PharmaSUG 2015 - Paper AD03

Proc STREAM: The Perfect Tool For Creating Patient Narratives

Joseph Hinson, inVentiv Health, Princeton, NJ, USA

ABSTRACT

STREAM is a new procedure in SAS® 9.4 which allows free-form text embedded with macro elements to be streamed to an external file. On the surface, one could immediately say: “big deal; I can easily do that with PUT statements, or even via text assignments”. But this nifty tool can do far more -- embedding even macro programs inside text. Furthermore, when the macro-embedded text is processed by the SAS® Word Scanner, the SAS® syntax rules get ignored. The procedure simply streams text directly to an external file, bypassing entirely the SAS® compiler in the process but engaging only the Macro Processor. Thus, Proc STREAM really shines when dealing with text containing such SAS® syntax-violators as HTML or XML tags. Patient narratives also present challenging text processing. Life would have been much easier if each patient had at most one adverse event, took only one concomitant medication, and possessed only a single medical history situation. But in the real world, each patient’s profile would most likely include multiple records of medical history, concomitant medication, and adverse events. With Proc STREAM, macro programs, rather than just macro variables, can be embedded in text to handle multiple data, generate tables, and insert figures, as demonstrated in this paper.

INTRODUCTION

Proc STREAM has much to do about how SAS® processes code sitting in the input stack. Normally, the Word Scanner examines and tokenizes the stream of characters from the input stack. Depending on the type of tokens, the Word Scanner triggers one of the following: the DATA Step Compiler, the Macro Processor, the Command Processor, or the SCL Compiler. The Macro Processor in particular is triggered by ampersand (&) or the percent (%) symbols. Proc STREAM allows various macro specifications to be embedded in arbitrary text such that only the Macro Processor is invoked while the rest of the text gets streamed directly to the external file. Without this selective processing by Proc STREAM, ordinary text containing macro triggers would have macro processing, but the DATA step compiler would flag syntax errors with the remaining text. The non-macro text may not be valid SAS® syntax.

By allowing text to be written as-is (except for macro specifications), it becomes much easier to compose specialized text such as clinical trial patients’ narratives.

As this paper will demonstrate, Proc STREAM allows the kind of versatility and flexibility that permits the inclusion of tables, figures, lists, HTML tags, macros, macro-variables in a patient narrative document.

THE PROCEDURE SYNTAX

a. The Proc STREAM statement specifies an external file with the “OUTFILE=” keyword, as well as options.

b. The arbitrary text is sandwiched inside a “BEGIN” and a four semi-colon ending “;;;;”.

c. There is no “RUN” or “QUIT”.

```
PROC STREAM OUTFILE=fileref <options>;
BEGIN
(some text which may contain macro triggers)
;;;;
```

Proc STREAM statement Options:

a. **RESETDELIM=** ”label”

The SAS® Word Scanner expects macro statements like “%LET” and “%INCLUDE” to begin on a statement boundary -- which means, the statements must be preceded by a semicolon and must end with a semicolon. Therefore to use %LET and %INCLUDE in a Proc STREAM input text, one must place a special marker token before the statements and end the statement with a semicolon. This special marker token is defined with the option RESETDELIM=*label*, where *label* can be any arbitrary SAS® name:

```
PROC  STREAM  OUTFILE= myfile  RESETDELIM=”goto”;

BEGIN

goto; %INCLUDE myotherfile;

;;;;
```

The marker token specified by RESETDELIM is also required when a carriage return is required in the input text. The keyword “NEWLINE” is used with the marker token:

```
PROC STREAM  OUTFILE= myfile  RESETDELIM=”goto”;

BEGIN

Dear Sir, goto NEWLINE;
The profile below is for patient 12345.

;;;;
```

b. **QUOTING=SINGLE** *(or* **DOUBLE** *or* **BOTH***)*

This option specifies that the single quotation mark (‘) should be treated like any other character, as expected for: *the patient’s blood pressure.*

QUOTING=DOUBLE would be required in cases where the text involves, for instance, XML elements in Define-XML documents:

```
<ItemRef ItemOID="ADSL.ARM" OrderNumber="6" Mandatory="No"/>
```

