

SESUG Paper 151-2019

HOW AM I DIFFERENT?

Dr. Kannan Deivasigamani, Wellcare – TSS

Abstract

This SESUG paper demonstrates a practical application of how Dictionary tables in SAS® can be used to make a programmer's task easier. While DICTIONARY is rich in meta-data, the example presented will just be one of the many possible ways COLUMNS table can be used to spot and present differences in column names between 2 different datasets easily. This paper is intended for audiences who wish to find the differences between 2 SAS tables/datasets as an alternative to PROC COMPARE to tabulate similarities and commonalities at a variable level. This is commonly used when there is new development or analysis, migration or reengineering of processes take place within organizations.

INTRODUCTION

Programmers often seek to use the procedure from SAS with the hope of finding differences between 2 datasets. One step further, the interest focuses on the variables that are different between the two datasets in question. Fortunately, "PROC COMPARE" provides information about the number of variables that are common between the 2 datasets and also prints the number of variables in the BASE dataset but not in the COMPARE dataset and vice versa. However, there is some inconvenience posed when it comes to listing out which variables are different and show them side by side in a table format. In many occasions, it turns out to be a manual process of eyeballing the fields or transferring the metadata to an Excel spreadsheet and using a VLOOKUP or some other function and then compiling the variables. As an alternative, the tool presented in this paper in the form of a macro can take two SAS dataset names as input and it provides a print of the common columns, differences in the 2 datasets by listing out columns in

BASE but not in COMPARE dataset and vice versa. It would save several minutes of one's time without scrolling through much or rerunning the PROC COMPARE code with LISTVAR, LISTBASEVAR, or LISTCOMPVAR options to get similar information. If this is done on multiple occasions, by multiple programmers, the benefit multiplies. The goal is to provide to make the process convenient just by calling the macro with the 2 dataset names intended for tabulation showing the differences and similarities.

BUSINESS REQUIREMENT

During development of new datasets that are similar but have variations there may often arise a need to find the differences between the two datasets. The differences that we are talking about are not at the data-level. The structural differences, to be more specific, are the ones of interest in this context. There may also be testing effort between two different environments such as "Production" and "QA" where there may be a need to identify the differences. Another situation that can very likely require such an exercise is where there may be a conversion of one system to another that may be better or even an enhancement requiring a comparison. No matter what the situation is, the tool presented in this paper will come in handy.

PRACTICAL EXAMPLE

The tool will accept the names of the two datasets to compare and will provide an output that will print common variables, variables in dataset-1 but not in dataset-2, and variables in dataset-2 but not in dataset-1 in 3 columns side by side. This can be presented to the individual who can address or justify the differences. The tool is a SAS macro that will accept two inputs. There are 2 examples presented here to demonstrate that the tool can be used with normal SAS datasets as well as SAS generation datasets.

