

Efficiently Converting Datasets to Vertical Structure using Metadata

Thevaki Mahadevan, Syne Qua Non, Norfolk, UK

Introduction

- **Raw Data** extracted from CRF in **horizontal data structure** gets **converted to vertical structure** by programmers for analysis purposes.
- By utilizing **variable prefixes** and **metadata**, via the **proc contents** procedure, horizontal datasets can then be converted into vertical structure via macro automation.
- Generally proc transpose or arrays are used for this purpose but the advantages of the proc contents approach is that it avoids the need for manually entering each variable and their labels, hence **saving time** and **improving quality** and **supporting traceability**, in particular, when there are many parameters to transpose.

Trends in aCRF Data Mapping

STUDY

Protocol

STUDY_SITE

Study Site

PATIENT

Subject

PAGE_LABEL

Page No.

SUBEVENT_NUMBER

Repeat No.

DOMAIN_AIM1

Visit name

CLIN_PLAN_EVE_NAME

ABNORMAL INVOLUNTARY MOVEMENT SCALE:

(Page 1 of 2)

☐ NOT DONE

AIMQ3

Date of Exam

(dd/MM/yyyy)

AIMQ1

8

Time of Exam

(24 hour clock)

AIMTM6

Instructions:

Complete examination procedure before making ratings. MOVEMENT RATINGS: Rate highest severity observed. Rate movements that occur upon activation one less than those observed spontaneously.

Code:

0 = None
1 = Minimal, may be extreme normal
2 = Mild
3 = Moderate
4 = Severe

FACIAL AND ORAL MOVEMENTS:

1. Muscles of Facial Expression
e.g., movements of forehead, eyebrows, periorbital area, cheeks; include frowning, blinking, smiling, grimacing

2. Lips & Perioral Area
e.g., picking, pouting, smacking

3. Jaw
e.g., biting, clenching, chewing, mouth opening, lateral movement

4. Tongue
Rate only increase in movement both in and out of mouth, NOT inability to sustain movement

EXTREMITY MOVEMENTS:

5. Upper arms, wrists, hands, fingers
Include choreic movements, (i.e., rapid, objectively purposeless, irregular, spontaneous), athetoid movements (i.e., slow, irregular, complex, serpentine). Do NOT include tremor (i.e., repetitive, regular, rhythmic)

6. Lower (legs, knees, ankles, toes)
e.g., lateral knee movement, foot tapping, heel dropping, foot squaring, inversion and eversion of foot

TRUNK MOVEMENTS:

7. Neck, Shoulders, Hips
e.g., rocking, twisting, squaring, pelvic gyrations

Variable names within a domain tend to have the same prefix letters.

e.g. **AIM** domain shown in the aCRF page has the **same prefix letters, AIMxxx**, for all the domain specific variables.

Figure 1: Example of an aCRF domain, showing the trends in variable names when responding to each questions

Horizontal RAW Dataset

SUBJID	CPEVENT	AIMND	AIMFACE	AIMFACEF	AIMLIP	AIMLIPF	AIMJAW	AIMJAWF	AIMTONG	AIMTONGF	AIMUPPR	AIMUPPRF	AIMLOW	AIMLOWF
10011001	DAY 1		0	0	0	0	0	0	0	0	0	0	0	0
10011001	DAY 8				0	0	0	0	0	0	0	0	0	0
10011001	DAY15		0	0	0	0	0	0	0	0	0	0	0	0
10011001	DAY22		0	0	0	0	0	0	0	0	0	0	0	0
10011001	END_OF_TREATMENT		0	0	0	0	0	0	0	0	0	0	0	0
10011001	FOLLOW_UP	NOT DONE	.	.	.	.	.	.	.	.	.	.	.	.
10011002	DAY 1		0	0	0	0	0	0	0	0	0	0	0	0
10011002	DAY 8		0	0	0	0	0	0	0	0	0	0	0	0
10011002	DAY15		0	0	0	0	0	0	0	0	0	0	0	0
10011002	DAY22		0	0	0	0	0	0	0	0	0	0	0	0
10011002	END_OF_TREATMENT		0	0	0	0	0	0	0	0	0	0	0	0
10011002	FOLLOW_UP	NOT DONE	.	.	.	.	.	.	.	.	.	.	.	.
10011004	DAY 1		0	0	0	0	0	0	0	0	0	0	0	0
10011004	DAY 8		0	0	0	0	0	0	0	0	0	0	0	0
10011004	DAY15		0	0	0	0	0	0	0	0	0	0	0	0

