

Writing and running this program may take very little time, but analyzing its output may take much longer. To find all the occurrences of “death”, you must manually scan the output from this program, but unfortunately, this method also does not provide a summary of all the matches. Also if a formatted value contains “death”, you will need to scan through all variables from all the data sets using that format, to identify all the variables using a formatted value of “death”.

Using Microsoft Explorer to scan for the word “death” in all the data sets and format catalog is similar to scanning for “abnormal” in scenario #1. A list of data sets containing “death” is created by the search, but Explorer does not identify any variables responsible for a match. If any formatted value in the format catalog contains “death”, the entire catalog is treated as a match. To find the variables using the formatted values, you must manually match the format name using a formatted value with the format name assigned to data set variables. In general, the format catalog can appear as a false positive match to any character search because of a match to a format name, and not to any formatted value.

SCENARIO #3

Validation of tables and listings can be a time-consuming, frustrating process. In many cases the best practical way to find out where discrepancies lie is for the programmer of the table and the validation programmer to print out lists or counts for same individual cell within the table. If multiple cells are not matching, numerous lists must be made and compared.

There is a vital signs table that you are to validate (see Figure 4). You count 6 subjects in treatment 2 (labeled XYZ in Figure 4) with diastolic blood pressure greater than 100, where as the table shows 5 such occurrences. In such a case, both you and the table programmer can use the search engine to list these occurrences. Defining a numeric search with the “Where Clause” set to “bpdia >=100 and trt=2” can be requested by setting the values in cells B8, B9, and B10 (see Figure 1). In the advanced search menu, a reference variable PID, the subject number, is defined. Also from this menu, we request that the search be limited to only specific variables, BPDIA and TRT, from one data set. The single data set scanned can first be that one used for validation and then the search engine can be rerun scanning that data set used by the table programmer. The resulting output (see Figure 4) lists all subjects meeting the criteria. The table of contents in Figure 4 indicates five unique PID values meeting the search parameter. Table 12 in Figure 4 lists all the PID values and their

ABC Pharmaceuticals
Protocol: XYX-001

Table 1
Summary of Blood Pressure
Subjects with Diastolic Blood Pressure >= 100 or Systolic Blood Pressure >=200

Test	Statistic	Placebo	XYZ
Diastolic Blood Pressure	N	3	5
	Mean	107.3	100.4
	Min, Max	100, 120	100, 102
Systolic Blood Pressure	N	2	1
	Mean	223.0	210
	Min, Max	216, 230	210, 210

Table of Contents

12.

Study_A: C:\sesug\Study_A
Study_B: C:\sesug\Study_B
Summary of Data Sets with the condition:
WHERE bpdia >=100 and trt=2
Includes variable: PID

Directory Name	Data Set Name	Variable Name	Variable Label	PID	Data Value	Total Matches	Cumulative Matches
STUDY_A	VITALS	BPDIA	Diastolic Blood Pressure in mmHg	07173	100	1	1
STUDY_A	VITALS	BPDIA	Diastolic Blood Pressure in mmHg	14657	100	1	2
STUDY_A	VITALS	BPDIA	Diastolic Blood Pressure in mmHg	15271	100	1	3
STUDY_A	VITALS	BPDIA	Diastolic Blood Pressure in mmHg	23757	100	1	4
STUDY_A	VITALS	BPDIA	Diastolic Blood Pressure in mmHg	27390	102	1	5

Figure 4 – Scenario #3: Search engine used to list occurrences (bottom) for table validation (top). Note column header "XYZ" (top) corresponds to "trt=2" (bottom).

corresponding BPDIA values. A comparison can be made between the lists generated from a search of both data sets.

This technique of scanning a specific data set instead of the entire database would also be useful when validating patient listings. Granted, when only making a comparison for one cell, it may be just as easy to do as Proc Print with a Where clause within your program. However, when comparisons must be made for multiple cells, the search engine becomes incredibly useful. The values in the Excel file completely define the search parameter. The Excel file can be renamed and saved as part of documenting the validation process. The search engine can also generate a subset of the source data set VITALS, containing only the observations that match the search parameter. This data set can also be kept as part of documenting the validation process.

ALTERNATIVE METHODS

You can create a new SAS program in order to create these lists. To list all occurrences for a given table cell, a typical short program would look like this:

```
Proc Print Data=mylib.mytable;
   Where (Condition X...);
   Var PID trt bpdia;
Run;
```

"Condition X" (see above) changes value for each cell in the table. Filtering a single data set can also be performed by using FSEDIT, SAS Viewer or an interactive session of SAS. But since the filtering operation cannot be permanently saved, documenting the use of these methods is more difficult than writing and saving a program, or using the search engine.

