

Dr.Thevaki Mahadevan



A Symbol of Excellence



Consistent Data Delivery- Introduction

- Inconsistencies in the raw data, between data delivery
- Data issues during mapping or transfer process
- Biostatistics best approaches
 - Manually checks all issues, including subject counts in key datasets
 - Write SAS code to check each delivery data
- The advantages of programmatic checks
 - more reliable
 - minimizes human error
 - less time consuming in the long run.
- A program can bulk check all raw datasets, variables and observations that are delivered each time.



Consistent Data Delivery- Typical findings

- External data issue
- Reference ranges changed
- Conversion formulas changed
- Code/decode values changed
- Datasets not delivered
- Missing variables/datasets.
- Extra variables
- Extra datasets



Consistent Data Delivery- Processes and Solutions

- Programmers concentrate on current data issues.
- Very difficult to spot these issues manually by stat/lead reviews.
- Check the data differences using programs.
- When datasets are saved in different folders, i.e. libraries, they can be referenced by libnames.

```
/* Location of Old DM data with the new delivered DM data*/
libname DMOLD "/Volumes/app/sites/project/icon_derm/protocol/data/data_old"; /*Old RAW data*/
libname DMNEW "/Volumes/app/sites/project/icon_derm/protocol/data/data_old"; /*New RAW data*/
```

FWTABLE: Sashelp.vcolumn

File View Tools Data Solutions Help

LIBNAME	MEMNAME	MEMTYPE	NAME	TYPE	LENGT	NPOS	VARNUM	LABEL	FORMAT	INFORMAT
DMNEW	ADVERSE	DATA	AESITE	num	8	0	1	AE Site Category	BODSITE.	
DMNEW	ADVERSE	DATA	AECLAS	num	8	8	2	AE Class	AECLS.	
DMNEW	ADVERSE	DATA	COLLDATF	char	80	248	3	Char Collection Date of CRF Page		
DMNEW	ADVERSE	DATA	FROMDATF	char	80	328	4	Char Adverse Event From Date - Partial		
DMNEW	ADVERSE	DATA	TODATEF	char	80	408	5	Char Adverse Event To Date - Partial		
DMNEW	ADVERSE	DATA	ACCESSTS	num	8	16	6	Accessible TS	DATETIME.	DATETIME20.
DMNEW	ADVERSE	DATA	LOGINTS	num	8	24	7	Login TS	DATETIME.	DATETIME20.
DMNEW	ADVERSE	DATA	LSTCHGTS	num	8	32	8	LstChg Ts	DATETIME.	DATETIME20.
DMNEW	ADVERSE	DATA	LOCKFLAG	char	1	488	9	Data Lock Flag	\$1.	\$1.
DMNEW	ADVERSE	DATA	RDCHID	num	8	40	10	Received DCI Id	10.	11.
DMNEW	ADVERSE	DATA	DOCNUM	char	20	489	11	Document Number	\$20.	\$20.
DMNEW	ADVERSE	DATA	DCMNAME	char	16	509	12	DCM Name	\$16.	\$16.

Figure 1: Sample sashelp.vcolumn dataset



A Symbol of Excellence

Consistent Data Delivery- Meta data and dataset Differences

A merge on datasets

```
/*Get list of datasets and variables names from both libnames by accessing Dictionaries*/  
data ardmv;  
  set sashelp.vcolumn (keep=libname memname name length label type format informat);  
  where libname in ('DMOLD','DMNEW');  
run;  
  
/*Check which datasets are in both library and if missing from one area*/  
proc sort data=ardmv out=dmo(drop=libname) nodupkey; where libname='DMOLD'; by memname; run;  
proc sort data=ardmv out=dnn(drop=libname) nodupkey; where libname='DMNEW'; by memname; run;  
  
data both data(keep=memname) inold_data innew_data ;  
  merge dmo(in=a) dnn(in=b);  
  by memname;  
  if a and b then output both_data;  
  if a and not b then output inold_data;  
  if b and not a then output innew_data;  
run;
```

MEMNAME	IN
ALLERGY	inold_dat
LAB_SDW	inold_dat
LB	inold_dat
NRS	inold_dat
OMIC	inold_dat

Figure 2: Example of output showing datasets only in old data



A Symbol of Excellence



Consistent Data Delivery- Meta data and variable Differences



A merge on datasets and variables

```
/*Check which datasets and variables are in both library and if missing from one area*/
proc sort data=ardmv out=dmov(drop=libname); where libname='DMOLD'; by memname name; run;
proc sort data=ardmv out=dmnv(drop=libname); where libname='DMNEW'; by memname name; run;

data both var inold var innw var;
merge dmov(in=a) dmnv(in=b);
by memname name;
if a and b then output both var;
if a and not b then output inold var;
if b and not a then output innw var;
run;
```

