

Go Compare!

Flagging up some under-used options
in PROC COMPARE

Michael Auld
PhUSE Brighton 2011

Coming up...

- Syntax and Report output
- ODS Output tables
- Compare of Compare
- Using OUT= to derive flags

The COMPARE Procedure
 Comparison of ADSDEV.ADCM with ADS.ADCM
 (Method=EXACT)

DATA SET SUMMARY

Data Set Summary					
Dataset	Created	Modified	Num Obs	Num Variables	Label
ADSDEV.ADCM	15APR10:09:39:14	15APR10:09:39:14	39	564	Concomitant Medications
ADS.ADCM	15APR10:15:11:48	15APR10:15:11:48	39	564	

Variables Summary

Number of Variables in Common: 39.
 Number of Variables with Differing Attributes: 13.

VARIABLE SUMMARY

Listing of Common Variables with Differing Attributes						
Variable	Dataset	Type	Length	Format	Informat	Label
STUDYID	ADSDEV.ADCM	Char	15	\$15.		Study Identifier
	ADS.ADCM	Char	15	\$15.	\$12.	Study Identifier

Observation Summary

OBSERVATION

Observation	Base	Compare
First Obs	1	1
First Unequal	508	508
Last Unequal	558	558
Last Obs	564	564

Number of Observations in Common: 564.
 Total Number of Observations Read from ADSDEV.ADCM: 564.
 Total Number of Observations Read from ADS.ADCM: 564.
 Number of Observations with Some Compared Variables Unequal: 0.
 Number of Observations with All Compared Variables Equal: 560.

SUMMARY

Variables with Unequal Values

Variable	Type	Len	Label	Ndif	MaxDif	MissDif
CMTYPE	CHAR	1	Medication Type	2		0
CMTYPEN	NUM	8	Medication Type, Numeric	2	1.000	0
CMDOSE	CHAR	200	Dose per Administration	1		0
CMOBSFL	CHAR	2	Changes to Listing	1		1

Value Comparison Results for Variables

		Medication Type	
		Base Value	Compare Value
Obs		CMTYPE	CMTYPE
		—	—
508		P	C
535		P	C

DIFFERENCES SUMMARY

4

Syntax

```
PROC COMPARE <options>;  
BY <sortoptions> variables;  
ID <sortoptions> variables;  
VAR variables;  
WITH variables;  
RUN;
```

VAR/WITH

(1) Compare variables with different names without resorting to RENAME data set options:

```
PROC COMPARE DATA=sds.ex COMPARE=ads.adex;  
ID subjid;  
VAR visitnum;  
WITH avisitn;  
RUN;
```

VAR/WITH

Similar to the following (but uglier):

```
PROC COMPARE DATA=sds.ex COMPARE=ads.adex  
(RENAME=(avisitn=visitnum)) ;  
ID subjid;  
VAR visitnum;  
RUN;
```

VAR/WITH

(2) Compare variables with different names in the same data set:

```
PROC COMPARE DATA=ads.adex;  
ID subjid;  
VAR visitnum;  
WITH avisitn;  
RUN;
```

The COMPARE Procedure
 Comparisons of variables in ADS.ADEX
 (Method=EXACT)

All Variables Compared have Unequal Values

Variable	Type	Len	Compare	Len	Label	Compare Label	Ndif	MaxDif
VISITNUM	NUM	8	AVISITN	8	Visit Number	Analysis Timepoint Number	78682	8.000

Value Comparison Results for Variables

SUBJID	Visit Number				
	Analysis Timepoint Number				
	Base	Compare			
	VISITNUM	AVISITN	Diff.	% Diff	
00010004	1.01	0	-1.0100	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	
00010004	-1.00	0	1.0000	-100.0000	

Options

- **MAXPRINT**
 - Default is 50 for each variable
 - Maximum permitted is 32767
- **CRITERION**
 - Default is 0.00001 – worth changing to something less
 - Can eliminate floating point errors
- **LISTx**
 - Includes LISTOBS, LISTVAR, LISTBASEOBS, LISTCOMPVAR, LISTALL
- **TRANSPPOSE**
 - Shows output per observation/ID variable rather than per variable

Comparison Results for Observations

OBS=14:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	-1.00	0	1.000000	-100.000000
OBS=15:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	-1.00	0	1.000000	-100.000000
OBS=16:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	-1.00	0	1.000000	-100.000000
OBS=17:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	-1.00	0	1.000000	-100.000000

Or...