APPLICATION TO PATIENT NARRATIVES:

a. THE SIMPLE CASE

A basic example of Proc STREAM would consist of just simple text containing macro variables:

```
filename aefolder “C:\Users\admin\Desktop\SASoutputs\aenarrative.rtf” ;
PROC  STREAM OUTFILE=aefolder;
BEGIN
Subject &patid. was a &age.-year old &race. &sex. from &country..  The subject consented to partake
in the &studyid. on &randdt. and was first exposed to study medication &trt.. The subject
 was &height. ft in height, &weight. lb in weight and suffered from &disease..
;;;;
```

b. A FULL-BLOWN PATIENT NARRATIVE

A real narrative for a clinical trial subject might involve more than as depicted above. First, the layout might consist of brief lines of demographic data at the top, then free text with various substitutions, followed by a list of items for concomitant medication, medical history, adverse events, then a small table of non-serious adverse events, and lastly a figure image depicting the adverse events profile. To allow the inclusion of such varied output forms, the HTML destination is the most suitable choice. The example for this type would illustrate the flexibility of Proc Stream. Not only does the text contain macro variables but also included are actual macro calls, %INCLUDE statements, and HTML statements. Such versatility allows for the automatic determination of “He” versus “She”, “her” versus “his”, and macro variables that generate a list of items, one item per line, regardless of how varying the number of items are. A macro call can invoke a large code like a table-creating program as well as a tiny HTML carriage return code. The text can contain a line of HTML tags so that a graphical image can be embedded in the output.

The full code below (Code (b)) for a complex patient narrative, can be described in discrete steps:

STEP 1:

Basic macro variables are defined. The HTML carriage return and bullet symbol require specific macro-quoted variables:

```
%let z=%str(<br/>);
```
⟸ for carriage return within HTML;

```
%let y=%nrstr(&#8226);
```
⟸ for the bullet symbol within HTML;

The non-breaking space code in HTML, “ ” also requires macro quoting:

```
%let nbsp=%nrstr( );
```
⟸ to prevent   resolving warning in LOG;

STEP 2:

Adverse Event preferred terms and Concomitant Medication names are stored in macro variables, using the macro variables for HTML carriage return codes and bullet sign codes as separators:

```
proc sql noprint;
select distinct aept into :aetexts separated by "&z.&y."
from patdata
where cat eq "Serious";
quit;

proc sql noprint;
select distinct ConMed into :cmtexts separated by "&z"
from patdata
quit;
```

STEP 3:

A macro, %gp, is created for getting parameters from a lookup table. The Patient Narratives data set, “PATDATA”, is converted into a hash look-up table. First, SAS Data Access functions are used to restructure PATDATA into a new data set, “PATPROFILE”, with just 3 variables: *usubjid, paramname, and paramval.*

```
data patprofile(keep=usubjid paramname paramval);
dsid=open("patdata");
nrows=attrn(dsid,"NOBS");
ncols=attrn(dsid,"NVARS");
do r=1 to nrows;
dc=fetchobs(dsid,r);
do c=2 to ncols;
usubjid=getvarc(dsid,1);
paramname=varname(dsid,c);
test=vartype(dsid,c);
if test="C" then paramval=getvarc(dsid,c);
else if test="N" then paramval=put(getvarn(dsid,c),best.);
output;
end;
end;
dc=close(dsid);
run;
```

PATPROFILE data set is then loaded into a hash table for look-up. The keys for the look-up are USUBJID and PARAMNAME. PARAMNAME refer to the columns of PATDATA: *usubjid, studyid, siteid, trt, age, race, sex, sept, cat, randdt, disease, diagdate,* and *conmed.* The corresponding data values are put in the hash table as PARAMVAL.

```
declare hash pf(dataset:'patprofile');
rc=pf.defineKey('usubjid','paramname');
rc=pf.defineData('paramval');
```

The entire hash code (which also creates a macro variable "&pv") is put into a macro variable as a string:

```
%let codetext=%nrstr(data _null_;
if(1=2) then set patprofile;
declare hash pf(dataset:'patprofile');
rc=pf.defineKey('usubjid','paramname');
rc=pf.defineData('paramval');
rc=pf.defineDone();
rcc=pf.check(key:"&patient.",key:"&pname.");
if rcc eq 0 then do;
rcf=pf.find(key:"&patient.",key:"&pname.");
end;call symputx("pv",paramval,"g");run;);
```