Figure 2: Example of a horizontally structured raw dataset

- As mapped in the aCRF, the same variable names are seen in the horizontally structured raw dataset and support an understanding of the relationship of aCRF to its raw dataset, aiding traceability.
- Generally as part of the mapping process, each variable collected is formatted to either two or three forms; the numeric, character and long/full (decoded/formatted) version.

Vertical Analysis Dataset

	CPEVENT	EFD1	EFND	EFQUES	EFQUESN	EFRAW	EFERSLT	SUBJID	EFTM
DAY 1	21AUG2014			AIMFACE	1	0	0	10011001	17:01:00
DAY 1	21AUG2014			AIMLIP	2	0	0	10011001	17:01:00
DAY 1	21AUG2014			AIMJAW	3	0	0	10011001	17:01:00
DAY 1	21AUG2014			AIMTONG	4	0	0	10011001	17:01:00
DAY 1	21AUG2014			AIMUPPR	5	0	0	10011001	17:01:00
DAY 8	28AUG2014			AIMFACE	1	0	0	10011001	14:10:00
DAY 8	28AUG2014			AIMLIP	2	0	0	10011001	14:10:00
DAY 8	28AUG2014			AIMJAW	3	0	0	10011001	14:10:00
DAY 8	28AUG2014			AIMTONG	4	0	0	10011001	14:10:00
DAY 8	28AUG2014			AIMUPPR	5	0	0	10011001	14:10:00
DAY15	04SEP2014			AIMFACE	1	0	0	10011001	15:02:00
DAY15	04SEP2014			AIMLIP	2	0	0	10011001	15:02:00
DAY15	04SEP2014			AIMJAW	3	0	0	10011001	15:02:00
DAY15	04SEP2014			AIMTONG	4	0	0	10011001	15:02:00
DAY15	04SEP2014			AIMUPPR	5	0	0	10011001	15:02:00
DAY22	10SEP2014			AIMFACE	1	0	0	10011001	16:20:00
DAY22	10SEP2014			AIMLIP	2	0	0	10011001	16:20:00
DAY22	10SEP2014			AIMJAW	3	0	0	10011001	16:20:00
DAY22	10SEP2014			AIMTONG	4	0	0	10011001	16:20:00
DAY22	10SEP2014			AIMUPPR	5	0	0	10011001	16:20:00
END_OF_TREATMENT	19SEP2014			AIMFACE	1	0	0	10011001	8:02:00
END_OF_TREATMENT	19SEP2014			AIMLIP	2	0	0	10011001	8:02:00
END_OF_TREATMENT	19SEP2014			AIMJAW	3	0	0	10011001	8:02:00
END_OF_TREATMENT	19SEP2014			AIMTONG	4	0	0	10011001	8:02:00
END_OF_TREATMENT	19SEP2014			AIMUPPR	5	0	0	10011001	8:02:00
FOLLOW_UP	26SEP2014	NOT DONE		AIMFACE	1	.	.	10011001	.
FOLLOW_UP	26SEP2014	NOT DONE		AIMLIP	2	.	.	10011001	.
FOLLOW_UP	26SEP2014	NOT DONE		AIMJAW	3	.	.	10011001	.
FOLLOW_UP	26SEP2014	NOT DONE		AIMTONG	4	.	.	10011001	.
FOLLOW_UP	26SEP2014	NOT DONE		AIMUPPR	5	.	.	10011001	.