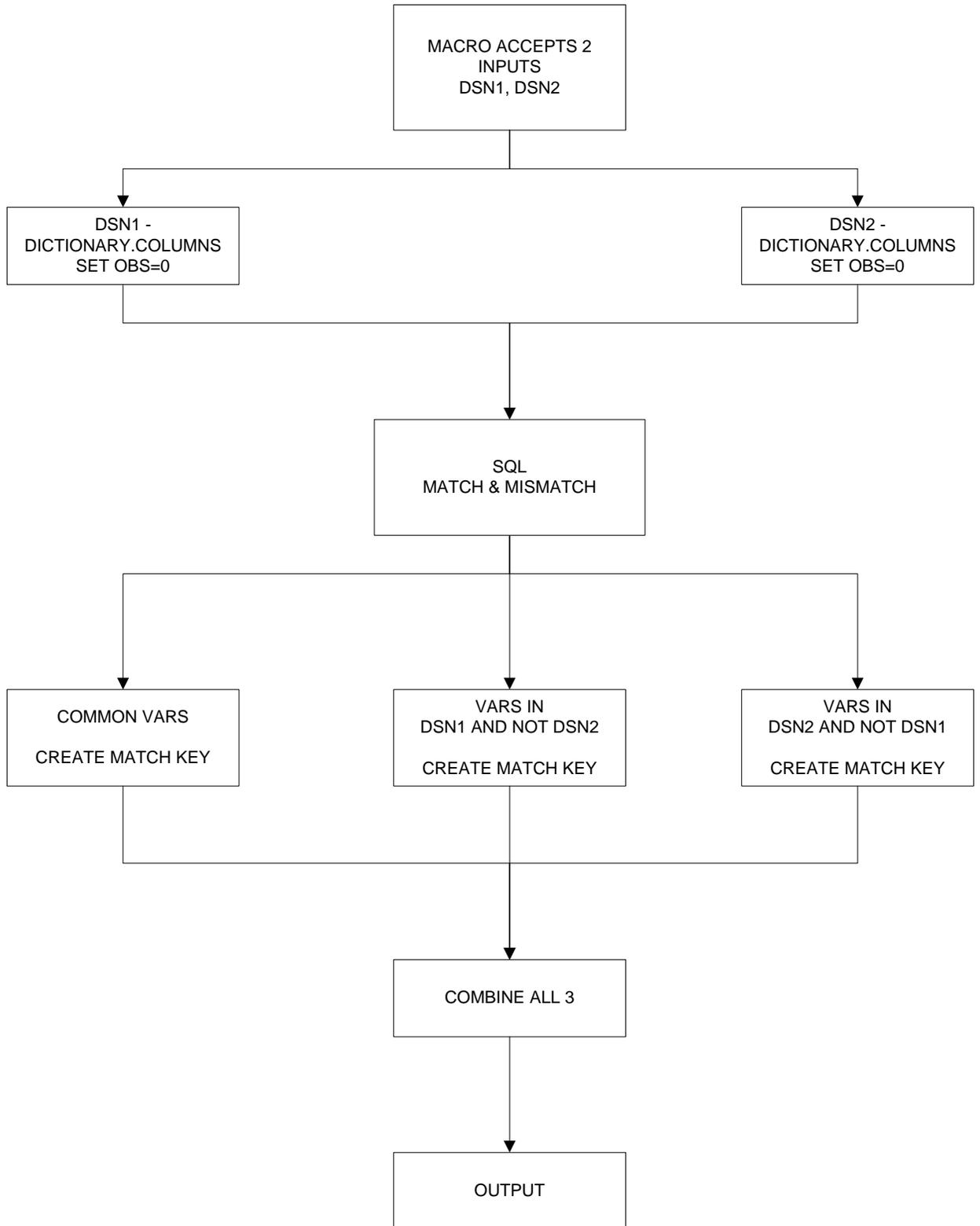
Bennett (2001) and Lee (2012) adopted an alternate way of capturing meta-data with use of the procedure commonly used by SAS programmers, "PROC CONTENTS".

```
1 proc contents data=sashelp.class noprint
2 out=out1(keep=name type format label length engine nobs);
3 run;
```

If one needs to flag the order of the variables in a SAS dataset can take advantage of PG Stats' (Esteemed Advisor) code in the user forum using the Dictionary meta-data shown below:

```
1 proc sql;
2 create table expect (var1 real, var2 real, var3 real);
3 create table other (var2 real, var3 real, var1 real);
4 quit;
5
6 title "Variables out of order in dataset OTHER";
7 proc sql;
8 select *
9 from
10 (select name, varnum "Wrong Order" from dictionary.columns
11 where libname = "WORK" and memname="OTHER" )
12
13 except
14
15 (select name, varnum from dictionary.columns
16 where libname = "WORK" and memname="EXPECT" );
17 quit;
```