The macro %gp accepts a key "pname" for the hash look-up. The macro also uses macro variable &codetext in the %gp macro definition:
The macro runs the hash code carried by &codetext through the DOSUBL function, and returns the parameter value as &pv:

```
%macro gp(pname);
%let x=%sysfunc(dosubl(&codetext));
&pv.
%mend gp;
```

Thus for example, %gp(race) would retrieve the value for race as "WHITE" via the look-up table and assign it to the macro variable "&pv".

STEP-4:
One hurdle to overcome if one wishes to automate the patient narrative creation is automatically deciding when to use "he", "she", "his", "her" in the generated text.

Two macros, %hsh and %hsr, can be defined to play that role, using the gender value of *SEX*, as shown below:

```
%let vv=%gp(sex);

%macro hsh();
%global heshe;
%if "&vv." eq "Female" %then %let heshe=She;
%else %if "&vv." eq "Male" %then %let heshe=He;
&heshe
%mend hsh;

%macro hsr();
%global hisher;
%if "&vv." eq "Female" %then %let hisher=her;
%else %if "&vv." eq "Male" %then %let hisher=his;
&hisher
%mend hsr;
```

Please note that the macro variables &heshe and &hisher should NOT have a semicolon at the end within the macro code in order to work in Proc Stream.

STEP-5:
ODS Report Writing Interface is used to create a Concomitant Medication table as an HTML file, "conmedtable.html". The file location of conmedtable.html will be provided to Proc Stream.
Also, a graphical plot is saved as an image (in JPEG, for example). This would be coded as an HTML "img src" statement within Proc Stream.

STEP-6 (Final):

The final step is the actual Proc Stream code. The free text embedded in Proc Stream would be made of four components:

a. Individual **%gp** macro calls for individual patient characteristics, such as age and race.

b. A list of macro variables **(&aetexts**) for adverse events.

c. A table presenting concomitant medications (**%include** *tabout*).

d. A graphical image depicting some data trend (<**img src=".....***adverseevents.jpg*"…….>).

The desired Proc STREAM output location is also provided, to be assigned to the option "OUTFILE=".

```
filename product "C:\Users\admin\Desktop\SASoutputs\patnarrative.html";
filename tabout "C:\Users\admin\Desktop\SASoutputs\conmedtable.html";


proc stream outfile=product resetdelim='goto';
BEGIN
Patient No.: &patient. &z.
Treatment Arm: %gp(trt) &z.
Demographics: %gp(age)-year-old %gp(race) %gp(sex) &z.
&z.&z.
This %gp(age)-year-old %gp(race) %gp(sex) was admitted to the
%gp(studyid) clinical trial at the %gp(siteid) study site on %gp(randdt).
&z.
%hsh() was diagnosed with %gp(disease) on %gp(DiagDate) by a team of
oncologists at site %gp(siteid)&z.
%hsh() exhibited, during the course of the trial, the following serious
disorders: &z.&y.&aetexts.. &z.
%hsh() had been taking before and during the study, some non-study
medication as shown in the table below: &z.
goto; %include tabout; &z.
The figure below summarizes all %hsr() non-serious adverse events
(Placebo vs Multivitamins vs NewDrug): &z.
goto; <img src= "C:\Users\admin\Desktop\SASoutputs\adverseevents.jpg"
alt= "AE Figure" style="width:690 px; height: 204 px">;
;;;;
```

CONCLUSION

The versatility of the Proc STREAM approach is clearly demonstrated in the example in this paper. By restricting oneself to just macro elements, free text can be composed in the exact layout of the desired output, with just a few macro element substitutions. By allowing even macro calls and %includes, the free text can be highly configured for automated rich content output, unsurpassed by any other method, to the best of the author's knowledge.

