

Generating SAS Code Using SAS

Xiu (Sue) Chen
Synectechs, Inc.

Presented to HASUG
November 10, 2005

Overview

- Stir some interest in this technique
- Provide three detailed examples using real life applications
- All examples are based on SAS v8.2

Agenda

- Example 1: Last Date Known to be Alive
- Example 2: Comments Listing
- Example 3: SAS Graph Tick Mark Labels

Example 1: Last Date Known to be Alive

- Objective: Find the most recent date
- Data Sources: 33 datasets, 170 variables
- Complications: Lack of consistency in naming of datasets and variables
- Respite: All date variables have the keyword 'DATE' in their variable labels
- Goal: To create the program with minimal coding and flexible structure so it could be easily adapted for other protocols

Example 1, Step 1a

- Get a list of datasets and variables

```
**** Identify datasets/variables with date data ****;  
proc contents data=raw._all_ position noprint  
              out=datasets(keep=mem: name label);  
run;
```

Example 1, Step 1b

- Keep date variables and filter out extraneous data

```
**** Specify the desired date variables ****;  
data datevars;  
  set datasets(drop=memtype memlabel);  
  where index(upcase(label), 'DATE')>0;  
;  
  ** Exclude unwanted date variables **;  
  if index(memname, 'C')>0 or memname in ('PLATE500', 'PLATE510', 'PLATE511') or  
  index(upcase(label), 'SIGNATURE')>0 or  
  index(upcase(label), 'INVESTIGATOR')>0 or  
  index(upcase(label), 'REPORT')>0 or  
  index(upcase(label), 'BIRTH')>0 or  
  index(upcase(label), 'AUTHORIZED')>0 or  
  index(upcase(label), 'REQUESTED')>0 or  
  index(upcase(label), 'N/AP')>0 then delete;  
run;
```

Example 1, Step 2

- Auto-generate the SAS code

```
**** Write SAS code to combine selected date data ****;  
filename sascode 'sascode.sas';  
  
data _null_;  
  file sascode old;  
  set datevars end=eof;  
  
  if _n_=1 then put 'data dates;';  
  if _n_=1 then put 'set';  
  
  put "raw." memname "(keep=dfplate patid " name "rename=(" name "=datevar))";  
  if eof then put ";";  
  if eof then put "run;";  
run;
```

Example 1, Step 3

- Execute the self-generated code:

```
%include 'sascode.sas';
```

```
data dates;
set
raw.PLATE001 (keep=dfplate patid DATEABC rename=(DATEABC =datevar))
raw.PLATE001 (keep=dfplate patid DATEXYZ rename=(DATEXYZ =datevar))
raw.PLATE001 (keep=dfplate patid DTEABC rename=(DTEABC =datevar))
raw.PLATE001 (keep=dfplate patid DTEXYZ rename=(DTEXYZ =datevar))
etc...
raw.PLATE032 (keep=dfplate patid DTEAAA rename=(DTEAAA =datevar))
raw.PLATE032 (keep=dfplate patid DTEBBB rename=(DTEBBB =datevar))
raw.PLATE032 (keep=dfplate patid DTECCC rename=(DTECCC =datevar))
;
run;
```

Example 1, Step 4

- Sort data by subject ID and date variable
- Select the last record per-subject as the ‘last date known to be alive’

Example 2: Comments Listing

- Objective: Create a listing containing comments data
- Data Source: 10 datasets, 48 variables

Example 2, Step 1a

- Get a list of datasets and variables

```
**** Identify datasets/variables with comments data ****;  
proc contents data=raw._all_ position noprint  
              out=datasets(keep=mem: name label);  
run;
```

Example 2, Step 1b

- Filter out extraneous data
- Make sure to track all sources

```
**** Specify the desired comment variables ****;  
data vars(drop=memlabel);  
  set datasets(drop=memtype);  
  length name3 $ 9;  
  where index(memname,'PLATE')>0 and (  
    index(upcase(label),'DESCRIBE')>0 or  
    index(upcase(label),'REASON')>0 or  
    index(upcase(label),'COMMENT')>0 or  
    index(upcase(label),'DESC')>0 or  
    index(upcase(label),'SPECIFY')>0)  
;  
  if index(name,'CMNT')>0 then name2=trim(name)||'_'||substr(memname,7,2);  
  else name2=name;  
  
  name3='_'||name2;  
  
  if memname in ('PLATE510','PLATE511','PLATE205') then delete;  
run;
```