A PICTORIAL REPRESENTATION OF THE PROCESS



CODE

```
1 /*MACRO THAT HELPS IDENTIFY MISSING VARIABLES FROM 2 SAS DATASETS*/
2 %MACRO DELTA_V(DSN1,DSN2);
3 /* THE 2 DATASETS ARE READ IN THE NEXT 2 STEPS WITH OBS=0 AS
4 WE ARE JUST INTERESTED IN THE VARIABLE NAMES AND NOT THE DATA PER SE */
5 options obs=0;
6 data DSN1;
7 set &DSN1;
8 run;
9 data DSN2;
10 set &DSN2;
11 run;
12 options obs=max;
13 /* THE NEXT 'PROC SQL' STEP CATEGORIZES THE VARIABLES INTO
14 3 CATEGORIES - COMMON, A-NOT-B, B-NOT-A*/
15 proc sql;
16 create table COMMON_VARS as
17 select UPCASE(name) AS COMMON_VARS from dictionary.columns
18 where libname='WORK'
19 and memname='DSN1'
20 and UPCASE(name) in (select UPCASE(name) AS NAME
21 from dictionary.columns
22 where libname='WORK'
23 and memname='DSN2')
24 ORDER BY COMMON_VARS;
25
26 CREATE TABLE VARS_ONLY_IN_1ST AS
27 select UPCASE(name) AS VARS_ONLY_IN_1ST from dictionary.columns
28 where libname='WORK'
29 and memname='DSN1'
30 and UPCASE(name) not in (select UPCASE(name) AS NAME
31 from dictionary.columns
32 where libname='WORK'
33 and memname='DSN2')
34 ORDER BY VARS_ONLY_IN_1ST;
35
36 CREATE TABLE VARS_ONLY_IN_2ND AS
37 select UPCASE(name) AS VARS_ONLY_IN_2ND from dictionary.columns
38 where libname='WORK'
39 and memname='DSN2'
40 and UPCASE(name) not in (select UPCASE(name) AS NAME
41 from dictionary.columns
42 where libname='WORK'
43 and memname='DSN1')
44 ORDER BY VARS_ONLY_IN_2ND;
45 QUIT;
```

```

45 QUIT;
46 /* THE FOLLOWING MACRO GENERATES A KEY USED FOR CONSOLIDATION */
47 %MACRO MAKE_KEY(PARM1);
48     DATA &PARM1.;
49     SET &PARM1.;
50     MATCH_KEY = _N_;
51     RUN;
52 %MEND MAKE_KEY;
53 /* KEYS ARE GENERATED FOR ALL 3 FILES
54 WITH THE FOLLOWING 3 MACRO CALLS */
55 %MAKE_KEY(COMMON_VARS);
56 %MAKE_KEY(VARS_ONLY_IN_1ST);
57 %MAKE_KEY(VARS_ONLY_IN_2ND);
58 PROC SQL;
59     CREATE TABLE step1 AS
60     SELECT
61         VARS_ONLY_IN_1ST,
62         VARS_ONLY_IN_2ND
63     FROM VARS_ONLY_IN_1ST FULL JOIN VARS_ONLY_IN_2ND
64         ON VARS_ONLY_IN_1ST.MATCH_KEY = VARS_ONLY_IN_2ND.MATCH_KEY;
65 QUIT;
66 %MAKE_KEY(STEP1);
67 /* FINAL STEP WHERE ALL 3 CATEGORIES ARE COMBINED IN ALPHABETICAL ORDER */
68 PROC SQL;
69     CREATE TABLE getall AS
70     SELECT
71         VARS_ONLY_IN_1ST,
72         VARS_ONLY_IN_2ND,
73         common_vars
74     FROM step1 FULL JOIN common_vars
75         ON step1.MATCH_KEY = common_VARS.MATCH_KEY;
76 QUIT;
77 proc print data=getall LABEL;
78     LABEL COMMON_VARS = "VARS COMMON TO BOTH &DSN1 & &DSN2"
79     VARS_ONLY_IN_1ST = "VARS IN &DSN1 ONLY"
80     VARS_ONLY_IN_2ND = "VARS IN &DSN2 ONLY";
81 run;
82 %MEND DELTAV;
83
84 data class(genmax=5);set sashelp.class(drop=height);run;
85 data class(genmax=5);set sashelp.class(drop=weight);run;
86 %deltav(work.class(gennum=0),work.class(gennum=-1))
87 %deltav(sashelp.class,sashelp.demographics)

```

LOG

```
134 124      data class(genmax=5);set sashelp.class(drop=height);run;
135
136 NOTE: There were 19 observations read from the data set SASHELP.CLASS.
137 NOTE: The data set WORK.CLASS has 19 observations and 4 variables.
138 NOTE: DATA statement used (Total process time):
139     real time          0.23 seconds
140     cpu time           0.03 seconds
141
142
143 125      data class(genmax=5);set sashelp.class(drop=weight);run;
144
145 NOTE: There were 19 observations read from the data set SASHELP.CLASS.
146 NOTE: The data set WORK.CLASS has 19 observations and 4 variables.
147 NOTE: DATA statement used (Total process time):
148     real time          0.08 seconds
149     cpu time           0.01 seconds
150
151
152 126      %deltav(work.class(gennum=0),work.class(gennum=-1))
153 MLOGIC(DELTA):  Beginning execution.
154 MLOGIC(DELTA):  Parameter DSN1 has value work.class(gennum=0)
155 MLOGIC(DELTA):  Parameter DSN2 has value work.class(gennum=-1)
156 MPRINT(DELTA):  options obs=0;
157 MPRINT(DELTA):  data DSN1;
158 SYMBOLGEN:  Macro variable DSN1 resolves to work.class(gennum=0)
159 MPRINT(DELTA):  set work.class(gennum=0);
160 MPRINT(DELTA):  run;
161
162 NOTE: There were 0 observations read from the data set WORK.CLASS.
163 NOTE: The data set WORK.DSN1 has 0 observations and 4 variables.
164 NOTE: DATA statement used (Total process time):
165     real time          0.07 seconds
166     cpu time           0.00 seconds
167
168
169 MPRINT(DELTA):  data DSN2;
170 SYMBOLGEN:  Macro variable DSN2 resolves to work.class(gennum=-1)
171 MPRINT(DELTA):  set work.class(gennum=-1);
172 MPRINT(DELTA):  run;
173
174 NOTE: There were 0 observations read from the data set WORK.CLASS (gennum=1).
175 NOTE: The data set WORK.DSN2 has 0 observations and 4 variables.
176 NOTE: DATA statement used (Total process time):
177     real time          0.08 seconds
```