CODE READY-TO-RUN *(SAS 9.4 ONLY):*

a. THE SIMPLE CASE

```
data aedata;
infile datalines dlm=" ";
input (studyid patid age race sex height weight country trt randdt disease) ($);
datalines;
A123 001 45 white female 6 165 Germany CoolDrug 27Apr99 diabetes
A123 002 45 black male 5 120 Jamaica CoolDrug 12Jul96 asthma
A123 003 45 asian female 5 137 Japan CoolDrug 31Mar97 glaucoma
;
run;
proc sql noprint;
       create table aetable as
              select studyid length=4, patid length=3, age length=2, race length=5,
                                  sex length=6, height length=1, weight length=3, country length=7,
                                  trt length=8, randdt length=7, disease length=8
                                         from aedata;
quit;

proc sql noprint;
       Select studyid, patid, age, race, sex, height, weight, country, trt, randdt, disease
                           into : studyid, :patid, :age, :race, :sex, :height, :weight, :country,
:trt, :randdt, :disease
                                  from aetable;
quit;

filename aefolder 'C:\Users\admin\Desktop\SASoutputs\aenarrative.rtf';
PROC STREAM OUTFILE=aefolder RESETDELIM='goto';
BEGIN
Subject &patid. was a &age.-year old &race. &sex. from &country..  The subject consented to
partake in study
&studyid. on &randdt. and was first exposed to study medication &trt.. The subject was &height.
ft in height,
&weight. lb in weight and suffered from &disease..
;;;;
```

OUTPUT-1:

> Subject 001 was a 45-year old white female from Germany. The subject consented to partake in study A123 on 27Apr99 and was first exposed to study medication CoolDrug. The subject was 6 ft in height, 165 lb in weight and suffered from diabetes.

Figure 1. The Proc STREAM output of a simple patient narrative.