Using the "search" feature in Microsoft Explorer to scan observations in any data set that conform to a specific numeric data condition is impossible, since Explorer does not recognize numeric SAS data.

DISCUSSION

Query features like FIND and WHERE exist in SAS applications like Display manager, FSVIEW procedure, FSEDIT procedure and SAS Viewer. However the starting point for using most features relies on you explicitly choosing specific data sets and variables by name. This requirement is impractical when many data sets and/or variables are involved or their names are unknown. The FIND feature is capable of scanning for matches to a search parameter across all variables of a given data set, jumping from the location of one match to the next match. The disadvantage of this tool is that it does not produce a summary of all the matches within the data set, and must be re-initiated if multiple data sets are scanned.

The Windows OS has a file search feature in Windows Explorer, which allows you to identify SAS data sets containing a given text (character) value. The ASCII text found within a SAS data set contains the ASCII value for each character variable in each observation, as well as elements of the data structure. Therefore, character matches can occur based on text values in the data set labels, variable names, labels, and format names. However, this method has several disadvantages. ASCII text such as "SAS", "DATA", "FILE" and "WIN" are contained in every data set and is prone to cause false positive matches. You can only enter "Containing text" when specifying a search parameter, and can not enter more sophisticated terms like "contains any word", "not containing all words" etc. Windows Explorer cannot search for numeric SAS data values or formatted values (i.e. format labels). The summary of matches shown for character value searches reveals only the matching data set name, not the variable name, the observation number or the number of matches per data set. This search engine application fills a void in performance left by the current SAS and Windows OS query tools.

CONCLUSION

The search engine application is an efficient tool to help guide you to specific areas of interest within many SAS data sets. Specifically, the application is useful when you do not know or are unsure of which explicit data sets and variables to query, or matches to your search parameter exist in many different locations. The latter case makes this application suitable for use in the topics of exploratory analysis and identification of data discrepancies (e.g. invalid dates, misspellings, extreme values etc), where matches to a search parameter can exist anywhere within the SAS database.

Knowledge of the database structure is not required to use the search engine. Without a search engine, significant time is required to learn the structure of all data sets, perform queries, or write SAS programs explicitly using variable names.

This search engine application is capable of scanning formatted values (i.e. format labels) as well as unformatted values. This allows you to find matches to "unknown", for example, without having to know that "unknown" is stored in any data set as a coded numeric value of -9, 9 or 99.

The search engine offers an easy way for you to express the search parameter from a drop-down menu with phrases such as "Contains all words", "Contains any word", "Does not contain", etc. The popularity of internet-based search engines has introduced the use of a search engine to the public, making the learning curve for using any search engine virtually nonexistent. This search engine application functions in the same basic manner as any internet search engine, as outlined in the following steps:

1. Determine your topic of interest
2. Enter it in as a search parameter
3. Click on "Start Search"
4. View a summary of the search results
5. Click on any item within the summary to view the particular item matching your search parameter

If needed, repeat Steps 1 to 5, and consider using advance search features to narrow in on the matches of interest.

Search parameters can be saved as new Excel files and later re-run when more or different data is available, or can be re-run on an entirely different set of data sets. For example, search parameters to identify different types of data discrepancies can exist as separate Excel files named: Invaliddates.xls, negativevalues.xls, misspellingA.xls.

Writing some SAS programs may be tedious when many explicit variable names must be given. In addition, analyzing the output from even very short SAS programs, like that one in scenario #2 can be a time consuming task. Therefore for certain tasks, using a search engine instead of writing a SAS program might be the most practical way to analyze your SAS data, especially when it involves many variables, data sets, or formatted data values. A SAS database search engine also allows more people, not just programmers, find relevant data within your SAS database in less time and with less effort.

REFERENCES

(1) Yeh, E. (2004), "A SAS® Database Search Engine Gives Everyone the Power to Know®," *Proceedings of the PharmaSUG 2004 Conference*, Paper AD02.

(2) Gareleck, C., Fain, T. (2003), "Mouse Clicking Your Way Viewing and Manipulating Data with Version 8 and 9 of the SAS® System," *Proceedings of the Twenty-eighth Annual SAS Users Group International Conference*, 28, Paper 53-28.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Eugene Yeh
Work Phone: 919-466-8219
Email: eugeneyeh@hotmail.com

Donovan Verrill
PharmaNet Inc.
1001 Winstead Drive, Suite 505
Cary, NC 27513, Work Phone: 919-465-4683
Email: dverrill@pharmanet.com