178: cpu time 0.01 seconds

179:

180:

181: 4

The SAS System

09:45 Tuesday, August 20, 2019

182:

183: MPRINT(DELTA): options obs=max;

184: MPRINT(DELTA): proc sql;

185: MPRINT(DELTA): create table COMMON_VARS as select UPCASE(name) AS COMMON_VARS from dictionary.columns where

186: libname='WORK' and memname='DSN1' and UPCASE(name) in (select UPCASE(name) AS NAME from dictionary.columns where

187: libname='WORK' and memname='DSN2') ORDER BY COMMON_VARS;

188: NOTE: Table WORK.COMMON_VARS created, with 3 rows and 1 columns.

189:

190: MPRINT(DELTA): CREATE TABLE VARS_ONLY_IN_1ST AS select UPCASE(name) AS VARS_ONLY_IN_1ST from dictionary.columns

191: where libname='WORK' and memname='DSN1' and UPCASE(name) not in (select UPCASE(name) AS NAME from dictionary.columns

192: where libname='WORK' and memname='DSN2') ORDER BY VARS_ONLY_IN_1ST;

193: NOTE: Table WORK.VARS_ONLY_IN_1ST created, with 1 rows and 1 columns.

194:

195: MPRINT(DELTA): CREATE TABLE VARS_ONLY_IN_2ND AS select UPCASE(name) AS VARS_ONLY_IN_2ND from dictionary.columns

196: where libname='WORK' and memname='DSN2' and UPCASE(name) not in (select UPCASE(name) AS NAME from dictionary.columns

197: where libname='WORK' and memname='DSN1') ORDER BY VARS_ONLY_IN_2ND;

198: NOTE: Table WORK.VARS_ONLY_IN_2ND created, with 1 rows and 1 columns.

199:

200: MPRINT(DELTA): QUIT;

201: NOTE: PROCEDURE SQL used (Total process time):

202: real time 0.33 seconds

203: cpu time 0.04 seconds

204:

205:

206: MLOGIC(MAKE_KEY): Beginning execution.

207: MLOGIC(MAKE_KEY): Parameter PARM1 has value COMMON_VARS

208: SYMBOLGEN: Macro variable PARM1 resolves to COMMON_VARS

209: MPRINT(MAKE_KEY): DATA COMMON_VARS;

210: SYMBOLGEN: Macro variable PARM1 resolves to COMMON_VARS

211: MPRINT(MAKE_KEY): SET COMMON_VARS;

212: MPRINT(MAKE_KEY): MATCH_KEY = _N_;

213: MPRINT(MAKE_KEY): RUN;

214:

215: NOTE: There were 3 observations read from the data set WORK.COMMON_VARS.

216: NOTE: The data set WORK.COMMON_VARS has 3 observations and 2 variables.

