

Convert dataset variables into macro variables: a new approach with the CALL SET routine and the SCL FETCH function

Patrice Agator
PhUSE 2010 – BERLIN
Paper CC02

Introduction

- New approach to convert dataset variables into macro variables using:
 - CALL SET routine
 - SCL FETCH function
- Presentation of the origin of the SAS code
- Example using laboratory data to explain the use of this non-friendly SAS code

Origin of the approach

- SAS code developed for the production of patient narratives at Sanofi-Aventis

```
%MACRO patient_profile;  
  %let dsid=%sysfunc(open(work.ADDM,I));  
  %if &dsid gt 0 %then %do;  
    %syscall set(dsid);  
    %do %while (%sysfunc(fetch(&dsid)) eq 0);  
      title1 "Unique Subject Identifier = %trim(&usubjid)";  
      title2 "Demography";  
      proc print data=ADDM noobs;  
        var age race dmbegdt dmenddt;  
        where usubjid eq "%trim(&usubjid)";  
      run;  
      title2 "Laboratory Data";  
      proc print data=ADLB noobs;  
        var lbtestcd visitnum lbdn lbstresn;  
        where usubjid eq "%trim(&usubjid)";  
      run;  
    %end;  
  %end;  
  %let rc=%sysfunc(close(&dsid));  
%MEND patient_profile;  
%patient_profile;
```

Origin of the approach

- Patient profiles/narratives are listings repeated for each patient

```
%MACRO patient_profile;
  %let dsid=%sysfunc(open(work.ADDM,I));
  %if &dsid gt 0 %then %do;
    %syscall set(dsid);
    %do %while (%sysfunc(fetch(&dsid)) eq 0);
      title1 "Unique Subject Identifier = %trim(&usubjid)";
      title2 "Demography";
      proc print data=ADDM noobs;
        var age race dmbegdt dmenddt;
        where usubjid eq "%trim(&usubjid)";
      run;
      title2 "Laboratory Data";
      proc print data=ADLB noobs;
        var lbtestcd visitnum lbdtd lbstresn;
        where usubjid eq "%trim(&usubjid)";
      run;
    %end;
  %end;
  %let rc=%sysfunc(close(&dsid));
%MEND patient_profile;
%patient_profile;
```

Origin of the approach

- Where does the macro variable usubjid come from?

```
%MACRO patient_profile;
  %let dsid=%sysfunc(open(work.ADDM,I));
  %if &dsid gt 0 %then %do;
    %syscall set(dsid);
    %do %while (%sysfunc(fetch(&dsid)) eq 0);
      title1 "Unique Subject Identifier = %trim(&usubjid)";
      title2 "Demography";
      proc print data=ADDM noobs;
        var age race dmbegdt dmenddt;
        where usubjid eq "%trim(&usubjid)";
      run;
      title2 "Laboratory Data";
      proc print data=ADLB noobs;
        var lbtestcd visitnum lbdtt lbstresn;
        where usubjid eq "%trim(&usubjid)";
      run;
    %end;
  %end;
  %let rc=%sysfunc(close(&dsid));
%MEND patient_profile;
%patient_profile;
```

Origin of the approach

- CALL SET routine: Links SAS data set variables to DATA step or macro variables that have the same name and data type
- FETCH function: Reads the next non-deleted observation from a SAS data set into the Data Set Data Vector (DDV)

```
%MACRO patient_profile;  
  %let dsid=%sysfunc(open(work.ADDM,I));  
  %if &dsid gt 0 %then %do;  
    %syscall set(&dsid);  
    %do %while (%sysfunc(fetch(&dsid)) eq 0);  
      title1 "Unique Subject Identifier = %trim(&usubjid)";  
      title2 "Demography";  
      proc print data=ADDM noobs;  
        var age race dmbegdt dmenddt;  
        where usubjid eq "%trim(&usubjid)";  
      run;  
      title2 "Laboratory Data";  
      proc print data=ADLB noobs;  
        var lbtestcd visitnum lbdtt lbstresn;  
        where usubjid eq "%trim(&usubjid)";  
      run;  
    %end;  
  %end;  
  %let rc=%sysfunc(close(&dsid));  
%MEND patient_profile;  
%patient_profile;
```