Figure 3: Example of a vertically structured analysis dataset

Typical Transposing Approaches

1. proc transpose
2. Arrays

ROWS					COLUMNS				
SUBID	COVENT	AIMFACE	AIMLIP	VALUE	SUBID	COVENT	AIMFACE	VALUE	
1	Day1	0	1		1	Day1	AIMFACE	0	
2	Day2	1	0		2	Day2	AIMFACE	1	
					2	Day2	AIMLIP	0	

Figure 4: Sample transposing of horizontally structured raw dataset to vertically structured dataset

- The disadvantages of these approaches include:
- 1) Requiring manual entry of variables and their attributes
 - 2) Will consume more time
 - 3) Has more chances for human error
 - 4) Transparency of the data is manually influenced

Metadata transposing approach minimises these disadvantages.

Proc transpose- Approach

```
proc transpose data=hv_data out=hvn&i_&var;
  by &_bygroup;
  var &&g_&var_sv&i;
run;

/* When &&g_&var_sv&i resolves to AIMFACE*/
proc transpose data=hv_data out=hvn1numvar;
by INV INVSITE PID PROJCODE PROTNO PT QUALIFYV REPEATSN SID STUDY SUBEVE TRIALNO VISIT SUBJID COLLDATE
CPEVENT EFD1 EFDTF EFTM EFTMF EFND EFNDN EFDTP;
var AIMFACE;
run;

Data numvar;
set hvn1numvar hvn2numvar....;
Run;

Data final;
merge numvar charvar shortvar;
by INV INVSITE PID PROJCODE PROTNO PT QUALIFYV REPEATSN SID STUDY SUBEVE TRIALNO VISIT SUBJID COLLDATE
CPEVENT EFD1 EFDTF EFTM EFTMF EFND EFNDN EFDTP _flag ;
rename short_n=efques short_lab=efquesf;
Run;
```

```
/* Numeric Variables List */ _numvar =
%str(AIMFACE|AIMLIP|AIMJAW|AIMTONG|AIMUPPR|AIMLOW|AIMNECK|AIMSEV|AIMINC|AIMAWAR|AIMTEEN|AIMDENTN),

/* Character Variables List */ _charvar = %str(AIMFACEF|AIMLIPF|AIMJAWF|AIMTONGF|AIMUPPRF|AIMLOWF|AIMNECKF|
AIMSEVF|AIMINCF|AIMAWARF|AIMTEEF),

/* Shortname Variables List */ _shortvar =
%str(AIMFACE|AIMLIP|AIMJAW|AIMTONG|AIMUPPR|AIMLOW|AIMNECK|AIMSEV|AIMINC|AIMAWAR|AIMTEEN|AIMDENTN),
```



An Array- Approach

```
* data e5d_ques;
  attrib EFQUES length=$50 label="Efficacy parameter short name"
  EFQUESF length=$200 label="Efficacy parameter long name"
  EFQUESN length=8 label="Efficacy parameter number"
  EFRSLT length=8 label="Efficacy result for analysis - num"
  EFRSLTF length=$100 label="Efficacy result for analysis - char"
  EFRRAW label="Efficacy raw result - num" length=8;
  EFRRAWF label="Efficacy raw result - char" length = $80;

set e5d_2;
array ques(6) $ 8. ('E5D_UTIL', 'E5D5MOB', 'E5D5SEL', 'E5D5ACT', 'E5D5DIS', 'E5D5ANX');
array ques_1(6) (1 2 3 4 5 6);
array quesf(6) $ 200. ('EQ-5D Utility Index Score', 'Mobility', 'Self-Care', 'Usual Activities', 'Pain/Discomfort', 'Anxiety/Depression');
array quesn_1(6) utscore E5D5MOBN E5D5SELN E5D5ACTN E5D5DISN E5D5ANXN;
array rawf_1(6) $ utf E5D5MOB E5D5SEL E5D5ACT E5D5DIS E5D5ANX;

do i = 1 to dim(ques);
  efques = ques(i);
  efquesn = ques_1(i);
  efquesf = quesf(i);
  EFRSLT = quesn_1(i);
  EFRSLTF = rawf_1(i);
  efraw = quesn_1(i);
  efrawf = rawf_1(i);
output;
end;
run;
```

Metadata transposing approach

First the content is output into a dataset, to be able to access each variable names and labels, and to analyse the meta data closer.

```
proc contents data=aim out=acont noprint;
run;

/* Understand the RAW DATASET contents */
```