217: NOTE: DATA statement used (Total process time):

218: real time 0.05 seconds

219: cpu time 0.01 seconds

220:

221:

222: MLOGIC(MAKE_KEY): Ending execution.

```

223 MPRINT(DELTA): ;
224 MLOGIC(MAKE_KEY): Beginning execution.
225 MLOGIC(MAKE_KEY): Parameter PARM1 has value VARS_ONLY_IN_1ST
226 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_1ST
227 MPRINT(MAKE_KEY): DATA VARS_ONLY_IN_1ST;
228 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_1ST
229 MPRINT(MAKE_KEY): SET VARS_ONLY_IN_1ST;
230 MPRINT(MAKE_KEY): MATCH_KEY = _N_;
231 MPRINT(MAKE_KEY): RUN;
232
233 NOTE: There were 1 observations read from the data set WORK.VARS_ONLY_IN_1ST.
234 NOTE: The data set WORK.VARS_ONLY_IN_1ST has 1 observations and 2 variables.
235 NOTE: DATA statement used (Total process time):
236     real time          0.05 seconds
237     cpu time           0.00 seconds
238
239
240 MLOGIC(MAKE_KEY): Ending execution.
241 5 The SAS System 09:45 Tuesday, August 20, 2019
242
243 MPRINT(DELTA): ;
244 MLOGIC(MAKE_KEY): Beginning execution.
245 MLOGIC(MAKE_KEY): Parameter PARM1 has value VARS_ONLY_IN_2ND
246 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_2ND
247 MPRINT(MAKE_KEY): DATA VARS_ONLY_IN_2ND;
248 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_2ND
249 MPRINT(MAKE_KEY): SET VARS_ONLY_IN_2ND;
250 MPRINT(MAKE_KEY): MATCH_KEY = _N_;
251 MPRINT(MAKE_KEY): RUN;
252
253 NOTE: There were 1 observations read from the data set WORK.VARS_ONLY_IN_2ND.
254 NOTE: The data set WORK.VARS_ONLY_IN_2ND has 1 observations and 2 variables.
255 NOTE: DATA statement used (Total process time):
256     real time          0.06 seconds
257     cpu time           0.02 seconds
258
259
260 MLOGIC(MAKE_KEY): Ending execution.
261 MPRINT(DELTA): ;
262 MPRINT(DELTA): PROC SQL;
263 MPRINT(DELTA): CREATE TABLE step1 AS SELECT VARS_ONLY_IN_1ST, VARS_ONLY_IN_2ND FROM VARS_ONLY_IN_1ST FULL JOIN
264 VARS_ONLY_IN_2ND ON VARS_ONLY_IN_1ST.MATCH_KEY = VARS_ONLY_IN_2ND.MATCH_KEY;
265 NOTE: Table WORK.STEP1 created, with 1 rows and 2 columns.
266
267 MPRINT(DELTA): QUIT;

```

```

268 NOTE: PROCEDURE SQL used (Total process time):
269     real time          0.05 seconds
270     cpu time           0.00 seconds
271
272
273 MLOGIC(MAKE_KEY): Beginning execution.
274 MLOGIC(MAKE_KEY): Parameter PARM1 has value STEP1
275 SYMBOLGEN: Macro variable PARM1 resolves to STEP1
276 MPRINT(MAKE_KEY): DATA STEP1;
277 SYMBOLGEN: Macro variable PARM1 resolves to STEP1
278 MPRINT(MAKE_KEY): SET STEP1;
279 MPRINT(MAKE_KEY): MATCH_KEY = _N_;
280 MPRINT(MAKE_KEY): RUN;
281
282 NOTE: There were 1 observations read from the data set WORK.STEP1.
283 NOTE: The data set WORK.STEP1 has 1 observations and 3 variables.
284 NOTE: DATA statement used (Total process time):
285     real time          0.04 seconds
286     cpu time           0.01 seconds
287
288
289 MLOGIC(MAKE_KEY): Ending execution.
290 MPRINT(DELTAV): ;
291 MPRINT(DELTAV): PROC SQL;
292 MPRINT(DELTAV): CREATE TABLE getall AS SELECT VARS_ONLY_IN_1ST, VARS_ONLY_IN_2ND, common_vars FROM step1 FULL JOIN
293 common_vars ON step1.MATCH_KEY = common_VARS.MATCH_KEY;
294 NOTE: Table WORK.GETALL created, with 3 rows and 3 columns.
295
296 MPRINT(DELTAV): QUIT;
297 NOTE: PROCEDURE SQL used (Total process time):
298     real time          0.04 seconds
299     cpu time           0.00 seconds
300
301  6 The SAS System 09:45 Tuesday, August 20, 2019
302
303
304 MPRINT(DELTAV): proc print data=getall LABEL;
305 SYMBOLGEN: Macro variable DSN1 resolves to work.class(gennum=0)
306 SYMBOLGEN: Macro variable DSN2 resolves to work.class(gennum=-1)
307 SYMBOLGEN: Macro variable DSN1 resolves to work.class(gennum=0)
308 SYMBOLGEN: Macro variable DSN2 resolves to work.class(gennum=-1)
309 MPRINT(DELTAV): LABEL COMMON_VARS = "VARS COMMON TO BOTH work.class(gennum=0) & work.class(gennum=-1)"
310 VARS_ONLY_IN_1ST = "VARS IN work.class(gennum=0) ONLY" VARS_ONLY_IN_2ND = "VARS IN work.class(gennum=-1) ONLY";
311 MPRINT(DELTAV): run;
312