Example using laboratory data – Source data

- Dataset containing overall laboratory data
- Dataset matching CDISC standard (not mandatory)

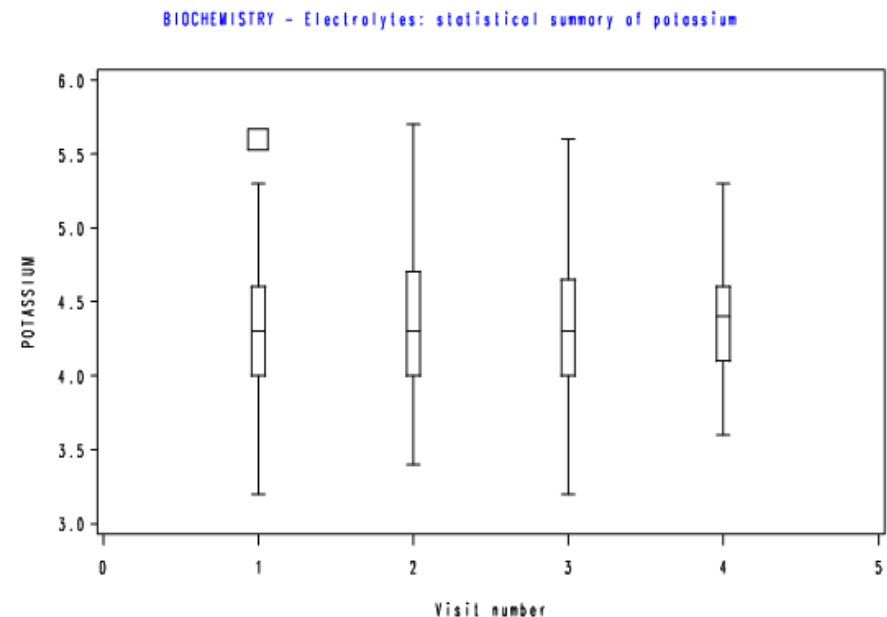
SUBJID	LBTESTCD	LBTEST	LBSTRESN	LBSTUNIT	VISIT	VISITNUM	LBDT	LBTM	LBCAT	LBSCAT	aman	ama
840003001	K	POTASSIUM	4.4	MMOL/L	DAY 40	4	2007-08-30	7:10	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003001	NA	SODIUM	141	MMOL/L	DAY 40	4	2007-08-30	7:10	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003002	HB	HEMOGLOBIN	141	G/L	D-30 TO D1	1	2007-08-08	9:00	HEMATOLOGY	RED BLOOD CELLS AND PLATELETS	1	Group A
840003002	K	POTASSIUM	4.6	MMOL/L	D-30 TO D1	1	2007-08-08	9:00	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003002	NA	SODIUM	138	MMOL/L	D-30 TO D1	1	2007-08-08	9:00	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003002	K	POTASSIUM	5.7	MMOL/L	DAY 5	2	2007-08-14	8:00	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003002	NA	SODIUM	139	MMOL/L	DAY 5	2	2007-08-14	8:00	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003002	K	POTASSIUM	4.3	MMOL/L	DAY 10 - DAY 28	3	2007-08-21	9:00	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003002	NA	SODIUM	131	MMOL/L	DAY 10 - DAY 28	3	2007-08-21	9:00	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003002	HB	HEMOGLOBIN	138	G/L	DAY 40	4	2007-09-17	8:50	HEMATOLOGY	RED BLOOD CELLS AND PLATELETS	1	Group A
840003002	K	POTASSIUM	4.4	MMOL/L	DAY 40	4	2007-09-17	8:50	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003002	NA	SODIUM	131	MMOL/L	DAY 40	4	2007-09-17	8:50	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003003	HB	HEMOGLOBIN	132	G/L	D-30 TO D1	1	2007-08-10	9:00	HEMATOLOGY	RED BLOOD CELLS AND PLATELETS	2	Group B
840003003	K	POTASSIUM	5.3	MMOL/L	D-30 TO D1	1	2007-08-10	9:00	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003003	NA	SODIUM	138	MMOL/L	D-30 TO D1	1	2007-08-10	9:00	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003003	K	POTASSIUM	4.6	MMOL/L	DAY 5	2	2007-08-14	7:00	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003003	NA	SODIUM	138	MMOL/L	DAY 5	2	2007-08-14	7:00	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003003	K	POTASSIUM	5.3	MMOL/L	DAY 10 - DAY 28	3	2007-08-29	8:15	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003003	NA	SODIUM	137	MMOL/L	DAY 10 - DAY 28	3	2007-08-29	8:15	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003003	K	POTASSIUM	5	MMOL/L	DAY 40	4	2007-09-13	9:00	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003003	NA	SODIUM	137	MMOL/L	DAY 40	4	2007-09-13	9:00	BIOCHEMISTRY	ELECTROLYTES	2	Group B
840003004	HB	HEMOGLOBIN	139	G/L	D-30 TO D1	1	2008-01-17	10:10	HEMATOLOGY	RED BLOOD CELLS AND PLATELETS	1	Group A
840003004	K	POTASSIUM	3.5	MMOL/L	D-30 TO D1	1	2008-01-17	10:10	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003004	NA	SODIUM	139	MMOL/L	D-30 TO D1	1	2008-01-17	10:10	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003004	K	POTASSIUM	3.6	MMOL/L	DAY 5	2	2008-01-22	10:10	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003004	NA	SODIUM	140	MMOL/L	DAY 5	2	2008-01-22	10:10	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003004	K	POTASSIUM	3.5	MMOL/L	DAY 10 - DAY 28	3	2008-02-12	10:15	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003004	NA	SODIUM	141	MMOL/L	DAY 10 - DAY 28	3	2008-02-12	10:15	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003004	HB	HEMOGLOBIN	139	G/L	DAY 40	4	2008-02-20	10:20	HEMATOLOGY	RED BLOOD CELLS AND PLATELETS	1	Group A
840003004	K	POTASSIUM	3.8	MMOL/L	DAY 40	4	2008-02-20	10:20	BIOCHEMISTRY	ELECTROLYTES	1	Group A
840003004	NA	SODIUM	139	MMOL/L	DAY 40	4	2008-02-20	10:20	BIOCHEMISTRY	ELECTROLYTES	1	Group A

