

Paper SM12

Using the full power of SAS to resolve Algorithmic MedDRA SMQs

Steve Prust, elderbrook solutions GmbH, Biberach an der Riß, Germany



Introduction

Algorithms, Turing, and "Baby"

SAS architecture

Programming "better"

Algorithmic SMQs

Resolving algorithms

Processing multiple observations concurrently

Process Review

Conclusion

Algorithms, Turing, and "Baby"

A Turing proof:

- If an algorithm can be written to solve a mathematical problem, then a Turing machine can execute that algorithm.

"Baby"

- first electronic stored program computer.
- programming approximates to Assembler code.

Our programming environment today ... is based on the ideas of Turing, von Neumann and other pioneers.



SAS Architecture

1970's computing

- sequential processing
- memory constraints
- slow performance

SAS

- sensible design (for the era)
- heritage has consequences now
 - complex / convoluted code
 - inefficient data storage / access



Making Programming Better

But what is better ?

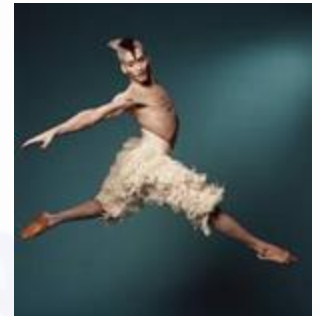
faster / stronger / simple / flexible ???

... you decide

range of tools / options in SAS ... for example

- PROC SQL
- Hash tables
- DSxxx functions ...

The macro in this uses hash tables, an interesting use of the macro RESOLVE function and a powerful macro and SAS user written function technique.



Macro to solve Algorithmic SMQs - 1 - definition

Algorithmic SMQs

- require assessment of a combination of Adverse Events (AEs)
- AEs categorised as A, B, C ...
- some examples ...

SMQ	Definition
Anaphylactic reaction (SMQ)	A or (B and C) or (D and (B or C))
Acute pancreatitis (SMQ)	A or (B and C)
Neuroleptic malignant syndrome (SMQ)	A or (B and C and D)
Systemic lupus erythematosus (SMQ)	A or Sum(Category Term Weight)>6

Macro to solve Algorithmic SMQs - 2 - boolean logic

How to solve a term such as "A or (B and C)"

```
data _null_;  
length condition $14 condition_prep $25;  
input condition 1-14; NOTE: Invalid numeric data,  
condition_prep = %eval('TRUE',trim(condition) || ' '); CONDITION='0 or (0 and 1)'  
if input(resolve(condition_prep),best.) then put 'TRUE';  
else put 'FALSE';  
cards;  
cards;  
0 or (0 and 1)  
0 or (0 and 1) FALSE  
0 or (1 and 1) TRUE  
1 or (0 and 1) TRUE  
1 or (0 and 1)  
;;;;;  
run;  
run;
```

Macro to solve Algorithmic SMQs - 3.1 - Processing AEs

To solve algorithmic SMQs read multiple observations ...

... concurrently

... from several data sources

Algorithm, AEs, MedDRA definitions

Macro variables

→ Algorithm categories

Hash tables are good ... provided data is small

→ MedDRA definitions

AE data ... (initially) ... use BY group processing

Macro to solve Algorithmic SMQs - 3.2 - Processing AEs

use Hash fns to populate arrays with PT codes - one per Algorithm category (A,B,C..)

FOR each subject

 initialise a result array, one item per category letter

FOR each AE for this subject

IF the AE code matches item in PT code array **THEN** corresponding category
result → 1

ENDFOR each AE for this subject

 Substitute category results into algorithm and use RESOLVE func

 output with algorithm result

ENDFOR each subject

Macro to solve Algorithmic SMQs - 3.3 - Contemporaneity

Not enough for AEs to have happened ... within a specified time of each other

BY group processing not an option

Solution of user-written SAS function to run an SQL operation ...

...

a function that reads multiple observations, returns a result, without disrupting normal DATA step processing

User written fn. - reading several obs. during a DATA step

```
data example;  
  set sashelp.class;  
  length query $3200;  
  query = "sashelp.c  
    " and age  
  count = GetSQLans(  
run;
```

a function that :

- reads multiple observations,
- returns a result,
- without disrupting normal DATA step processing

Obs	NAME	SEX	AGE	HEIGHT	WEIGHT	COUNT
1	Alfred	M	14	69.0	112.5	2
2	Alice	F	13	56.5	84.0	2
3	Barbara	F	13	65.3	98.0	2
4	Carol	F	14	62.8	102.5	2
5	Henry	M	14	63.5	102.5	2
6	James	M	12	57.3	83.0	3
7	Jane	F	12	59.8	84.5	2
8	Janet	F	15	62.5	112.5	2
9	Jeffrey	M	13	62.5	84.0	1
10	John	M	12	59.0	99.5	3
11	Joyce	F	11	51.3	50.5	1
12	Judy	F	14	64.3	90.0	2
13	Louise	F	12	56.3	77.0	2
14	Mary	F	15	66.5	112.0	2
15	Philip	M	16	72.0	150.0	1
16	Robert	M	12	64.8	128.0	3
17	Ronald	M	15	67.0	133.0	2
18	Thomas	M	11	57.5	85.0	1
19	William	M	15	66.5	112.0	2

Macro to solve Algorithmic SMQs - 3.4 - Processing AEs

To solve algorithmic SMQs read multiple observations ...