SUBJID=00010004:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	99.00	91.000000	-8.000000	-8.080808
SUBJID=00010004:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	99.00	91.000000	-8.000000	-8.080808
SUBJID=00010004:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	99.00	91.000000	-8.000000	-8.080808
SUBJID=00010004:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	99.00	91.000000	-8.000000	-8.080808
SUBJID=00010004:					
Variable	With	Base Value	Compare	Diff.	% Diff
VISITNUM	AVISITN	99.00	91.000000	-8.000000	-8.080808

ODS Output

- CompareDatasets
- CompareSummary
- CompareDifferences
- CompareVariables
- CompareDetails

The COMPARE Procedure
 Comparison of ADSDEV.ADCM with ADS.ADCM
 (Method=EXACT)

Compare Datasets

Dataset Summary

Dataset	Created	Modified	Number of Observations	Number of Variables	Label
ADSDEV.ADCM	15APR10:09:39:14	15APR10:09:39:14	39	564	Concomitant Medications
ADS.ADCM	15APR10:15:11:48	15APR10:15:11:48	39	564	

Variables Summary

Number of Variables in Common: 39.
 Number of Variables with Differing Attributes: 13.

Compare Variables

Listing of Common Variables with Differing Attributes

Variable	Dataset	Type	Length	Format	Informat	Label
STUDYID	ADSDEV.ADCM	Char	15	\$15.		Study Identifier
	ADS.ADCM	Char	15	\$15.	\$12.	Study Identifier

Observation Summary

Compare Summary

Observation	Base	Compare
First Obs	1	1
First Unequal	508	50
Last Unequal	558	558
Last Obs	564	564

Number of Observations in Common: 564.
 Total Number of Observations Read from ADSDEV.ADCM: 564.
 Total Number of Observations Read from ADS.ADCM: 564.

Number of Observations with Some Compared Variables Unequal: 4.
 Number of Observations with All Compared Variables Equal: 560.

Variables with Unequal Values

Variable	Type	Len	Label	Ndif	MaxDif	MissDif
CMTYPE	CHAR	1	Medication Type	2		0
CMTYPEN	NUM	8	Medication Type, Numeric	2	1.000	0
CMDOSE	CHAR	200	Dose per Administration	1		0
CMOBSFL	CHAR	2	Changes to Listing	1		1

Value Comparison Results for Variables

			Medication Type
			Base Value
Obs		CMTYPE	Compare Value
			CMTYPE
		—	—
508		P	C
535		P	C

CompareDifferences

4

The COMPARE Procedure
Comparison of WORK.MYCM with SDSOLD.CM
(Method=EXACT)

Comparison Results for Observations

Observation 22697 in WORK.MYCM not found in SDSOLD.CM: USUBJID=0086-0009
CMSEQ=49.

Observation 22925 in WORK.MYCM not found in SDSOLD.CM: USUBJID=0088-0004
CMSEQ=17.

Observation 22953 in WORK.MYCM not found in SDSOLD.CM: USUBJID=0088-0009
CMSEQ=18.

Observation 22961 in WORK.MYCM not found in SDSOLD.CM: USUBJID=0088-0010
CMSEQ=8.

Observation 23055 in WORK.MYCM not found in SDSOLD.CM: USUBJID=0088-0018
CMSEQ=16.

CompareDetails

ODS TRACE ON

- Datasets listed don't come out in the order in the report
- ODS Data set only generated when used in the report
 - eg if duplicate ID variables appear then CompareDifferences section will be dropped

```
Output Added:
-----
Name:      CompareDatasets
Label:     Datasets
Data Name: BatchOutput
Path:      Compare.CompareDatasets
-----
```

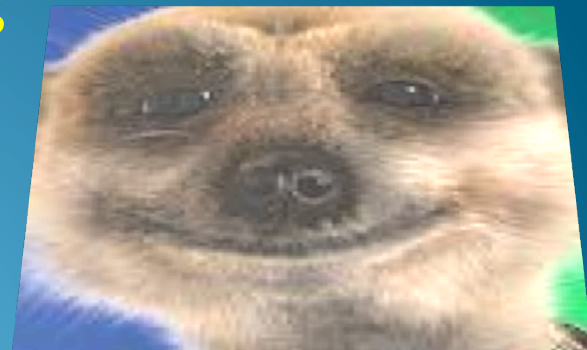
Compare the Market...