b. THE FULL-BLOWN PATIENT NARRATIVE:

```
data patdata;
infile datalines;
input usubjid $ studyid $ siteid $ trt $ age $ race $ sex $ aept : $15. cat $ randdt : $10.
disease : $30. DiagDate : $10. ConMed : $30.;
datalines;
999004 ABC123 XY005 NewDrug 63 Asian Female Diarrhea Mild 2010-02-24 ColonCancer 2010-01-03
Ibuprofen
```

```
999005 ABC123 XY005 NewDrug 58 White Male Dyspepsia Mild 2010-03-12 LungCancer 2010-01-04
Clemastine
999006 ABC123 XY005 NewDrug 66 Black Male Epilepsy Serious 2010-03-28 BrainCancer 2010-01-05
Acetamenophen
999007 ABC123 XY005 NewDrug 61 Asian Female Arrhythmia Serious 2010-03-24 BoneCancer 2010-01-06
AlphaLipoicAcid
999007 ABC123 XY005 NewDrug 61 Asian Female Diarrhea Mild 2010-03-24 BrainCancer 2010-01-06
AlphaLipoicAcid
999007 ABC123 XY005 NewDrug 61 Asian Female Syncope Serious 2010-03-24 KidneyCancer 2010-01-06
AlphaLipoicAcid
999010 ABC123 XY005 NewDrug 65 White Male Diarrhea Mild 2010-05-24 BoneCancer 2010-01-09
NicotinamideRibose
999011 ABC123 XY005 NewDrug 74 White Female Constipation Mild 2010-06-24 ColonCancer 2010-11-07
Ibuprofen
999012 ABC123 XY005 NewDrug 53 White Female Migraine Serious 2010-02-24 BrainCancer 2010-02-08
Dexamethasone
999013 ABC123 XY005 NewDrug 62 White Male Delirium Serious 2010-07-24 BrainCancer 2010-01-09
Chloroquine
999014 ABC123 XY005 NewDrug 87 White Male Diarrhea Mild 2010-01-24 ColonCancer 2010-01-02
AlphaLipoicAcid
999015 ABC123 XY005 NewDrug 93 Asian Female Nausea Mild 2010-04-24 LungCancer 2010-01-08
Acetamenophen
999016 ABC123 XY005 NewDrug 49 White Female Hallucination Serious 2010-09-24 BrainCancer 2010-01-
04  Hydrochlorothiazide
;
run;

*Step:1-------------------------------------------------------------------------
CREATE BASIC MACRO VARIABLES FOR SUBSETTING DATA AND FOR TEXT STYLING
-------------------------------------------------------------------------------;
%let patient=999005;
%let z=%str(<br/>);*<------for carriage return within HTML;
%let y=%nrstr(&#8226);*<-----for the bullet symbol within HTML;
%let nbsp=%nrstr( );*<----to prevent   resolving warning in LOG;

*Step:2-------------------------------------------------------------------------
CREATE ADVERSE EVENTS AND CONMED MACRO VARIABLES
-------------------------------------------------------------------------------;
proc sql noprint;
select distinct aept into :aetexts separated by "&z.&y."
from patdata
where cat eq "Serious";
quit;

proc sql noprint;
select distinct ConMed into :cmtexts separated by "&z"
from patdata;
quit;

*Step:3-------------------------------------------------------------------------
          RESTRUCTURE INPUT DATA INTO PARAMETER NAME / PARAMETER VALUE PAIRS
             FOR USE AS A LOOK-UP TABLE FOR PARAMETER-GETTING MACRO (%gp)
-------------------------------------------------------------------------------;
data patprofile(keep=usubjid paramname paramval);
dsid=open("patdata");
nrows=attrn(dsid,"NOBS");
ncols=attrn(dsid,"NVARS");
do r=1 to nrows;
       dc=fetchobs(dsid,r);
       do c=2 to ncols;
       usubjid=getvarc(dsid,1);
       paramname=varname(dsid,c);
       test=vartype(dsid,c);
       if test="C" then paramval=getvarc(dsid,c);
       else if test="N" then paramval=put(getvarn(dsid,c),best.);
       output;
       end;
end;
dc=close(dsid);
run;
```

```
%let codetext=%nrstr(data _null_;
if(1=2) then set patprofile;
declare hash pf(dataset:'patprofile');
rc=pf.defineKey('usubjid','paramname');
rc=pf.defineData('paramval');
rc=pf.defineDone();
rcc=pf.check(key:"&patient.",key:"&pname.");
if rcc eq 0 then do;
rcf=pf.find(key:"&patient.",key:"&pname.");
end;call symputx("pv",paramval,"g");run;);


*==================================;
%macro gp(pname);
%let x=%sysfunc(dosubl(&codetext));
&pv.
%mend gp;
*==================================;

*Step:4-----------------------------------------------------------------------------------------
               CREATE MACROS THAT SELECT “He” OR  “She”, “His” OR “Her”
                       BASED ON GENDER VALUE IN INPUT DATA
-----------------------------------------------------------------------------------------------;

%let vv=%gp(sex);

%macro hsh();
%global heshe;
%if "&vv." eq "Female" %then %let heshe=She;
%else %if "&vv." eq "Male" %then %let heshe=He;
&heshe
%mend hsh;

%macro hsr();
%global hisher;
%if "&vv." eq "Female" %then %let hisher=her;
%else %if "&vv." eq "Male" %then %let hisher=his;
&hisher
%mend hsr;


*Step:5-----------------------------------------------------------------------------------------
       GENERATE CONCOMITANT MEDICATION TABLE FOR INSERTION INTO PROC STREAM TEXT
                      USING ODS REPORT WRITING INTERFACE
-----------------------------------------------------------------------------------------------;

%let outputlocation=C:\Users\admin\Desktop\SASoutputs\conmedtable.html;

        *STEP D1--INITIALIZE ODS REPORT WRITING INTERFACE------;
        options nonumber nodate nocenter linesize=max pagesize=min;
               title;*<----to suppress system title;
        ods listing close;
        ods escapechar="^";
               ods html file="&outputlocation." ;

               data _null_;
