


SUGME
April 17
2009

*Bulletproofing Your
SAS® Results*

Vanessa C. Hayden
Independent Consultant



SUGME
April 17
2009

An olio of bulletproofing ideas

- Prevent yourself from making coding errors
- Save yourself time
 - Answering questions
 - Rechecking your code
- Create client confidence
- Have "vacation security"
- Facilitate future changes
- Endear yourself to the programmer who inherits your code

2

SUGME
April 17
2009

Principles

- ✓ Conditional LOG messages
- ✓ De-magic numbering
- ✓ Automated links in code
- ✓ Documented output

3

SUGME
April 17
2009

**Recommendation 1:
Create conditional warning messages in log.**

4

SUGME
April 17
2009

*Conditional warnings to check your data
assumptions: Example 1*

```

data new;
  set old;
  by usubjid visitcd;
  select (trtgrp);
    when ("Placebo")      trtcd = 0;
    when ("MyDrug 100 mg") trtcd = 1;
    when ("MyDrug 250 mg") trtcd = 2;
    when ("MyDrug 500 mg") trtcd = 3;
    otherwise put "WAR" "NING: Check trt grp: "
      trtgrp;
end;
run;
    
```

SAS LOG:
WARNING: Check trt grp: TRTGRP=SomeOtherDrug 25 mg

5

SUGME
April 17
2009

*Conditional warnings to check your data
assumptions: Example 2*

```

data new;
  set old;
  where labtest = 'Glucose, fasting' and visitcd='S';
  if resu ^= 'MG/DL' then
    put "WAR" "NING: Need to convert units: "
      usubjid= visitdt= resn= resu=;
  label resn = 'Baseline Fasting Glucose (mg/dL)';
run;
    
```

SAS LOG:
WARNING: Need to convert units: USUBJID=12345
VISITDT=06OCT2007 RESN=6.0 RESU=MMOL/L

6

SUGME April 17 2009

Recommendation 2:
Display actual date and time on output.
(with an extremely brief introduction to macro variables and macro routines)

7

SUGME April 17 2009

Date, time and page number on output in interactive SAS sessions (default vs. desired)

SURVEY 07:52 Monday, September 25, 2000 2057

trt	Frequency	Percent
Placebo	162	71.05
Miracle Drug	66	28.95

SURVEY 1

trt	Frequency	Percent
Placebo	162	71.05
Miracle Drug	66	28.95

25SEP00 at 01:33:12 PM

8

SUGME April 17 2009

Extremely brief introduction to macro variables (symbol tables)

- Macro variables
 - Symbol replacement
 - Character strings (no numeric computation)
- Timing
 - Values assigned before data processing
 - %let in open code
 - %window user input
 - or
 - Data-driven values
 - call symput (DATA step)

Macro Variable	Replacement Text
&realdt	14Apr2009
&realtm	14:52
&whereclause	loanamt ge 25000
&drgcost	1.34
&pboN	162
&trtN	166
&z	2 + 3

9

SUGME April 17 2009

Updating date and time on output (one time)

```
data _null_;
  call symput("Realdt",put(today(),date9.));
  call symput("Realtm",put(time(),time.));
run;
footnote1 "Report run on &Realdt at &Realtm";
options pageno=1;
```

Footnote with resolved macro variables:
 footnote1 "Report run on 12Apr2009 at 14:25"

- DATA step functions to access system date and time
- MACRO variables to store values

10

SUGME April 17 2009

Extremely brief introduction to macro routines (stored procedures)

- Reusable code (DATA steps and procedures, macro syntax, etc.)
- Can have input parameters to control processing
- Defined before data processing
- Can be stored in a permanent catalog
- Called (executed) as needed

```
%macro mymacro (parm1=);
  <macro code here ... data steps and procedures>;
%mend mymacro;

%mymacro(parm1=somevalue);
```

11

SUGME April 17 2009

Updating date and time on output (macro routine)

```
%macro realdt;
data _null_;
  call symput("Realdt",put(today(),date9.));
  call symput("Realtm",put(time(),time.));
run;
footnote1 "Report run on &Realdt at &Realtm";
options pageno=1;
%mend realdt;

%realdt;
```

alternative:
 options dtreset;
 -Automated
 -But no macro vars for custom titles

12

SUGME
April 17
2009

Recommendation 3:
Use macro variables to guarantee internal validity.