- INDEXES

- Works with BY statement
- Has no effect with ID statement – data set needs sort or NOTSORTED option applied

- FORMATS

- Unlike SUMMARY/MEANS, formats applied don't get used by COMPARE – the variables are output without the formatting



OUT= options

- OUTBASE
 - Writes observation to the output dataset for every obs in BASE
- OUTCOMP
 - likewise for every obs in COMP data set
- OUTDIF
 - writes observation with the differences between COMP and BASE
- OUTPERCENT
 - like OUTDIF but % difference
- OUTALL
 - equivalent of all the above
- OUTNOEQUAL
 - Only writes the obs when there is a difference

	TYPE	_OBS_	CMSEQ	CMSTDTC	CMSTDY
New observations	COMPARE	647	30	2009-05-06	197
	COMPARE	648	31	2011-01-12	813
	BASE	647	32	2010-10-06	715
No change	COMPARE	649	32	2010-10-06	715
	DIF	647	32		0
	BASE	648	33	2009-05-06	197
Char: pad with .	COMPARE	650	33	2011-01-12	813
	DIF	648	33	..XX..X.XX.....	616
	BASE	649	34	2009-05-06	197
Num: zero	COMPARE	651	34	2009-05-06	197
	DIF	649	34		0
	BASE	650	35	2010-04-21	547
Differences (Char):	COMPARE	652	35	2009-05-06	547
	DIF	650	35	..XX..X.XX.....	0
	BASE	651	36	2009-05-06	197
X indicates change	COMPARE	653	36	2010-04-21	547
	DIF	651	36	..XX..X.XX.....	350
	BASE	652	37	2010-03-17	512
at that position	COMPARE	654	37	2009-05-06	197
	DIF	652	37	..XX..X.XX.....	-315
	BASE	653	38	2009-05-06	197
Differences (Num):	COMPARE	655	38	2009-05-06	512
	DIF	653	38		315
	BASE	654	39	2010-03-24	519
Arithmetic difference	COMPARE	656	39	2009-05-06	197
	DIF	654	39	..XX..X.XX.....	-322
	BASE	655	40	2010-05-26	582
stored in the DIF	COMPARE	657	40	2010-03-24	519
	DIF	655	40	X..X.....	-63
observation					

Flagging the changes

- a case study

- Study report submitted to FDA, but required 120 day safety update as part of New Drug Application
 - Changes to data needed to be shown in the update
- Live database – could not roll back changes
- Database required a data cut on both occasions
 - eg, if an AE ended after the date of data cut-off, this needed amending to ongoing

COMPARE to the rescue!

- Generate SDTM for each variation of data slice and data cut
- Generate Supplementary domains if required
- Apply relevant data cut (if required) to parent and supplementary domains
- Compare each variation of generated SDTM data and create flags from the results of that comparison
- Add flags to the previously generated SUPP domain. If it didn't exist then create one.

```

*-----*
* Program Name : S_AE.sas *
* Program Type : SDS *
* Author : xxxx *
* Date : 29JUL2009 *
*-----*
* DESCRIPTION: Construct AE SDTM *
*-----*
* INPUT: *
* raw.AEC *
* OUTPUT: *
* SDS.AE, SDS.SUPPAE *
*-----*
* Modification Log *
*-----*

%let proiname=S_AE;
%logfile(switch=ON);
%m_clear;

* read raw data from RAW.AEC;
proc sql noprint;
  create table AE1 as select
    AEC.STUDYID as STUDYID
  ,
    "AE" as DOMAIN
  ,
    compress(AEC.STUDYID) || "-" || put(AEC.SUBJID,z8.) as USUBJID
  ,
    left(put(AEC.AESPID,best.)) as AESPID
  ,
    AEC.AESPID as SORTORD
  ,
    AEC.AETERM as AETERM
  ,
    AEC.ASCODED as AEMODIFY