Example using laboratory data - Objective

- Objective for each laboratory test:
 - Create a summary table
 - Create a graph

BIOCHEMISTRY - Electrolytes: statistical summary of potassium

POTASSIUM	N	Mean	Std	Min	Max
<u>Visitnum</u>					
1	79	4.31	0.47	3.20	5.60
2	79	4.31	0.48	3.40	5.70
3	72	4.33	0.53	3.20	5.60
4	67	4.35	0.40	3.60	5.30



Example using laboratory data - SAS code

- From the overall laboratory data, SAS code is divided in 2 parts:
 - 1st part: creation of a summary dataset from the overall laboratory data
 - 2nd part: use the summary dataset, the CALL SET routine and the SCL FETCH function

Example using laboratory data - SAS code

- 1st part : summarize overall laboratory data with an SQL statement

```
proc sql;
  create table _pilot_ as
  select distinct lbcat, propcase(lbscat) as LBSCAT, lbtestcd, lbtest
    ,min(visitnum) as lower_xaxis
    ,max(visitnum) as upper_xaxis
    ,min(lbstresn) as min_lbstresn
    ,max(lbstresn) as max_lbstresn
    ,case lbtestcd
      when "HB" then 10
      when "NA" then 5
      when "K" then 0.5
      else 1
    end as step label="Step definition for graphs"
    ,floor(min(lbstresn/calculated step))*(calculated step) as lower_yaxis
    ,ceil(max(lbstresn/calculated step))*(calculated step) as upper_yaxis
  from ADLB
  where lbstresn ne .
  group by lbtestcd
  ;
quit;
```