13

SUGME
April 17
2009

Using macro variables to guarantee internal validity (de-magic numbering)

```
%let drgcost=1.34;
data work.costing;
set trial.dat.validity;
select (drugcode);
when (0) drugcost = 0;
when (1) drugcost = no_doses * &drgcost;
end;
label drugcost =
"(Cost of Miracle Drug at $&drgcost./pill)";
run;
proc means data=work.costing n mean;
var drugcost;
class drugcode;
run;
```

14

SUGME
April 17
2009

Using macro variables to guarantee internal validity (de-magic numbering)

COST OF STUDY MEDICATION
The MEANS Procedure
Analysis Variable : drugcost (Cost of Miracle Drug at \$1.34/pill)

TREATMENT GROUP	N Obs	N	Mean
Placebo	45	45	0
Miracle Drug (10mg)	44	44	24.3940909
Miracle Drug (20mg)	44	44	48.9100000

15

SUGME
April 17
2009

Using macro variables to guarantee internal validity (linking titles to report criteria)

```
%let whereclause = loanamt ge 25000;
proc print data=work.june;
where &whereclause ;
title "June loans with &whereclause";
run; title;
```

16

SUGME
April 17
2009

Recommendation 4:
Never hard-code summary counts (n=xx).
Use data-driven macro variables.

17

SUGME
April 17
2009

Using data-driven macro variables to display summary counts

```
proc freq data=perm.surveys noprint;
tables trt / out=work.smp1;
run;
data _null_;
set work.smp1 end=eof;
retain totN;
if _n_ eq 1 then totN = count;
else totN = totN + count;
select (trt)
when (0) call symputx ('pbn', put(count, comma6.));
when (1) call symputx ('trtN', put(count, comma6.));
end;
if eof then
call symputx ('totN', put(totN, comma6.));
run;
```

18

SUGME
SUGME
April 17
2009

Using data-driven macro variables to display summary counts

```
proc format;
  value grpNf
    0 = "Placebo (n=&pb0N)"
    1 = "Miracle Drug (n=&trtN)";
run;
```

←

←

19

SUGME
SUGME
April 17
2009

Using data-driven macro variables to display summary counts

How do you rate Miracle Drug overall?			
POOR/ FAIR	GOOD	V GOOD /XNT	
Pct	Pct	Pct	
MEDICATION			
Placebo (n=162)	45.6	26.6	27.8
Miracle Drug (n=166)	31.8	22.7	45.5

20

SUGME
SUGME
April 17
2009

Recommendation 5:
Hand-key input date no more than once!

21

SUGME
SUGME
April 17
2009

Report code requiring dates keyed in multiple places and formats

MIRACLE DRUG SALES FOR August 2000	
REGION	SALES
Midwest	\$189,500
Northeast	\$217,000
Southern	\$741,500
All	\$1,485,200

where date between "01aug2000"D and "31aug2000"D;
Title1 "MIRACLE DRUG SALES FOR August 2000";

22

SUGME
SUGME
April 17
2009

Automation of date criteria/display using SAS functions

```
data _null;
  begdt = intnx('MONTH',today(),-1,'beg');
  enddt = intnx('MONTH',today(),-1,'end');
  call symput('begdt',put(begdt,date9.));
  call symput('enddt',put(enddt,date9.));
  monthnm = put(begdt,monname12.);
  yr = put(year(begdt),4.);
  length titledt $ 16;
  titledt = compbl(monthnm || ' ' || yr);
  call symput('titledt',titledt);
run;
```

23

SUGME
SUGME
April 17
2009

Recommendation 6:
Use meaningful names for binary variables.

24

SUGME
April 17 2009

Unclear output without naming convention

The REG Procedure
Model: MODEL1
Dependent Variable: purchases (YTD Purchases at Pro Shop)

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	12.84211	0.93008	13.81	<.0001
gender	1	3.08097	1.45922	2.11	0.0432

25

SUGME
April 17 2009

Binary coding convention

Binary Notation:
1 = Yes
0 = No

26

SUGME
April 17 2009

*Naming convention for binary variables:
Name of variable specifies direction*

The REG Procedure
Model: MODEL1
Dependent Variable: purchases (YTD Purchases at Pro Shop)

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	12.84211	0.93008	13.81	<.0001
maleflg	1	3.08097	1.45922	2.11	0.0432

27

SUGME
April 17 2009

Recommendation 7:
Leverage the colon wildcard for variable lists.

