



Use of ARRAYs in SAS

Ruurd Bennink, OCS Consulting B.V., 's-Hertogenbosch,
the Netherlands

Purposes



- **From N variables to N records (example 1)**
- **Same operation on many variables**
- **Creating N derived variables in one data step (example 2)**
- **Creating N derived variables from N original variables (example 3)**
- **Perform analyses on datasets with many variables (example 4)**

**All examples are gained from my daily life experience as
Statistical programmer in the Life Science industry.**

Example 1: From N variables to N records (visit dependent populations)



```
DATA popul;  
  KEEP subjid dvisitnum itt pp product;  
  MERGE inp_fa.itt_pp  
        ad_fa.adsl (Keep=subjid product);  
  BY subjid;  
  
  ARRAY pops (4) pprotbl pprotv2 pprotv3  
        pprotv4;  
  
  DO dvisitnum = 1 TO 4;  
    pp = pops(dvisitnum);  
    OUTPUT;  
  END;  
  
RUN;
```



Example 1 (Continued)

Input dataset INP_FA.ITT_PP has structure:

SUBJID	PRODUCT	ITT	PPROTV1	PPROTV2	PPROTV3	PPROTV4
1	A	YES	YES	YES	NO	YES

And output dataset POPUL:

SUBJID	PRODUCT	DVISITNUM	ITT	PP
1	A	1	YES	YES
1	A	2	YES	YES
1	A	3	YES	NO
1	A	4	YES	YES

Example 2: Creating N derived variables in one data step (Cumulative incidence)



```
PROC SORT Data=ae_2 Out=ae_3;  
  BY subjid Descending AE_YN;  
RUN;
```

```
PROC SORT Data=ae_3 Out=ae_4 Nodupkey;  
  BY subjid;  
RUN;
```

Input dataset AE_2 has structure:

SUBJID	GRP	DAYONSP	AE_YN
1	ACTIVE	1	0
1	ACTIVE	2	0
1	ACTIVE	3	0
1	ACTIVE	4	1
1	ACTIVE	5	0
.	ACTIVE	.	.
1	ACTIVE	129	0



Example 2 (Continued 1)

```
DATA cumul;  
  KEEP subjid grp stday;;  
  SET AE_4;  
  
  FORMAT stday: yesnodv.;  
  
  ARRAY startday (129) stday1 - stday129;  
  
  DO i = 1 TO 129;  
    IF (i >= dayonsp ) AND (AE_YN = 1) THEN  startday(i) = 1;  
    ELSE startday(i) = 0;  
  END;  
/* Dayonsp = number of days the subject used the study  
product */  
RUN;
```



Example 2 (Continued 2)

Output dataset CUMUL has structure:

SUBJID	GRP	STDAY1	STDAY2	STDAY3	STDAY4	STDAY5	STDAY6	...	STDAY129
1	ACTIVE	0	0	0	1	1	1	1	1

After applying:

```
PROC FREQ Data=cumul;
```

```
TABLES stday: * grp / Chisq;
```

```
ODS OUTPUT crosstabfreqs=freqtabdata;
```

```
RUN;
```

it will be possible to produce plots with the cumulative incidence of the event by day on study product.

Example 3: Creating N derived variables from N original variables (change from baseline)



```
DATA chol2;  
  MERGE chol0  
        chol1  
        derived.infchar;  
  BY identity;  
  
  ARRAY blvalues (12) ffa_bl tchol_bl tg_bl hdlc_bl ldlc_bl  
    hdl_ldl_bl /* etc */;  
  ARRAY values (12) ffa tchol tg hdlc ldlc hdl_ldl /* etc */;  
  ARRAY cblvalues (12) ffa_cbl tchol_cbl tg_cbl hdlc_cbl /* etc */;  
  ARRAY percdbl (12) ffa_pdbl tchol_pdbl tg_pdbl hdlc_pdbl  
    ldlc_pdbl hdl_ldl_pdbl /* etc */;  
  
  DO i = 1 TO 12;  
    cblvalues(i) = values(i) - blvalues(i);  
    percdbl(i) = Round((cblvalues(i)/blvalues(i))*100,0.1);  
  END;  
  
RUN;
```

Example 4: Perform analyses on datasets with many variables (PBMC lab data)



```
DATA extra;  
  LENGTH name $32;  
  KEEP name j;  
  SET pbmc;  
  
  /* all numeric variables in immune_num */  
  ARRAY def{*} _numeric_;  
  
  DO j = 1 to Dim(def);  
    /* get name of numeric variable */  
    CALL VNAME(def{j}, name);  
    /* write name to an observation */  
    OUTPUT;  
  END;  
  STOP;  
RUN;
```

Example 4 (Continued 1)



```
DATA _NULL_;  
  SET extra END=LAST;  
  
  IF (j > 0) THEN CALL SYMPUT('var' || Left(j), name);  
  IF LAST THEN CALL SYMPUT('N', Left(j));  
  
RUN;  
  
%MACRO multivar;  
  
%DO j = 1 %TO &N;  
  
  PROC MIXED Data=pbmc (Where=(asrxy IN (1,2)));  
    CLASS asrxy hanif12mon birthmod;  
    MODEL &&var&j = asrxy hanif12mon birthmod;  
    ODS OUTPUT tests3=file&j;  
RUN;
```

Example 4 (Continued 2)



PBMC variable	Description	Treatment	AEDS
cd14negdrposdrmfi_unc	mean fluorescence intensity of hladr expression on cd14 negative mononuclear cells in uncultured pbmc	0.46440	0.23863
cd14negdrposperc_unc	% mononuclear cells which were cd14negative, hladr positive in uncultured pbmc	0.01805	0.81463
cd14posdrmfi_unc	mean fluorescence intensity of hla-dr expression (percp) of total mononuclear cells population	0.64649	0.42312
cd14posfitemfi_unc	mean fluorescence intensity of cd14 expression (fite) of mononuclear cells population	0.71498	0.89996
cd14posperc_unc	% CD14 positive, hla-dr positive cells of total mononuclear cells population in uncultured PBMC	0.29321	0.16303
meddivthnum_60dcult	Number of CD3+CD4+ cells that are dividing in Medium 6day cultured PBMC	0.67768	0.86677
meddivthperc_60dcult	% of CD3+CD4+ cells that are dividing in Medium 6day cultured PBMC	0.78878	0.87920
meddivtregcd25mfi_60dcult	Mean fluorescence intensity of CD25 (pe-cy7) on dividing Treg cells in Medium 6day cultured PBMC	0.85163	0.23896

CONCLUSION



There are many programming challenges that may ask for the use of SAS ARRAYS as an alternative to commonly used procedures such as PROC CONTENTS, TRANSPOSE, or SQL. In this presentation only some examples are presented that I came across with during my daily life practice in the Life Science industry. The number of possibilities and applications for ARRAYS are practically endless. Hopefully this presentation has convinced you of the elegance and transparency of SAS ARRAYS.

Questions?