```

Step 1: Convert S_XX code to a makeXX macro:

BEFORE

```

*-----*
* Program Name : makeAE.sas *
* Program Type : SDS *
* Author : xxxx *
* Date : 29JUL2009 *
*-----*
* DESCRIPTION: Construct AE SDTM *
*-----*
* INPUT: *
* raw.AEC *
* OUTPUT: *
* SDS.AE, SDS.SUPPAE *
*-----*
* Modification Log *
* 15FEB2010 MA Conversion of main domain creation to macro *
* 04MAR2010 MA Correction to LLT label *
* 25MAR2010 MA Correction to AEOUT: Not resolved->Not Resolved *
*-----*

%macro makeAE(rawlib = raw
              ,outds = sds.AE
              ,suppds = SUPPAE1
              );

* read raw data from RAW.AEC;
proc sql noprint;
  create table AE1 as select
    AEC.STUDYID as STUDYID
  ,
    "AE" as DOMAIN
  ,
    compress(AEC.STUDYID) || "-" || put(AEC.SUBJID,z8.) as USUBJID
  ,
    left(put(AEC.AESPID,best.)) as AESPID
  ,
    AEC.AESPID as SORTORD
  ,
    AEC.AETERM as AETERM
  ,
    AEC.ASCODED as AEMODIFY
  ,

```

Step 1: Convert S_XX code to a makeXX macro:

AFTER

```

* Program Name : S_AE.sas *;
* Program Type : SDS *;
* Author : xxxxx *;
* Date : 01MAR2010 *;
*-----*
* DESCRIPTION: Construct AE SDTM *;
*-----*
* INPUT: *;
* call to makeSDTM macro *;
* OUTPUT: *;
* SDS.AE, SDS.SUPPAE *;
*-----*
* Modification Log *;
* M Auld 01MAR2010 Moved original domain creation to macros *;
* This allows for the code to stay neutral from repeated calls*
* but substituting different data cuts and data libraries *;
*-----*

%let proiname=S_AE;
%logfile(switch=ON);
%m_clear;

%makeSDTM(domain = AE
           ,suppvars = LLT AEOUTC
           ,datevar = AESTDTC AEENDTC
           ,suppdatevar =
           ,dayvar = AESTDY AEENDY
           ,compareID = studyid usubjid visit aebodsys aedecod aespids
           );

%logfile(switch=OFF);

```

Step 1: Convert S_XX code to a makeXX macro:

Also need to create the calling program too!

Step 2: Create generic makeSDTM macro:

```
*****;  
* SDTM Creation *;  
* (1) ZZ prefix *;  
* - 4MSU data with no datacut *;  
*****;  
%make&domain;  
%if &numSUPP %then %do;  
    %makeSupp;  
%end;  
*****;  
* SDTM Creation *;  
* (2) no prefixes *;  
* - 4MSU data with 30NOV2009 datacut *;  
* - reading from the ZZprefix SDTMs *;  
*****;  
%datacut_date;  
%if &numSUPP and &domain ne AE %then %do;  
    %makeSupp;  
    %datacut_date;  
%end;  
*****;  
* SDTM Creation *;  
* (3) CC prefix *;  
* - 4MSU data with Cycle 1 datacut *;  
* - reading from the ZZprefix SDTMs *;  
*****;  
  
*****;  
* SDTM Creation *;  
* (4) MM prefix *;  
* - CSR data with 30NOV09 datacut *;  
* - reading from oldClraw libname *;  
*****;
```

Step 2: Create generic makeSDTM macro:

```
*****;  
* Create Observation Flag 1: *;  
* (5) Compare Old data (with C1 cut) *;  
* with new data (with C1 cut) *;  
* Identify changed observations $ *;  
* Identify new observations £ *;  
*****;  
%makeCompare(domain=&domain  
    ,inds=oldclsds.&domain  
    ,compds=sds.CC&domain  
    ,outds=change&domain.1  
    ,idvars=&compareID  
    ,flagvar=obsf11  
    );  
  
*****;  
* Create Observation Flag 2: *;  
* (6) Compare Old data (with 4MSU cut) *;  
* with new data (with 4MSU cut) *;  
* Identify the changed observations $ *;  
* Identify the new observations £ *;  
*****;  
%makeCompare;
```