28

SUGME
April 17 2009

Variable prefixes suitable for wildcard

TTE "Time to Event Data" (hazard analysis)							
pat	trtgrp	fl_si	dur_si	fl_sa	dur_sa	fl_dth	dur_dth
001	New SSRI	0	370	0	370	0	370
002	Placebo	1	17	0	345	0	345
003	New SSRI	1	65	1	98	0	102
004	New SSRI	0	250	0	250	0	250
005	Placebo	0	107	1	107	1	107

Prefix Naming Conventions:

- fl_ = flag for event (0=no event, 1=event)
- dur_ = time to event or censoring (days)

Suffix Naming Conventions:

- si = suicidal ideation
- sa = suicide attempt
- dth = death (all cause)

- ✓ One pair of variables per endpoint
- ✓ numerous endpoints

29

SUGME
April 17 2009

*Use of colon wildcard in code
(flexibility to add/remove variables)*

```

data timetoevent;
  set o1d;

  array ff(*) flg_;;
  array dd(*) dur_;;
  do i = 1 to dim(ff);
    if ff[i] = . then ff[i] = 0;
    if ff[i] = 0 then dd[i] = TimeOnStudy;
  end;

  format flg_ ynfmt.;
run;
    
```

30

SUGME
April 17 2009

Recommendation 8:
Verify your assumptions about the primary key.

31

SUGME
April 17 2009

Database primary keys

Doses		PatID	Date	Time	Dosage
PK	FK1	PatID			
PK		Date			
PK		Time			
		Dosage			

PatID	Date	Time	Dosage
0001	01Feb2002	8:00 AM	10
0001	01Feb2002	8:00 PM	20
0001	02Feb2002	8:00 AM	10
0001	02Feb2002	8:00 PM	20

Primary key (PK)

- Uniquely identifies each row
- Can be one or several variables

32

SUGME
April 17 2009

Trying to identify the "primary key":
Example

PatID	Date	Lab Code	Lab ID	Result
90001	26Oct2006	2000	Sodium	140 mEq/L
90001	26Oct2006	2001	Potassium	4.2 mEq/L
90001	26Oct2006	2003	C-Reactive Protein	6.0 mg/L
90001	26Oct2006	2004	Blood Urea Nitrogen	8.7 mg/dL
90001	26Oct2006	2005	Creatinine	0.9 mg/dL

- ✓ Primary key appears to be PatID-Date-LabCode
- ✓ Naming of LabCode / LabID implies a 1:1 relationship (denormalized lookup data)

33

SUGME
April 17 2009

Trying to identify the "primary key":
Example... Whoops!

PatID	Date	Lab Code	Lab ID	Result
90001	26Oct2006	2000	Sodium	140 mEq/L
90001	26Oct2006	2000	Oral Glucose Tolerance Test	4.2 mEq/L
90001	26Oct2006	2000	LabCode = 2020	6.0 mg/L
90001	26Oct2006	2000		8.7 mg/dL
90001	26Oct2006	2005	Creatinine	0.9 mg/dL
90001	26Oct2006	2020	Glucose, 0 minutes	104 mg/dL
90001	26Oct2006	2020	Glucose, 30 minutes	130 mg/dL
90001	26Oct2006	2020	Glucose, 60 minutes	153 mg/dL
90001	26Oct2006	2020	Glucose, 90 minutes	121 mg/dL
90001	26Oct2006	2020	Glucose, 120 minutes	100 mg/dL

34

SUGME
April 17 2009

Conditional warning to test your assumption about what is a primary key

```
proc sql;
  create table ulabcd as
  select distinct labcode, labid from old;
quit;

data _null_;
  set ulabcd;
  by labcode labid;
  if sum(first.labcode, last.labcode) ^= 2 then
  put "WAR" "NING: MULT ID/LABCODE: " labcode= labid=;
run;
```

LOG:

```
WARNING: MULT ID/LABCODE: LABCODE=2020 LABID=Glucose, 0 minutes
WARNING: MULT ID/LABCODE: USUBJID=2020 LABID=Glucose, 30 minutes
WARNING: MULT ID/LABCODE: USUBJID=2020 LABID=Glucose, 60 minutes
WARNING: MULT ID/LABCODE: USUBJID=2020 LABID=Glucose, 90 minutes
WARNING: MULT ID/LABCODE: USUBJID=2020 LABID=Glucose, 120 minutes
```

35

SUGME
April 17 2009

Recommendation 9:
Keep only meaningful variables when you change the unit of analysis.