MEMNAME	NAME	TYPE	LENGTH	LABEL	FORMAT	INFORMAT	IN
ADVERSE	DBSSITEF	char	200	Body Site Involvement-FUL	\$200.	\$200.	inold var
ADVERSE	DBSSITEL	char	200	Body Site Involvement-DLV	\$200.	\$200.	inold var
ADVERSE	DBSSITEN	num	8	Body Site Involvement-DVN	10.		inold var

Figure 3: Example of output showing only variables in old dataset

A macro variable referencing to call common dataset

```
/*
Proc compare DM old with DM new
/ Create macro variables for each common datasets found in both library and for total number /
of datasets found.
*/
proc sort data=both_data out= both_dat1 nodupkey; by memname; run;

data _null_;
set both_dat1 end=last;
if last then call symput('tot1',_n_);
call symput('DSET1'||left(_N_), trim(left(memname)));
run;
```



Consistent Data Delivery- Bulk Proc compare macro

- All numeric variables selected as by variable
- Conditional key by variable
- Conditional where statement

BULK RUN OF PROC COMPARE

```
%macro comp;
  %do i= 1 %to %tot;
    data dbvvar(keep=memname byvar byvar2 none1);
      set both_var;
      retain byvar byvar2 none1;
      format byvar2 $18. byvar $900. none1 $5.;
      by memname;
      if strip(type) eq 'num' then do;
        if first.memname then byvar=name;
        else byvar=catx(' ',byvar.name);
      end;
      if first.memname then byvar2='';
      if name eq 'SUBJID' then byvar2='study inv subjid';
      if first.memname then none1='';
      if name eq 'NONE1' then none1='none1';
      if last.memname then output;
    run;

    data _null_;
      set dbvvar;
      if _n_;
        if strip(memname) eq "%dsset%i" then call symput ('bvar' || left(_n_), trim(left(byvar)));
        if strip(memname) eq "%dsset%i" then call symput ('bvar2' || left(_n_), trim(left(byvar2)));
        if strip(memname) eq "%dsset%i" then call symput ('non' || left(_n_), trim(left(none1)));
      run;
```



A Symbol of Excellence



Consistent Data Delivery- Bulk Proc compare macro

- Datasets sorted before proc compare
- Proc means kept within the macro

```
proc sort data=dmold.&&dset&i out=o&&dset&i;  
  by %if %index(&sbvar2&i,subjid) %then %do; &sbvar2&i &sbvars&i; %end;  
     %else %do;&sbvars&i; %end;  
     %if %index(&nons&i,nonel) %then %do;  
       where nonel ne 'NONE'; %end;  
run;  
  
proc sort data=dmnew.&&dset&i out=n&&dset&i;  
  %if %index(&nons&i,nonel) %then %do;  
    where nonel ne 'NONE';  
  %end;  
  by %if %index(&sbvar2&i,subjid) %then %do; &sbvar2&i &sbvars&i; %end;  
     %else %do;&sbvars&i; %end;  
run;  
  
proc compare data=o&&dset&i compare=n&&dset&i listall novalues out=&&dset&i outnoequal ;  
  id %if %index(&sbvar2&i,subjid) %then %do; &sbvar2&i &sbvars&i; %end;  
     %else %do;&sbvars&i; %end;  
run;  
  
proc means data = n&&dset&i n nmiss;  
run;  
  
%end;  
%mend;  
%comp;
```



A Symbol of Excellence



Consistent Data Delivery- Proc compare output

- A top level dataset summary with total number of obs and vars present in the two datasets.
- The variable summary with variables differences found between the two datasets.

The COMPARE Procedure
Comparison of WORK.0ADVERSE with WORK.NADVERSE
(Method=EXACT)

Data Set Summary

Dataset	Created	Modified	NVar	NObs
WORK.0ADVERSE	21AUG14:15:06:32	21AUG14:15:06:32	92	1829
WORK.NADVERSE	21AUG14:15:06:33	21AUG14:15:06:33	93	2038

Variables Summary

Number of Variables in Common: 89.
Number of Variables in WORK.0ADVERSE but not in WORK.NADVERSE: 3.
Number of Variables in WORK.NADVERSE but not in WORK.0ADVERSE: 4.
Number of Variables with Differing Attributes: 1.
Number of ID Variables: 29.