Example using laboratory data - SAS code

- The result is a table with 1 observation for each laboratory test

```
proc sql;
  create table _pilot_ as
  select distinct lbcat, propcase(lbcat) as LBSCAT, lbtestcd, lbtest
    ,min(visitnum) as lower_axis
    ,max(visitnum) as upper_axis
    ,min(lbstresn) as min_lbstresn
    ,max(lbstresn) as max_lbstresn
    ,case lbtestcd
      when "HB" then 10
      when "NA" then 5
      when "K" then 0.5
      else 1
    end as step label="Step definition for graphs"
    ,floor(min(lbstresn/calculated step))*(calculated step) as lower_yaxis
    ,ceil(max(lbstresn/calculated step))*(calculated step) as upper_yaxis
  from ADLB
  where lbtestresn ne .
  group by lbtestcd
  ;
quit;
```

The FLOOR function takes one numeric argument and returns the largest integer which is less than or equal to the argument.

The CEIL function also takes one numeric argument and returns the smallest integer which is greater than or equal to the argument.

	LBCAT	LBSCAT	LBTESTCD	LBTEST	lower_axis	upper_axis	min_lbstresn	max_lbstresn	step	lower_yaxis	upper_yaxis
1	BIOCHEMISTRY	Electrolytes	K	POTASSIUM	1	4	3.2	5.7	0.5	3	6
2	BIOCHEMISTRY	Electrolytes	NA	SODIUM	1	4	113	148	5	110	150
3	HEMATOLOGY	Red Blood Cells And Platelets	HB	HEMOGLOBIN	1	4	94	180	10	90	180

Example using laboratory data - SAS code

2nd part : produce the report for each laboratory parameter (1/3)

- start a macro program, open `_PILOT_` dataset and use CALL SET routine and FETCH function to convert dataset variables into macro variables

```
%MACRO lab_report;  
  * -- Open the _pilot_ dataset which contains each laboratory test (I corresponds  
    to Input mode);  
  %let dsid=%sysfunc(open(_pilot_,I));  
  %if &dsid gt 0 %then %do;  
    * -- Invokes the SAS CALL SET routine (it links SAS data set variables  
      to macro variables) (i.e. creates &lbcat, &lbscat, &lbtestcd,...);  
    %syscall set(dsid);  
    * -- Loop the _pilot_ data set for each observation (laboratory test);  
    %do %while (%sysfunc(fetch(&dsid)) eq 0);
```

(continued)

4 - FETCH function reads each observation in `_PILOT_` dataset

3 - Invokes the CALL SET routine to create macro variables with the same name as the variables in `_PILOT_` dataset

2 - Use the SCL function to open the dataset `_PILOT_`

1 - Start a macro program without any macro parameters

Example using laboratory data - SAS code

- 2nd part : produce the report for each laboratory parameter (2/3)
 - create reporting procedures and use macro variables

```
title1 "%trim(&lbcat) - %trim(&lbscat): summary of %lowcase(&lbtest)";
proc tabulate data=ADLB formchar="-----" noseps format=6.2;
  class visitnum visit;
  var lbstresn;
  tables visitnum="Visitnum", lbstresn=" *(n*f=6. mean std min max) /
    box="&lbtest";
  where lbtestcd eq "%trim(&lbtestcd)";
run;

title1 c=blue h=0.8 f=TimesRoman
      "%trim(&lbcat) - %trim(&lbscat): statistical summary of %lowcase(&lbtest)";

symbol1 value=square interpol=boxt10 h=2 w=2.0 c=black;

axis1 label=(H=0.8 A=0 R=0 F="TimesRoman" "Visit number")
value=(H=0.8 F="TimesRoman")
order=(%eval(&lower_xaxis-1) to %eval(&upper_xaxis+1) by 1 ) minor=none;
axis2 label=(H=0.8 A=90 R=0 F="TimesRoman" "%trim(&lbtest)")
order=(&lower_yaxis to &upper_yaxis by &step)
value=(H=0.8 F="TimesRoman") minor=none;

proc gplot data=ADLB;
  plot lbstresn * visitnum / haxis=axis1 vaxis=axis2;
  where lbtestcd eq "%trim(&lbtestcd)";
run;
```

(continued)