```

```

313 NOTE: There were 3 observations read from the data set WORK.GETALL.
314 NOTE: PROCEDURE PRINT used (Total process time):
315     real time          0.04 seconds
316     cpu time           0.00 seconds
317
318
319 MLOGIC(DELTA): Ending execution.
320 127     %deltav(sashelp.class,sashelp.demographics)
321 MLOGIC(DELTA): Beginning execution.
322 MLOGIC(DELTA): Parameter DSN1 has value sashelp.class
323 MLOGIC(DELTA): Parameter DSN2 has value sashelp.demographics
324 MPRINT(DELTA): options obs=0;
325 MPRINT(DELTA): data DSN1;
326 SYMBOLGEN: Macro variable DSN1 resolves to sashelp.class
327 MPRINT(DELTA): set sashelp.class;
328 MPRINT(DELTA): run;
329
330 NOTE: There were 0 observations read from the data set SASHELP.CLASS.
331 NOTE: The data set WORK.DSN1 has 0 observations and 5 variables.
332 NOTE: DATA statement used (Total process time):
333     real time          0.06 seconds
334     cpu time           0.00 seconds
335
336
337 MPRINT(DELTA): data DSN2;
338 SYMBOLGEN: Macro variable DSN2 resolves to sashelp.demographics
339 MPRINT(DELTA): set sashelp.demographics;
340 MPRINT(DELTA): run;
341
342 NOTE: There were 0 observations read from the data set SASHELP.DEMOGRAPHICS.
343 NOTE: The data set WORK.DSN2 has 0 observations and 18 variables.
344 NOTE: DATA statement used (Total process time):
345     real time          0.10 seconds
346     cpu time           0.02 seconds
347
348
349 MPRINT(DELTA): options obs=max;
350 MPRINT(DELTA): proc sql;
351 MPRINT(DELTA): create table COMMON_VARS as select UPCASE(name) AS COMMON_VARS from dictionary.columns where
352 libname='WORK' and memname='DSN1' and UPCASE(name) in (select UPCASE(name) AS NAME from dictionary.columns where
353 libname='WORK' and memname='DSN2') ORDER BY COMMON_VARS;
354 NOTE: Table WORK.COMMON_VARS created, with 1 rows and 1 columns.
355
356 MPRINT(DELTA): CREATE TABLE VARS_ONLY_IN_1ST AS select UPCASE(name) AS VARS_ONLY_IN_1ST from dictionary.columns
357 where libname='WORK' and memname='DSN1' and UPCASE(name) not in (select UPCASE(name) AS NAME from dictionary.columns

