PhUSE 2017

Paper DH12

Metadata-driven Tool for Creation of a Dataset for Exploratory Analyses

Alexey Kuznetsov, Grünenthal, Aachen, Germany

ABSTRACT

The paper describes the concept of a metadata-driven tool used to pre-populate SAS® code for creation of a dataset for exploratory data mining and/or visualization of clinical trial data. The resulting dataset is a "wide" dataset containing 1 line per subject with a lot of variables, containing data from various SDTM or ADaM domains at each visit/timepoint. The advantage is that all the relevant trial data is pooled into one dataset that can be used for modeling and/or plotting the data with various statistical packages or data visualization tools without additional pre-processing of the data.

INTRODUCTION

This paper describes an idea of a data structure convenient for exploratory analyses, data mining and data visualization as well as an approach of how to convert SDTM/ADaM data into the structure in a flexible and convenient way. The examples in this paper are based on converting ADaM to the proposed structure, since it is more analysis-ready. However SDTM can be very similarly processed considering xxTESTCD as PARAMCD, VISITNUM as AVISITN, xxSTRESN as AVAL etc. The conversion is implemented via a package of SAS programs and macros to automatically generate SAS code which converts trial datasets into the proposed structure. The complete code of the corresponding programs will not be provided in the paper, but the logic of the programs will be described. Programming languages other than SAS can be also considered for creating a similar tool.

DEFINITIONS

Data visualization

Data visualization is the visual and interactive exploration and graphic representation of data of any size, type (structured and unstructured) or origin. [1]

Data mining?

Data mining is the computing process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. [2]

Metadata-driven program?

Metadata is data [information] that provides information about other data. [3] And metadata-driven program is the program that decides on how to process the data based on the provided metadata.

DATA MINING DATASET STRUCTURE OVERVIEW

The main data mining dataset will be referred to as AXDM (following CDISC naming conventions for non-ADaM analysis datasets). AXDM is a dataset containing 1 line per subject with many variables from various domains (e.g. demographic, disposition, medical history characteristics as well as efficacy, laboratory, vital signs, procedures data for all or selected parameters at each period/visit/timepoint. The advantage is that you have all the trial data combined in one dataset and you can easily do plotting and modeling of the data.

ADaM datasets

adae.xpt	6666560
adcm.xpt	40189280
addv.xpt	746720
adeg.xpt	2606255120
adex.xpt	7354000
adfa.xpt	45138480
adlb.xpt	1398837840
admh.xpt	39826800
admqs.xpt	15753600
adpc.xpt	37872000
adpi.xpt	890981920
adpis.xpt	11455298000
adqs.xpt	2376602160
adsl.xpt	2435760
adsv.xpt	2530720
adtte.xpt	9415920
advs.xpt	97746160
adzl.xpt	1203416480

AXDM dataset contents

NAME	LABEL
ADEG_1_HRMEAN_2_30	ADEG_HOLTER ECG EXTRACT_Mean Heart Rate (beats/min)_VISIT 2_30 minutes after the start of infusion
ADEG_1_HRMEAN_2_45	ADEG_HOLTER ECG EXTRACT_Mean Heart Rate (beats/min)_VISIT 2_45 minutes after the end of infusion
ADEG_1_HRMEAN_2_60	ADEG_HOLTER ECG EXTRACT_Mean Heart Rate (beats/min)_VISIT 2_60 minutes after the start of infusion
ADEG_1_HRMEAN_2_90	ADEG_HOLTER ECG EXTRACT_Mean Heart Rate (beats/min)_VISIT 2_90 minutes after the end of infusion
ADLB_CREAT_1	ADLB_Creatinine (umol/L)_VISIT 1
ADLB_CREAT_2	ADLB_Creatinine (umol/L)_VISIT 2
ADLB_CREAT_3	ADLB_Creatinine (umol/L)_VISIT 3
ADPI_WKAVNOW_1	ADPI_Weekly average current pain intensity_Week 1
ADPI_WKAVNOW_2	ADPI_Weekly average current pain intensity_Week 2
ADPI_WKAVNOW_3	ADPI_Weekly average current pain intensity_Week 3
AGE	
SITEGR1	
STRATA	
TRT01P	
USUBJID	

The second proposed type of dataset for data mining is AXDMV – the same concept, but with 1 record by subject/visit(timepoint), whereas all the parameters will still be in columns. The subject-level data (e.g. coming from ADSL is merged to all records.