Example using laboratory data - SAS code

- 2nd part : produce the report for each laboratory parameter (3/3)
 - close the do loops
 - close the _PILOT_ dataset
 - end the macro program
 - call the macro program

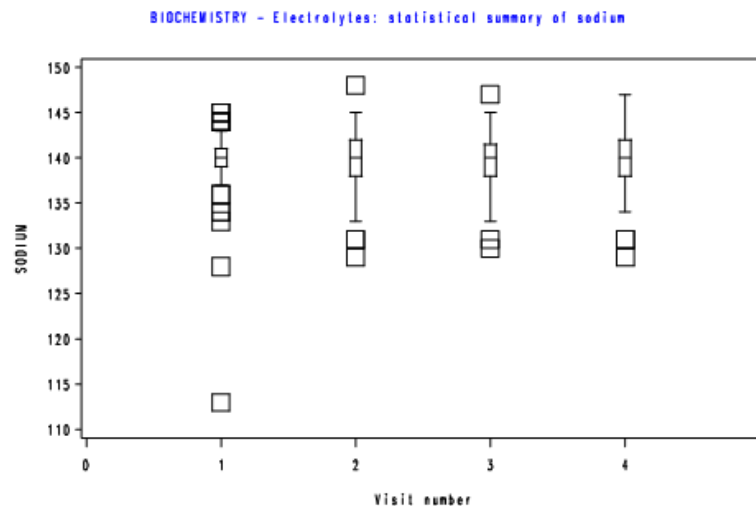
```
        * -- END OF DO WHILE;  
        %end;  
    %* -- END OF DSID > 0;  
    %end;  
    %let rc=%sysfunc(close(&dsid));  
%MEND lab_report;  
  
%lab_report;
```

Example using laboratory data - Result

- For each laboratory parameter a table and a graph are produced dynamically from the different values of the dataset variables

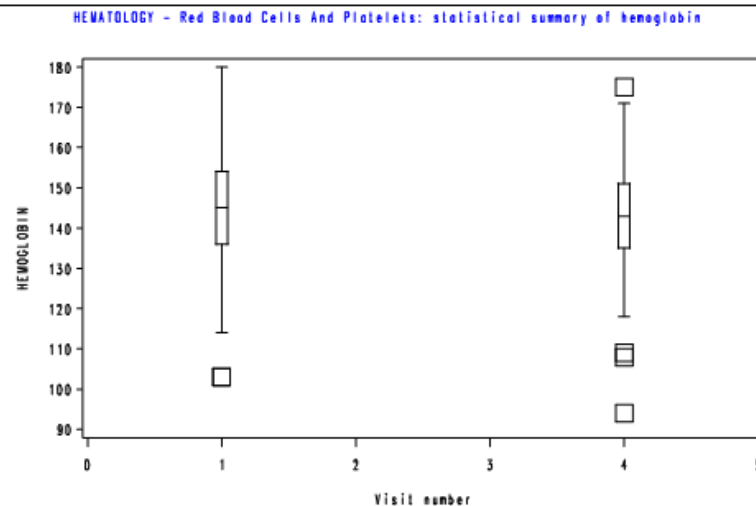
BIOCHEMISTRY - Electrolytes: statistical summary of sodium

SODIUM	N	Mean	Std	Min	Max
<u>Visitnum</u>					
1	78	139.60	4.16	113.00	145.00
2	79	139.86	3.18	129.00	148.00
3	72	139.82	2.96	130.00	147.00
4	67	139.51	3.46	129.00	147.00



HEMATOLOGY - Red Blood Cells And Platelets: statistical summary of hemoglobin

HEMOGLOBIN	N	Mean	Std	Min	Max
<u>Visitnum</u>					
1	77	143.60	15.83	103.00	180.00
4	65	142.34	15.24	94.00	175.00



Conclusion

- New approach to create macro variables is easy-to-use and powerful
- Macro variables are dynamically created and assigned
- Reduction and simplification of the SAS code (no CALL SYMPUT, no PROC SQL...INTO, no &&&, no &i)
- Can apply to many data (i.e. Vital Signs, ECG, Questionnaire, ...)
- Code is widely reusable

Thank you and
enjoy this technique!