```

```

358 where libname='WORK' and memname='DSN2') ORDER BY VARS_ONLY_IN_1ST;
359 NOTE: Table WORK.VARS_ONLY_IN_1ST created, with 4 rows and 1 columns.
360
361 7 The SAS System 09:45 Tuesday, August 20, 2019
362
363 MPRINT(DELTA): CREATE TABLE VARS_ONLY_IN_2ND AS select UPCASE(name) AS VARS_ONLY_IN_2ND from dictionary.columns
364 where libname='WORK' and memname='DSN2' and UPCASE(name) not in (select UPCASE(name) AS NAME from dictionary.columns
365 where libname='WORK' and memname='DSN1') ORDER BY VARS_ONLY_IN_2ND;
366 NOTE: Table WORK.VARS_ONLY_IN_2ND created, with 17 rows and 1 columns.
367
368 MPRINT(DELTA): QUIT;
369 NOTE: PROCEDURE SQL used (Total process time):
370     real time      0.20 seconds
371     cpu time       0.03 seconds
372
373
374 MLOGIC(MAKE_KEY): Beginning execution.
375 MLOGIC(MAKE_KEY): Parameter PARM1 has value COMMON_VARS
376 SYMBOLGEN: Macro variable PARM1 resolves to COMMON_VARS
377 MPRINT(MAKE_KEY): DATA COMMON_VARS;
378 SYMBOLGEN: Macro variable PARM1 resolves to COMMON_VARS
379 MPRINT(MAKE_KEY): SET COMMON_VARS;
380 MPRINT(MAKE_KEY): MATCH_KEY = _N_;
381 MPRINT(MAKE_KEY): RUN;
382
383 NOTE: There were 1 observations read from the data set WORK.COMMON_VARS.
384 NOTE: The data set WORK.COMMON_VARS has 1 observations and 2 variables.
385 NOTE: DATA statement used (Total process time):
386     real time      0.05 seconds
387     cpu time       0.01 seconds
388
389
390 MLOGIC(MAKE_KEY): Ending execution.
391 MPRINT(DELTA): ;
392 MLOGIC(MAKE_KEY): Beginning execution.
393 MLOGIC(MAKE_KEY): Parameter PARM1 has value VARS_ONLY_IN_1ST
394 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_1ST
395 MPRINT(MAKE_KEY): DATA VARS_ONLY_IN_1ST;
396 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_1ST
397 MPRINT(MAKE_KEY): SET VARS_ONLY_IN_1ST;
398 MPRINT(MAKE_KEY): MATCH_KEY = _N_;
399 MPRINT(MAKE_KEY): RUN;
400
401 NOTE: There were 4 observations read from the data set WORK.VARS_ONLY_IN_1ST.
402 NOTE: The data set WORK.VARS ONLY IN 1ST has 4 observations and 2 variables.

```

```

403 NOTE: DATA statement used (Total process time):
404     real time          0.05 seconds
405     cpu time           0.01 seconds
406
407
408 MLOGIC(MAKE_KEY): Ending execution.
409 MPRINT(DELTAV): ;
410 MLOGIC(MAKE_KEY): Beginning execution.
411 MLOGIC(MAKE_KEY): Parameter PARM1 has value VARS_ONLY_IN_2ND
412 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_2ND
413 MPRINT(MAKE_KEY): DATA VARS_ONLY_IN_2ND;
414 SYMBOLGEN: Macro variable PARM1 resolves to VARS_ONLY_IN_2ND
415 MPRINT(MAKE_KEY): SET VARS_ONLY_IN_2ND;
416 MPRINT(MAKE_KEY): MATCH_KEY = _N_;
417 MPRINT(MAKE_KEY): RUN;
418
419 NOTE: There were 17 observations read from the data set WORK.VARS_ONLY_IN_2ND.
420 NOTE: The data set WORK.VARS_ONLY_IN_2ND has 17 observations and 2 variables.
421 8 The SAS System 09:45 Tuesday, August 20, 2019
422
423 NOTE: DATA statement used (Total process time):
424     real time          0.05 seconds
425     cpu time           0.00 seconds
426
427
428 MLOGIC(MAKE_KEY): Ending execution.
429 MPRINT(DELTAV): ;
430 MPRINT(DELTAV): PROC SQL;
431 MPRINT(DELTAV): CREATE TABLE step1 AS SELECT VARS_ONLY_IN_1ST, VARS_ONLY_IN_2ND FROM VARS_ONLY_IN_1ST FULL JOIN
432 VARS_ONLY_IN_2ND ON VARS_ONLY_IN_1ST.MATCH_KEY = VARS_ONLY_IN_2ND.MATCH_KEY;
433 NOTE: Table WORK.STEP1 created, with 17 rows and 2 columns.
434
435 MPRINT(DELTAV): QUIT;
436 NOTE: PROCEDURE SQL used (Total process time):
437     real time          0.04 seconds
438     cpu time           0.01 seconds
439
440
441 MLOGIC(MAKE_KEY): Beginning execution.
442 MLOGIC(MAKE_KEY): Parameter PARM1 has value STEP1
443 SYMBOLGEN: Macro variable PARM1 resolves to STEP1
444 MPRINT(MAKE_KEY): DATA STEP1;
445 SYMBOLGEN: Macro variable PARM1 resolves to STEP1
446 MPRINT(MAKE_KEY): SET STEP1;
447 MPRINT(MAKE_KEY): MATCH_KEY = _N_;