        length blank0 $1 blank1 $10 blank2 $30;
        blank0=byte(32);
        blank1=repeat(byte(32),10);
        blank2=repeat(byte(32),30);

               *STEP D2---CREATE LOOK-UP TABLE---------------------------;
               if(1=2) then set patprofile;
               declare hash pf(dataset:'patprofile');
               rc=pf.defineKey('usubjid','paramname');
               rc=pf.defineData('paramval');
               rc=pf.defineDone();


        *STEP D3---CREATE ODS OBJECT-----------------------------;
```

```
        declare odsout summ();
        *STEP D4---BEGIN ODS RWI/ CREATE TITLE AND HEADINGS-------;

        summ.title(data:"CONCOMITANT MEDICATIONS:",start:2);
        summ.layout_gridded(columns:1);
        summ.region();
        summ.line();
        summ.table_start();
        summ.row_start();

        summ.format_cell( text:"Treatment#Arm",overrides:"cellwidth=2cm just=center" ,
split:"#");
        summ.format_cell( text:"Patient#ID",overrides:"cellwidth=1.5cm just=center" , split:"#");
        summ.format_cell( text:"Diagnosis#Date",overrides:"cellwidth=2.0cm just=center" ,
split:"#");
        summ.format_cell( text:"Race",overrides:"cellwidth=1.0cm just=center" , split:"#");
        summ.format_cell( text:"Sex",overrides:"cellwidth=1.0cm just=center" , split:"#");
        summ.format_cell( text:"Age",overrides:"cellwidth=1.0cm just=center" , split:"#");
        summ.format_cell( text:"Disease",overrides:"cellwidth=3.0cm just=center");
        summ.format_cell( text:"Con Meds",overrides:"cellwidth=5.0cm just=center");

        summ.row_end();

               rcf=pf.find(key:"&patient.",key:"trt");
               TREATARM=paramval;call missing(paramval);
               rcf=pf.find(key:"&patient.",key:"usubjid");
               USUBJID=&patient.;
               rcf=pf.find(key:"&patient.",key:"DiagDate");
               DIAGNODT=paramval;call missing(paramval);
               rcf=pf.find(key:"&patient.",key:"race");
               RACEVAL=paramval;call missing(paramval);
               rcf=pf.find(key:"&patient.",key:"sex");
               SEXVAL=paramval;call missing(paramval);
               rcf=pf.find(key:"&patient.",key:"age");
               AGEVAL=paramval;call missing(paramval);
               rcf=pf.find(key:"&patient.",key:"disease");
               DIAGNOSE=paramval;call missing(paramval);
               rcf=pf.find(key:"&patient.",key:"ConMed");
               CONMEDS=paramval;call missing(paramval);

        *STEP D5-----CREATE ROWS OF SUMMARY DATA------------------;
        summ.row_start();
        summ.format_cell( text:TREATARM, overrides:"cellwidth=2cm just=center");
        summ.format_cell( text:USUBJID, overrides:"cellwidth=1.5cm just=center");
        summ.format_cell( text:DIAGNODT, overrides:"cellwidth=2.0cm just=center");
        summ.format_cell( text:RACEVAL, overrides:"cellwidth=1.0cm just=center");
        summ.format_cell( text:SEXVAL, overrides:"cellwidth=1.0cm just=center");
        summ.format_cell( text:AGEVAL, overrides:"cellwidth=1.0cm just=center");
        summ.format_cell( text:DIAGNOSE, overrides:"cellwidth=3.0cm just=center");
        summ.format_cell( text:CONMEDS, overrides:"cellwidth=5.0cm just=center");
        summ.row_end();

        summ.table_end();
        summ.line();
        summ.region();
        *STEP D6-----CREATE TABLE FOOTNOTE------------------------;
        summ.table_start();
        summ.row_start();
        summ.format_cell(inhibit:"BTLR", text:"FOOTNOTE:...Generated with SAS version
9.4",column_span:5,overrides:"cellwidth=15cm");
        summ.row_end();
        summ.table_end();
        summ.layout_end();
        stop;
run;



*Step:6 ------------------------------------------------------------------------
               THE ACTUAL PROC STREAM CODE
```

```
-------------------------------------------------------------------------------------;
ods html close;


filename product "C:\Users\admin\Desktop\SASoutputs\patnarrative.html";
filename tabout "C:\Users\admin\Desktop\SASoutputs\conmedtable.html";
filename figout "C:\Users\admin\Desktop\SASoutputs\aefigure.html";


proc stream outfile=product resetdelim='goto';
BEGIN
Patient No.: &patient. &z.
Treatment Arm: %gp(trt) &z.
Demographics: %gp(age)-year-old %gp(race) %gp(sex) &z.
&z.&z.
This %gp(age)-year-old %gp(race) %gp(sex) was admitted to the %gp(studyid) clinical trial at the
%gp(siteid) study site on %gp(randdt).
&z.
%hsh() was diagnosed with %gp(disease) on %gp(DiagDate) by a team of oncologists at site
%gp(siteid)&z.
%hsh() exhibited, during the course of the trial, the following serious disorders:
&z.&y.&aetexts.. &z.
%hsh() had been taking before and during the study, some non-study medication as shown in the
table below: &z.
goto; %include tabout; &z.
The figure below summarizes all %hsr() non-serious adverse events (Placebo vs Multivitamins vs
NewDrug): &z.
goto; <img src="C:\Users\admin\Desktop\SASoutputs\adverseevents.jpg" alt="AE Figure"
style="width:690 px; height: 204 px">;
;;;;

*=========================END OF PROGRAM=====================================;
```