Example 2, Step 2a

- Generate data extraction SAS code

```
**** Write SAS code to combine selected comment data ****;
filename sascode2 'sascode2.sas';

data _null_;
  file sascode2 old;
  set vars end=eof;

  if _n_=1 then put 'data cmnts;';
  if _n_=1 then put 'set';

  put "raw." memname "(keep=dfplate dfseq patid "
      name "rename=(" name "=comment)" " in=" name3 ")";

  if eof then put ";";
run;
```

Example 2, Step 2b

- Generate data tracking SAS code

```
data _null_;
  file sascode3 old;
  set vars end=eof;

  if _n_=1 then put 'length source $ 10;';

  put "if " name3 "then source='" name2 "'";

  if eof then put "if comment^=' '";
  if eof then put "run;";
run;
```

Example 2, Step 3

- Execute the self-generated code

```
%include "sascode2.sas";  
%include "sascode3.sas";
```

```
(sascode2.sas)
```

```
data cmnts;  
set  
raw.PLATE001 (keep=dfplate dfseq patid ECGABN rename=(ECGABN =comment) in=_ECGABN )  
raw.PLATE002 (keep=dfplate dfseq patid CMMNT rename=(CMMNT =comment) in=_CMMNT_02 )  
raw.PLATE002 (keep=dfplate dfseq patid NOPREG rename=(NOPREG =comment) in=_NOPREG )  
...
```

```
(sascode3.sas)
```

```
length source $ 10;  
if _ECGABN then source='ECGABN ';  
if _CMMNT_02 then source='CMMNT_02 ';  
if _NOPREG then source='NOPREG ';  
...  
if comment^=' ';  
run;
```