```

```

448 MPRINT(MAKE_KEY):  RUN;
449
450 NOTE: There were 17 observations read from the data set WORK.STEP1.
451 NOTE: The data set WORK.STEP1 has 17 observations and 3 variables.
452 NOTE: DATA statement used (Total process time):
453     real time          0.06 seconds
454     cpu time           0.00 seconds
455
456
457 MLOGIC(MAKE_KEY):  Ending execution.
458 MPRINT(DELTA):  ;
459 MPRINT(DELTA):  PROC SQL;
460 MPRINT(DELTA):  CREATE TABLE getall AS SELECT VARS_ONLY_IN_1ST, VARS_ONLY_IN_2ND, common_vars FROM step1 FULL JOIN
461 common_vars ON step1.MATCH_KEY = common_VARS.MATCH_KEY;
462 NOTE: Table WORK.GETALL created, with 17 rows and 3 columns.
463
464 MPRINT(DELTA):  QUIT;
465 NOTE: PROCEDURE SQL used (Total process time):
466     real time          0.04 seconds
467     cpu time           0.01 seconds
468
469
470 MPRINT(DELTA):  proc print data=getall LABEL;
471 SYMBOLGEN:  Macro variable DSN1 resolves to sashelp.class
472 SYMBOLGEN:  Macro variable DSN2 resolves to sashelp.demographics
473 SYMBOLGEN:  Macro variable DSN1 resolves to sashelp.class
474 SYMBOLGEN:  Macro variable DSN2 resolves to sashelp.demographics
475 MPRINT(DELTA):  LABEL COMMON_VARS = "VARS COMMON TO BOTH sashelp.class & sashelp.demographics" VARS_ONLY_IN_1ST =
476 "VARS IN sashelp.class ONLY" VARS_ONLY_IN_2ND = "VARS IN sashelp.demographics ONLY";
477 MPRINT(DELTA):  run;
478
479 NOTE: There were 17 observations read from the data set WORK.GETALL.
480 NOTE: PROCEDURE PRINT used (Total process time):
481 9
482
483     real time          0.03 seconds
484     cpu time           0.00 seconds
485
486
487 MLOGIC(DELTA):  Ending execution.

```

RESULTS

Send to Create Publish Properties

Obs	VARS IN work.class(gennum=0) ONLY	VARS IN work.class(gennum=-1) ONLY	VARS COMMON TO BOTH work.class(gennum=0) & work.class(gennum=-1)
1	HEIGHT	WEIGHT	AGE
2			NAME
3			SEX

Page Break

Obs	VARS IN sashelp.class ONLY	VARS IN sashelp.demographics ONLY	VARS COMMON TO BOTH sashelp.class & sashelp.demographics
1	AGE	ADOLESCENTFPCT	NAME
2	HEIGHT	ADOLESCENTFPYEAR	
3	SEX	ADULTLITERACY PCT	
4	WEIGHT	CONT	
5		FEMALESCHOOLPCT	
6		GNI	
7		ID	
8		ISO	
9		ISONAME	
10		MALESCHOOLPCT	
11		POP	
12		POPAGR	
13		POPPOVERTYPCT	
14		POPPOVERTY YEAR	
15		POPURBAN	
16		REGION	
17		TOTALFR	

Page Break

CONCLUSION

The macro can be further enhanced for comparing 3 or 4 or even multiple datasets at once if necessary just by cloning the code presented based on a user's requirement. The more frequently this process is used, the more benefit the team reaps as each time a comparison is created, the differences are presented right in front categorized. This is just another example that shows how DICTIONARY tables can be harnessed to its maximum benefit.

REFERENCES

Bennett, A. D. (2001). SUGI26: Side By Side: Comparing Two Data Sets Using SAS® Macros. In *SUGI26:*

Side By Side: Comparing Two Data Sets Using SAS® Macros.

Lee, B. C. (2012). A Simple yet Effective Way to Perform a Variable Cross Walk Between Multiple

Datasets. In *A Simple yet Effective Way to Perform a Variable Cross Walk Between Multiple*

Datasets. SAS Global Forum.

PG Stats, E. A. (2015, April 12). Retrieved from <https://communities.sas.com/t5/SAS->

[Procedures/Comparing-variable-order-between-2-datasets/td-p/197876](https://communities.sas.com/t5/SAS-Procedures/Comparing-variable-order-between-2-datasets/td-p/197876)

ACKNOWLEDGMENTS

Wellcare management and leadership, for the support and encouragement.

RECOMMENDED READING

- Base SAS® Procedures Guide
- Advanced SAS

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dr. Kannan Deivasigamani

TEXTdrk@gmail.com

<https://www.linkedin.com/in/kannandeivasigamani>