HEXNAME	NAME	TYPE	LENGTH	VARNUM	LABEL	FORMAT	FORMATL	INFORMAT	NPOS	NOB
AIM	AIMHHR	1	8	52	Awareness of Abnormal Movements		2		168	386
AIM	AIMHHRF	2	200	53	Awareness of Abnormal Movements-FUL	\$	200 \$		3162	386
AIM	AIMBENT	2	3	59	Wears Dentures	\$	3 \$		3765	386
AIM	AIMBENTF	2	200	61	Wears Dentures-FUL	\$	200 \$		3968	386
AIM	AIMBENTL	2	200	60	Wears Dentures-BLV	\$	200 \$		3768	386
AIM	AIMBENTN	1	8	58	Wears Dentures-BVN		10		184	386
AIM	AIMDT	1	8	2	Actual Date of Visit/Collection/Exam	DATE	9		8	386
AIM	AIMDTF	2	200	33	Actual Date of Visit/Collection/Exam-FUL	\$	200 \$		1162	386
AIM	AIMFACE	1	8	34	Muscles of Face		2		96	386
AIM	AIMFACEF	2	200	35	Muscles of Face-FUL	\$	200 \$		1362	386
AIM	AIMINC	1	8	50	Incapacity Due Abnormal Movements		2		160	386
AIM	AIMINCF	2	200	51	Incapacity Due Abnormal Movements-FUL	\$	200 \$		2962	386
AIM	AIMJAW	1	8	38	Jaw		2		112	386

Figure 5: Example of a proc content output dataset

```
/****** Identify unique variables needed as macro parameters******/
proc sort data=acont out=aconts;
  by name type;
run;

data pacont;
  set aconts;
  by name type;
  /*Remove variables not needed for transposing, e.g. date/time/not done */
  where name ? 'AIM' and name not in ('AIMND', 'AIMNDN', 'AIMNDF', 'AIMTMF', 'AIMDT', 'AIMDTF', 'AIMTEEL', 'AIMDENTL');
  /***** Keep only unique variables and remove duplicate vars ****/
  if STRIP(lag(name))||'F' eq name then delete;
  /******Handle original character separately ******/
  if STRIP(lag(name))||'N' eq name then delete;
run;

data _null_;
  set pacont end=last;
  if last then call symput('tot',_n_);
  call symput('NAME')||left(_N_), trim(left(name));
  call symput('NAML')||left(_N_), trim(left(LABEL));
  call symput('TYPE')||left(_N_), trim(left(TYPE));
run;

%put &tot &NAME1 &NAML1 &TYPE1;
```

```
/******Convert to vertical structure ******/
%macro vertical;
  data AIMq2 (DROP=AIM.);
  set AIMq1;
  length efquesn 8 efques $8 efquesf $80 efraw 8 efrawf $80 efrslf 8 efrsltf $80;
  label EFQUESN = "Efficacy parameter number"
  EFQUES = "Efficacy parameter short name"
  EFQUESF = "Efficacy parameter long name"
  EFRRAW = "Efficacy raw result - num"
  EFRRAWF = "Efficacy raw result - char"
  EFRSLT = "Efficacy result for analysis - num"
  EFRSLTF = "Efficacy result for analyses - char" ;
  %do i=1 %to &tot;
    EFQUESN = &i; /* parameter number */
    EFQUES = "&&name&i"; /* parameter short name */
    /* raw result - num */
    %if &&type&i eq 2 %then %do; EFRRAW = &&name&i.N; %end; %else
    %if &&type&i eq 1 %then %do; EFRRAW = &&name&i; %end; /*
    EFRRAWF = &&name&i.F; /* raw result - char */
    EFQUESF = "&&name&i"; /*parameter long name */
    EFRSLT = efraw; EFRSLTF = efrawf;
  output;
  %end;
run;
%mend;
%vertical;
```

Conclusion

- Several ways to transpose a Horizontally structured dataset to Vertical but they all require a **good understanding of the dataset metadata**.
- When there are many variables to transpose, a **data dependent automated approach**, has many advantages over other manually inputted approaches.