36

SUGME
April 17
2009

*Examples of "unit of analysis"
(multiple UofA levels within a single study)*

Clinical Trial Laboratory Parameters

Hematology Summary Tables
Unit of Analysis: Patient-Parameter-Visit
Analysis Variables: White blood cell (WBC) count

Hematology Shift Tables
Unit of Analysis: Patient-Parameter
Analysis Variables: baseline WBC category, end-of-trt WBC category

Summary of Clinically Significant Hematology Parameters
Unit of Analysis: Patient
Analysis Variables: Flag for patients who experienced any clinically significant hematology values during treatment

37

SUGME
April 17
2009

*Keeping only meaningful variables with
"proc sort nodupkey"*

```
proc sort data= labs
  out = labcodes (keep=labcode)
  nodupkey;
  by labcode;
run;
```

↑ ↑

Keep *only* the BY variables
Forces awareness

Corollary...
In a DATA step with subsetting OUTPUT (e.g., "if last.patno then output;"),
keep only
✓ BY variables
✓ RETAIN variables (with explicit summarization logic)

38

SUGME
April 17
2009

*"proc sort nodupkey" without the convention
(incorrect data; no useful LOG messages)*

```
252 proc sort data = labs
253   out = labcodes (keep = labcode labid)
254   nodupkey;
255   by labcode;
256 run;
```

NOTE: There were 10 observations read from the data set WORK.LABS
NOTE: 4 observations with duplicate key values were deleted.
NOTE: The data set WORK.LABCODES has 6 observations and 2 variables.

labcode	labid
2000	Sodium
2001	Potassium
2003	C-Reactive Protein
2004	Blood Urea Nitrogen
2005	Creatinine
2020	Glucose, 0 minutes

← Lookup data is incorrect (semi-random labid selected)

39

SUGME
April 17
2009

*proc sql – no convention necessary
(WARNING message desired!)*

```
257 proc sql;
258   create table labcodes as
259   select distinct labcode
      , labid
260   from labs;
261
262 NOTE: Table WORK.LABCODES created,
263 with 10 rows and 2 columns.
264
265 proc sql;
266   create table labcodes as
267   select labcode, labid
268   from labs
269   group by labcode;
```

labcode	labid
2000	Sodium
2001	Potassium
2003	C-Reactive Protein
2004	Blood Urea Nitrogen
2005	Creatinine
2020	Glucose, 0 minutes
2020	Glucose, 120 minutes
2020	Glucose, 30 minutes
2020	Glucose, 60 minutes
2020	Glucose, 90 minutes

WARNING: A GROUP BY clause has been transformed into an ORDER BY clause because neither the SELECT clause nor the optional HAVING clause of the associated table-expression referenced a summary function.

NOTE: Table WORK.LABCODES created, with 10 rows and 2 columns.

40

SUGME
April 17
2009

Recommendation 10:
Don't reuse variable names.

41

SUGME
April 17
2009

*Just one of many reasons to avoid reusing
variable names*

```
data new;
  set old;
  lastvisdt = coalesce(lastvisdt, termtd);
run;
```

- Suppose old.lastvisdt does not exist.
- No UNINITIALIZED message!

42

SUGME
SUGME
April 17
2009


**Recommendation 11:
Don't reinvent the wheel
(i.e., be cautious with RETAIN)...**

43


SUGME
SUGME
April 17
2009

Many thanks

My Favorite Technical
Writer:
Lexi Hayden



SUGME (travel):
Dave Briggs
Alan Dickson
Craig Dickstein
Judy Loren
and all others from the
audience who
offered comments!



44

SUGME
SUGME
April 17
2009


About the Speaker

Speaker Vanessa C. Hayden
Independent Consultant

Address 377 Metropolitan Avenue
Roslindale, MA 02131

Telephone 617.333.9441
Cell 617.512.2455

E-Mail vanessa.hayden (at gmail)



45

SUGME
SUGME
April 17
2009

Questions or Comments?

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

46