Example 2, Step 4

- Use proc report to generate the listing

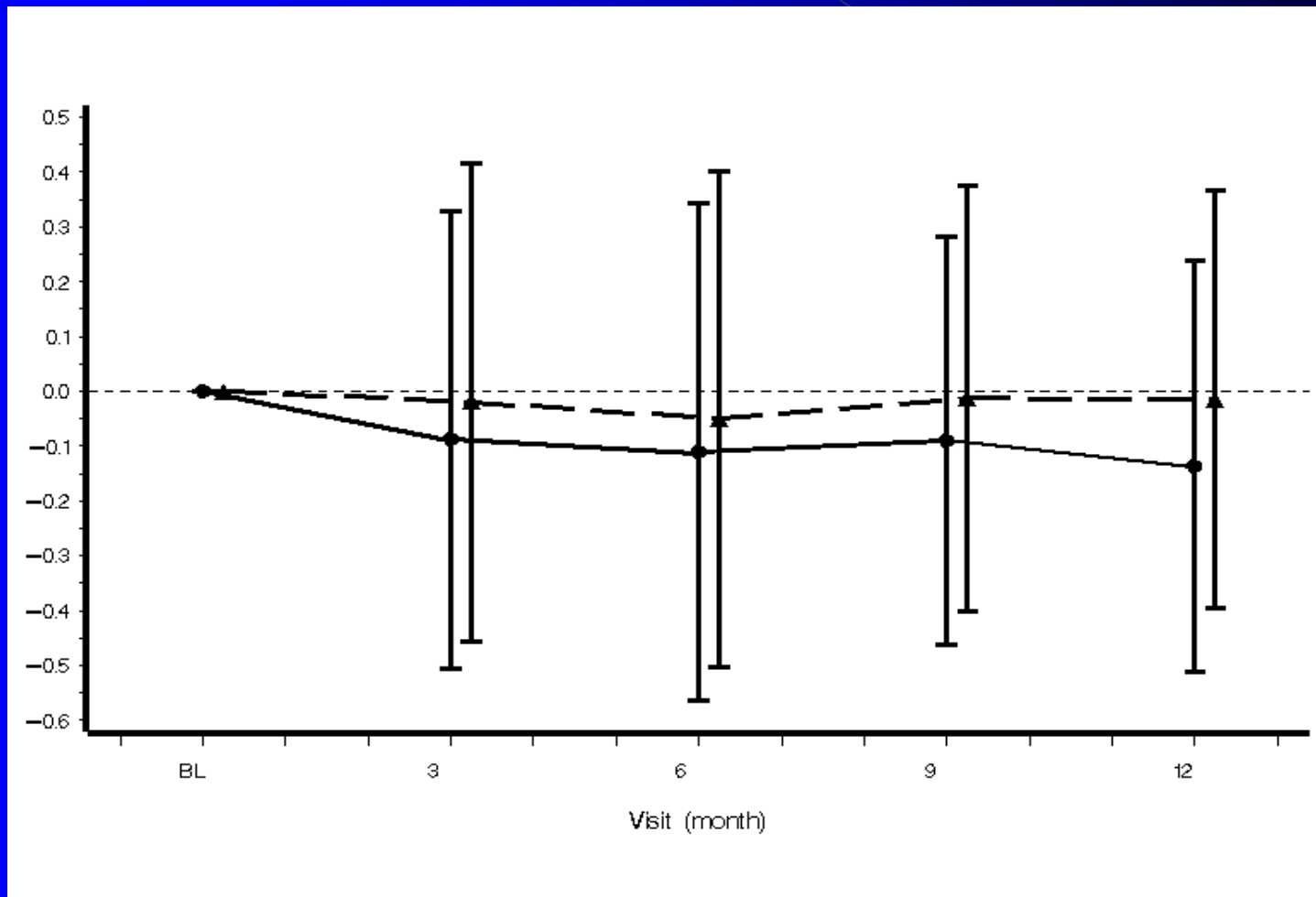
```
Listing Bxx  
Comments and Other Text Fields
```

```
-----  
Subject  
Number          Data From      Visit #      CRF Question/Source Variable      Text  
-----
```