Listing of Variables in WORK.0ADVERSE but not in WORK.NADVERSE

Variable	Type	Length	Format	Informat	Label
DBSSITEN	Num	8	10.		Body Site Involvement-DUN
DBSSITEL	Char	200	\$200.	\$200.	Body Site Involvement-DLV
DBSSITEF	Char	200	\$200.	\$200.	Body Site Involvement-FUL

Listing of Variables in WORK.NADVERSE but not in WORK.0ADVERSE

Variable	Type	Length	Format	Informat	Label
AESITE	Num	8	BODSITE.		AE Site Category
AESITEN	Num	8	10.		Body Site Involvement-DUN
AESITEL	Char	200	\$200.	\$200.	Body Site Involvement-DLV
AESITEF	Char	200	\$200.	\$200.	Body Site Involvement-FUL

Consistent Data Delivery- Proc compare output

- The observation summary with number of observation differences found between the two datasets.
- Any significant drop in observation count from the old data is of potential concern.

Observation Summary

```
Number of Observations in Common: 1702.  
Number of Observations in WORK.0ADVERSE but not in WORK.NADVERSE: 127.  
Number of Observations in WORK.NADVERSE but not in WORK.0ADVERSE: 336.  
Total Number of Observations Read from WORK.0ADVERSE: 1829.  
Total Number of Observations Read from WORK.NADVERSE: 2038.  
  
Number of Observations with Some Compared Variables Unequal: 8.  
Number of Observations with All Compared Variables Equal: 1694.
```

- The value comparison summary is of much less value at this stage and can be skimmed through.

Values Comparison Summary

```
Number of Variables Compared with All Observations Equal: 58.  
Number of Variables Compared with Some Observations Unequal: 2.  
Number of Variables with Missing Value Differences: 1.  
Total Number of Values which Compare Unequal: 8.
```

Variables with Unequal Values

Variable	Type	Len1	Len2	Label	Ndif	MaxDif	MissDif
AEACTT	CHAR	80	200	Action Treatment Other Text	5		0
AEDECD1	CHAR	100	100	Lowest Level Term (LLT)	2		2



A Symbol of Excellence

Consistent Data Delivery- Proc compare on selected dataset

Individual detail dataset comparison

```
/*  
/  When only need to run few_dataset.usa below Proc compare  
/  
*/  
%let dataset=adverse;  
%let by=subiid aeterm colldate rdcid.;  
*/  
proc sort data=dmold.&dataset out=o&dataset;  
by &by;  
where nonel ne 'NONE';  
run;  
  
proc sort data=dmnew.&dataset out=n&dataset;  
where nonel ne 'NONE';  
by &by;  
run;  
  
proc compare data=o&dataset compare=n&dataset listbase out=a&dataset outnoequal;  
id &by;  
run;
```

Further merge with tailored key by variables

```
data both_obs inold_obs innnew_obs;  
merge o&dataset(in=a) n&dataset(in=b);  
by &by;  
if a and b then output both_obs;  
if a and not b then output inold_obs;  
if b and not a then output innnew_obs;  
run;
```

Subject ID	Lowest Level Term (LLT)	AE Investigator Text - Raw - MedDRA	Adverse Event From Date	Adverse Event To Date	IN
0011014		ARM ABRASIONS	09JUL2014	.	inold_obs
0011014	Abrasions	LEFT ARM ABRASIONS	09JUL2014	05AUG2014	innnew_obs

Figure 4: Example of output showing the observation differences between the old and new data

Consistent Data Delivery-Missing values

- Missing and non-missing values checked by using proc means.
- Any high number of missing values of concern can be identified this way.

The SAS System

The MEANS Procedure

Variable	Label	N	N Miss
AESITE	AE Site Category	48	1990
AECLAS	AE Class	2038	0
ACCESSTS	Accessible TS	2038	0
LOGINTS	Login TS	2038	0
LSTCHGTS	LstChg Ts	2038	0
RDCIID	Received DCI Id	2038	0
SUBSETSM	DCM Subset Number	2038	0
QUALIFYQ	Qualifying Question	1990	48

- Spec on required variables can be crossed checked for missing required variables



A Symbol of Excellence



Consistent Data Delivery-Summary

This programmatic approach will help us to identify most of the differences, including:

- any differences in the variables or datasets
- significant drop in number of observations
- any common patterns seen across the dataset
- concern level of missing observations
- missing variables or datasets

Once these are checked and are accounted for, one can be **more confident** in the **consistency of the data** being delivered.

Consistent Data Delivery

Questions?



ICON

A Symbol of Excellence