... concurrently

... from several data sources

Algorithm, AEs, MedDRA definitions

Macro variables

→ Algorithm categories

Hash tables are good ... provided data is small

→ MedDRA definitions

AE data ... when a "hit" ... function call to find AEs in other categories in time window

Macro to solve Algorithmic SMQs - 3.5 - Processing AEs

AE data ... when a "hit" ... issue function call to find AEs within timeframe

```
query = "<<AE dataset>> (where=(usubjid='\"||trim(usubjid)||\"' and \" ||  
\"('\"||put(<<start>>,date9.)||\"'d<=astdt<='\"||put(<<end>>,date9.)||\"'d) \"  
|| \" or ('\"||put(<<start>>,date9.)||\"'d<=aendt<='\"||  
put(<<end>>,date9.)||\"'d)\"|| \" or (astdt<'\"||put(<<start>>,date9.)||\"'d  
and aendt>'\"||put(<<end>>,date9.)||\"'d )))\" || \"  
and aeptcd in ( \" <<list of terms for the relevant category>> || ) ))\";  
  
count = GetSQLans (query, \"usubjid\");
```

Macro to solve Algorithmic SMQs - 4 - Review

RESOLVE function: algorithm text → boolean processing

Hash table functionality for MedDRA data - worked well - small size

The user-written function : powerful. What can be put in an SQL query can be used in a fn call. (and fn calls can be used in many places)

Macro tailored to current SMQ definitions

Speed

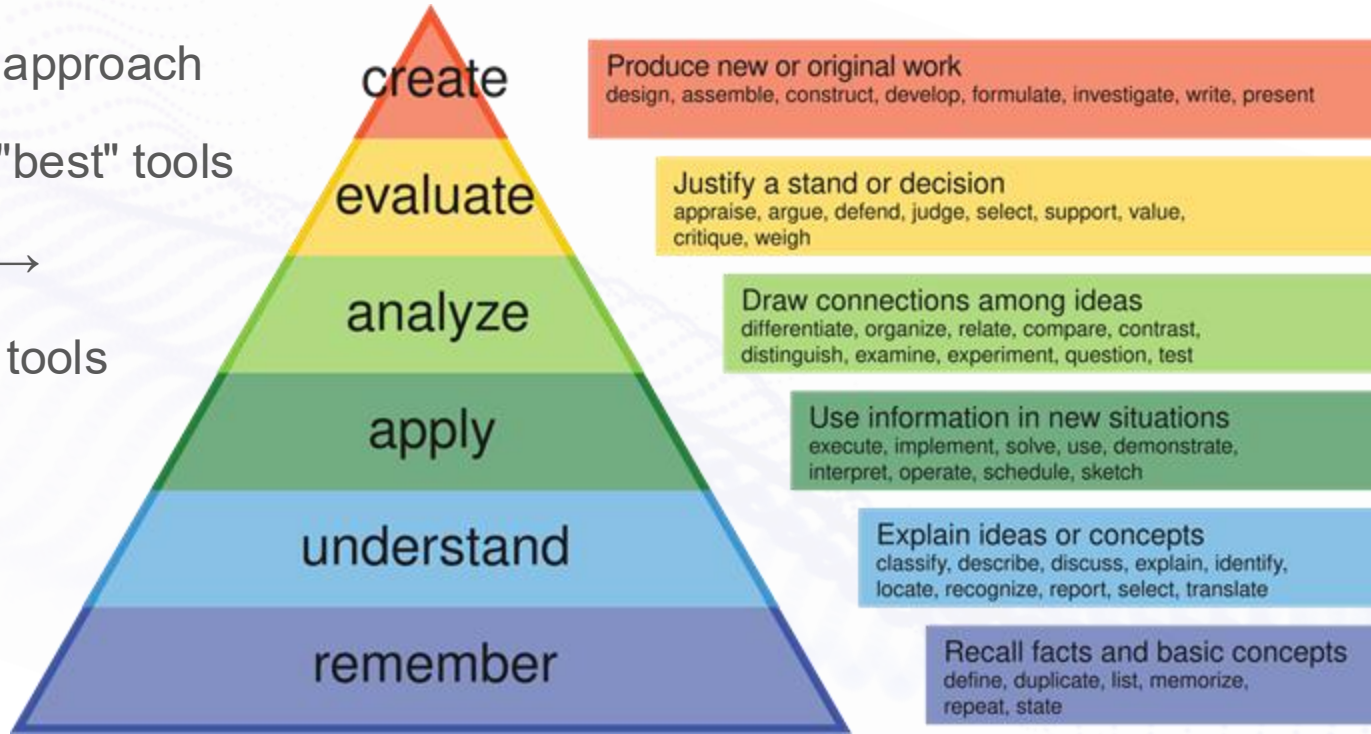
Development process - review

Data driven approach

Finding the "best" tools

Synthesise →

create new tools



Conclusion

Old architectures can do some unexpected things

Difficult problems force us to develop

"The computer is incredibly fast, accurate, and stupid. Man is unbelievably slow, inaccurate, and brilliant. The marriage of the two is a challenge and an opportunity beyond imagination"

Howard Walesh

Finally

Thank you for listening

Questions and comments are encouraged

- ★ elderbrook booth
- ★ stephen.prust@ext.elderbrooksolutions.com