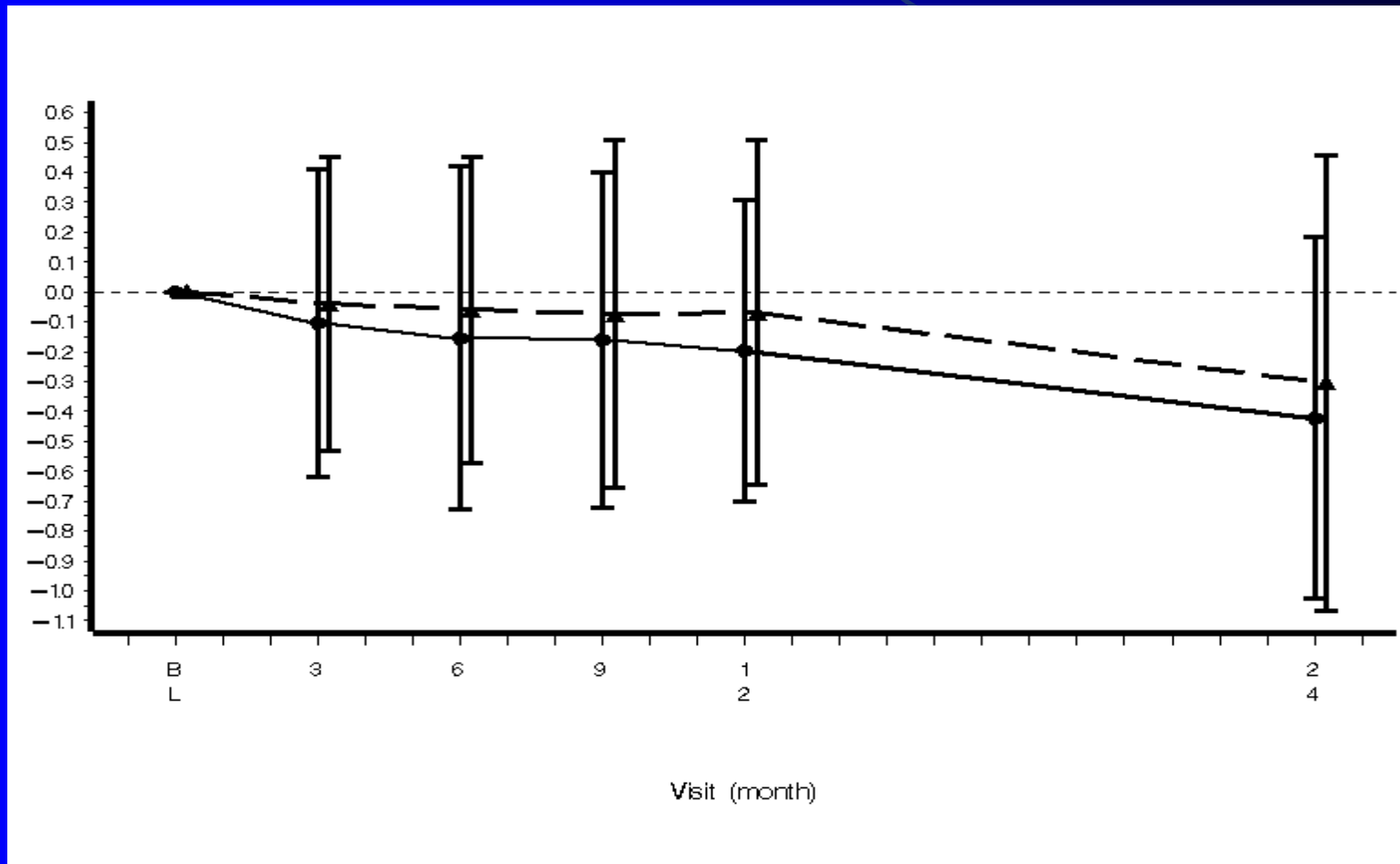
Example 3, SAS Graph tick mark Labels

- Objective: Create tick mark labels for only those visits with data

1st figure



2nd figure



Example 3, Step 1

- Create a dataset containing the visit data

```
proc sort data=stats out=tmp(keep=&winvar) nodupkey;  
  by &winvar;  
run;
```

Example 3, Step 2

- Create data-driven x-axis values

```
filename INCFILE "sascode.sas";

/* Create x-axis label values */
%let timevar=&winvar;
data _null_;
  file INCFILE old;
  set tmp end=eof;

  if _n_=1 then put '%' 'macro sascodes;';
  if NOT eof then put "%" "if &" "i=" &timevar "%" 'then "' &timevar winf. '";' "%" "else";
  else do;
    put "%" "if &" "i=" &timevar "%" 'then "' &timevar winf. '";' "%" 'else " ";';
  end;
  if eof then put '%' 'mend sascodes;';
run;
```

Example 3, Step 2 (continued)

- Self-generated SAS code

```
%macro sascodes;  
%if &i=0 %then "BL ";%else  
%if &i=12 %then "3 ";%else  
%if &i=24 %then "6 ";%else  
%if &i=36 %then "9 ";%else  
%if &i=48 %then "12 ";%else  
%if &i=96 %then "24 ";%else " ";  
%mend sascodes;
```

Example 3, Step 3

Execute the self-generated code

```
value=(h=0.9
      %do i=%eval(&v0-&inc) %to %eval(&w0+&inc) %by &inc;
          %sascodes
      %end; )
```

```
MPRINT(SASCODES): " "
MPRINT(SASCODES): "BL "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): "3 "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): "6 "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): "9 "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): "12 "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): " "
MPRINT(SASCODES): "24 "
MPRINT(SASCODES): " "
```

Conclusion

- The technique shown in this presentation is simple yet can be used to create highly data specific statements using rather generic programs
- Using this technique eliminates potential errors prevalent in manually keyed statements
- This technique is especially effective when generating vast quantity of repetitive code

Questions and Answers

Contact Information

Comments and questions are welcome

Xiu (Sue) Chen

Synectechs, Inc.

190 Governor Winthrop Blvd.

New London, CT 06320

860-447-0170

chenx@synec.com