OUTPUT

Patient No.: 999005
Treatment Arm: NewDrug
Demographics: 58-year-old White Male

This 58-year-old White Male was admitted to the ABC123 clinical trial at the XY005 study site on 2010-03-12.
He was diagnosed with LungCancer on 2010-01-04 by a team of oncologists at site XY005
He exhibited, during the course of the trial, the following serious disorders:
•Arrhythmia
•Delirium
•Epilepsy
•Hallucination
•Migraine
•Syncope.
He had been taking before and during the study, some non-study medication as shown in the table below:

CONCOMITANT MEDICATIONS:

Treatment Arm	Patient ID	Diagnosis Date	Race	Sex	Age	Disease	Con Meds
NewDrug	999005	2010-01-04	White	Male	58	LungCancer	Clemastine

FOOTNOTE:...Generated with SAS version 9.4

The figure below summarizes all his non-serious adverse events (Placebo vs Multivitamins vs NewDrug):

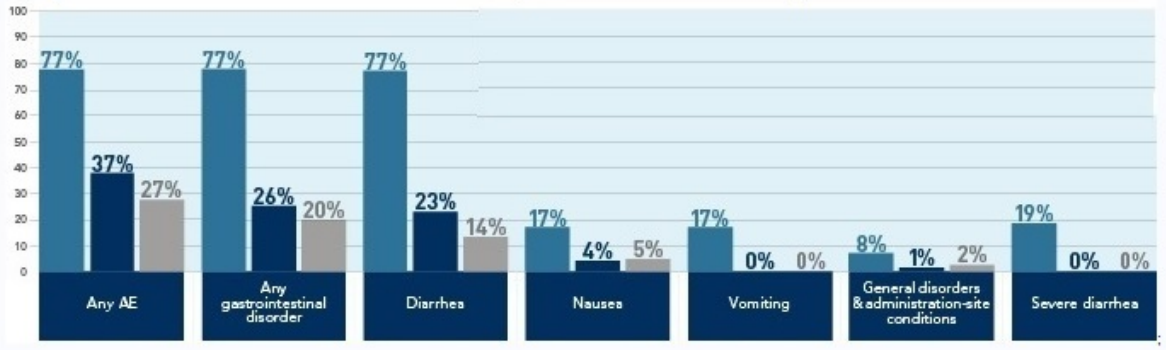

Figure 2. The Proc STREAM output of a complex patient narrative.

REFERENCES

1. The STREAM Procedure (SAS Institute Inc)
http://support.sas.com/documentation/cdl/en/proc/67327/HTML/default/viewer.htm#n12zrkr08eiacmn17lcv4fmt79tb.htm

2. Henderson, Don. "PROC STREAM and SAS® Server Pages: Generating Custom HTML Reports"
Proceedings of the SAS Global Forum 2014 Conference, Paper 1738-2014
http://support.sas.com/resources/papers/proceedings14/1738-2014.pdf

3. Langston, Rick. "Submitting SAS® Code On The Side"
Proceedings of the SAS Global Forum 2013 Conference, Paper 032-2013
https://support.sas.com/resources/papers/proceedings13/032-2013.pdf

ACKNOWLEDGMENTS

The author is very grateful to Don Henderson (Henderson Consulting Services) and Rick Langston (SAS Institute) for their very useful input, on many occasions.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please contact the author at:

Joseph W. Hinson, PhD
inVentiv Health
504 Carnegie Center
Princeton, NJ, 08540
1-609-951-6596
joehinson@outlook.com