Etc.

```

*****;
* SDTM Creation *;
* (9) Add flags to final SUPP SDTM *;
* (if it exists) *;
*****;

proc sql noprint;
  create table &domain.flags as
  select &compSqlList,
         &domain..&idvar,
         pre.obsfl1,
         pre.obsfl2,
         pre.obsfl3,
         pre.obsfl4
  from sds.&domain left join pre_&domain.flags as pre
  on &compSqlOn;
quit;

%makeSupp(domain=&domain, idvar=&idvar, inds=&domain.flags, outds=flagsupp&domain, qnam=obsfl1 obsfl2
obsfl3 obsfl4);

%if &numSUPP %then %do;
  proc append base=sds.supp&domain data=flagsupp&domain;
  run;
%end;
%else %do;
  %copyDS(inds=flagsupp&domain, outds=sds.supp&domain);
%end;

```

Step 2: Create generic makeSDTM macro:

```

proc compare data=inds compare=compds out=compset(drop=_obs_)
  noprint outbase outcomp outdif;
  id &idvars;
run;

data dif;
  &attrib _TYPE_ $8;
  label &flagvar="&flaglab";
  dsid1 = open('compset(where=(_TYPE_ eq "DIF"))');
  call set(dsid1);
  numVars = attrn(dsid1, "NVARs");
  numObs = attrn(dsid1, "NOBS");
  do obsloop = 1 to numObs;
    &flagvar = '';
    rc = fetchobs(dsid1, obsloop);
    do varLoop = 1 to numVars;
      _varname_ = upcase(varname(dsid1,varLoop));
      if _varname_ not in (&IDinList '_TYPE_' "&noComp") then select (vartype(dsid1,varLoop));
        when ('C') if index(getvarc(dsid1, varloop), 'X') then do;
          &flagvar = '$';
        end;
        when ('N') if getvarn(dsid1, varloop) not in (., 0) then do;
          &flagvar = '$';
        end;
        otherwise;
      end;
    end;
    if not missing(&flagvar) then output;
  end;
  keep &idvars &flagvar;
run;

```

Step 3: Make Compare macro: Flag the differences with a \$

Step 3: Make Compare macro: Flag the new observations with a £

```
data newobs oldobs;
  &attrib _TYPE_ $8;
  label &flagvar="&flaglab";
  set compset(where=(_TYPE_ ne "DIF"));
  by &IDvars _TYPE_;
  if first.&&id&numID eq last.&&id&numID then do;
    if _TYPE_ eq "COMPARE" then do;
      &flagvar = "£";
      output newobs;
    end;
    else if _TYPE_ eq "BASE" then output oldobs;
  end;
  keep &idvars &flagvar;
run;
```

Step 4: Create the ADaM data from the SDTM and generate Listings with the flags

Listing 16.2.8.3.1
Urinalysis Data
All Patients

Patient ID/ Age(yr), Sex, Race	Visit	Test Performed	Date of Assessment	Study Day	Laboratory Test	Unit	Result	Chg^
10011006/ 53,M,W	CYCLE 2 DAY 1	Yes	2009-06-02	26	Bacteria		ND	
					Casts		ND	
					Crystals		ND	
					Epithelial cells		ND	
					Glucose	Not Applicable	NEG	
					Ketones		NEG	
					Occult blood		ND	
					pH		6	
					Protein	Grams per 24 hours	+/-	
					RBC	Picogram	+	
					Specific gravity	Kilogram per Liter	1.03	
					WBC	Picogram	+	
	STUDY TERMINATION	Yes	2009-06-23	47	Bacteria		NEG	_£
					Casts		NEG	_£
					Crystals		NEG	_£
					Epithelial cells		NEG	_£

(CONTINUED)

^Changes to Listing from the Study Phase Listing are identified as follows; \$_ = changed observation in data,
 _\$ = change as a result of removing original Study Phase cut assumptions,
 \$\$ = changed observation in data and change as a result of removing original Study Phase cut assumptions,
 £_ = New Observation (Study Phase), _£ = New Observation (since Study Phase).

XXXXXXXXXXXXXXXXXXXXXXXXXXXX/4MSU/PROD/pg/Listings/l_uril.sas

FINAL

25MAR2010:18:21

Questions...