TYPES OF VARIABLES IN AXDM

The proposed types of variables in AXDM are the following:

- Subject-level: variables coming from the “1-line-per-subject” datasets e.g. ADSL
- Transposed from BDS: variables created by transposing BDS datasets variable(s) AVAL/CHG/PCHG with id variables [PARCATx] <PARAMCD> [AVISIT(N)] [ATPT(N)] by subject
- Occurrence flags: flag variables indicating occurrence of an event for a subject in a period e.g. occurrence of a particular treatment-emergent adverse event (AEDECOD) for a subject or presence of a particular medical history term in a specific system organ class (MHBODSYS) for a subject
- Occurrence counts: numeric variables containing number of events that occurred for a subject e.g. number of occurrences of a particular treatment-emergent adverse event (AEDECOD)

The list can be extended if required for specific exploratory analyses.

VARIABLE NAMING CONVENTIONS AND RATIONALE

- Subject-level:

  <Dataset name>_<Orig. variable name>

  Example: ADSL_RACE

  Rationale:

  1. Clear traceability to the original dataset
  2. Ability to easily refer to all the variables coming from a particular domain as an array (e.g. ADSL_: in SAS)

- Transposed from BDS

  <Dataset name>[_PARCATx][_PARCATy]<_PARAMCD>[_AVISIT(N)][_ATPT(N)][_ANLzzFL]

  Example: ADEF_RESP_RED30P_5

Dataset	PARCAT1	PARAMCD	AVISITN

  Rationale:

  1. Same as for Subject-level variables
  2. Easy code for creation of the dataset

  e.g.:

```
proc transpose data = ADEF delimiter=_;
       by USUBJID;
       var AVAL;
       id PARAMCD AVISITN;
run;
```

- Occurrence flags

  <Dataset name>_FL[_Standardized term variable name]<_ Standardized term code>

  Examples:
  ADAE_FL_AEDECOD_HEADACHE
  ADMH_FL_MHBODSYS_1
  Standardized term code here is converted to a number since otherwise the variable name would become too long. The complete description of the term may be provided in the variable label.

- Occurrence counts

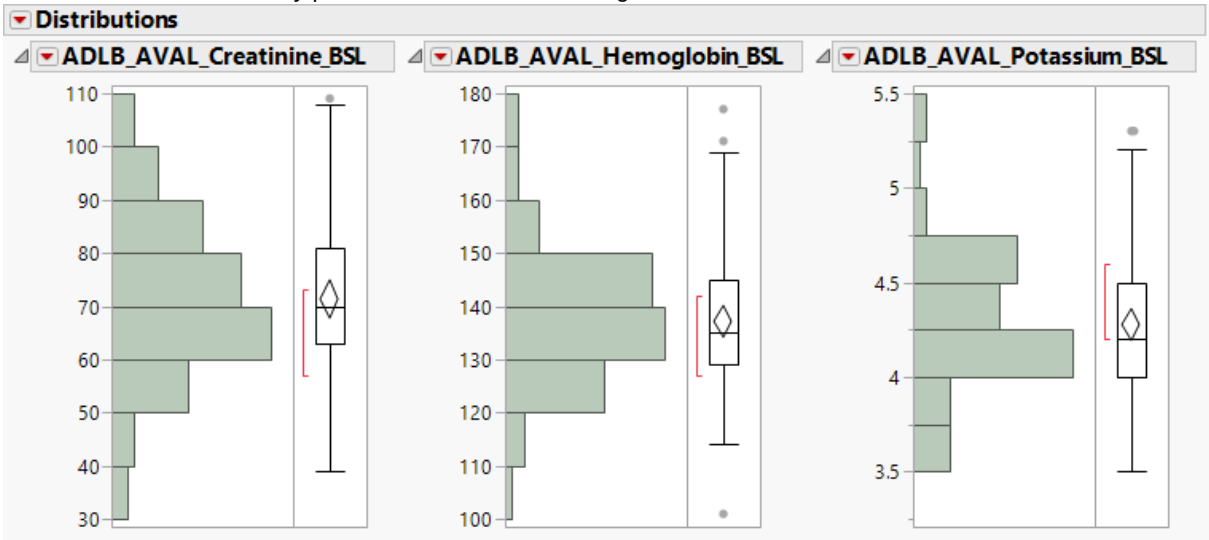
  <Dataset name>_N[_Standardized term variable name]<_ Standardized term code>

  Example:
  ADAE_N_AEDECOD_HEADACHE
  The same concept as for the flags, but the variable is numeric and represents the counts of occurrences of the AE.

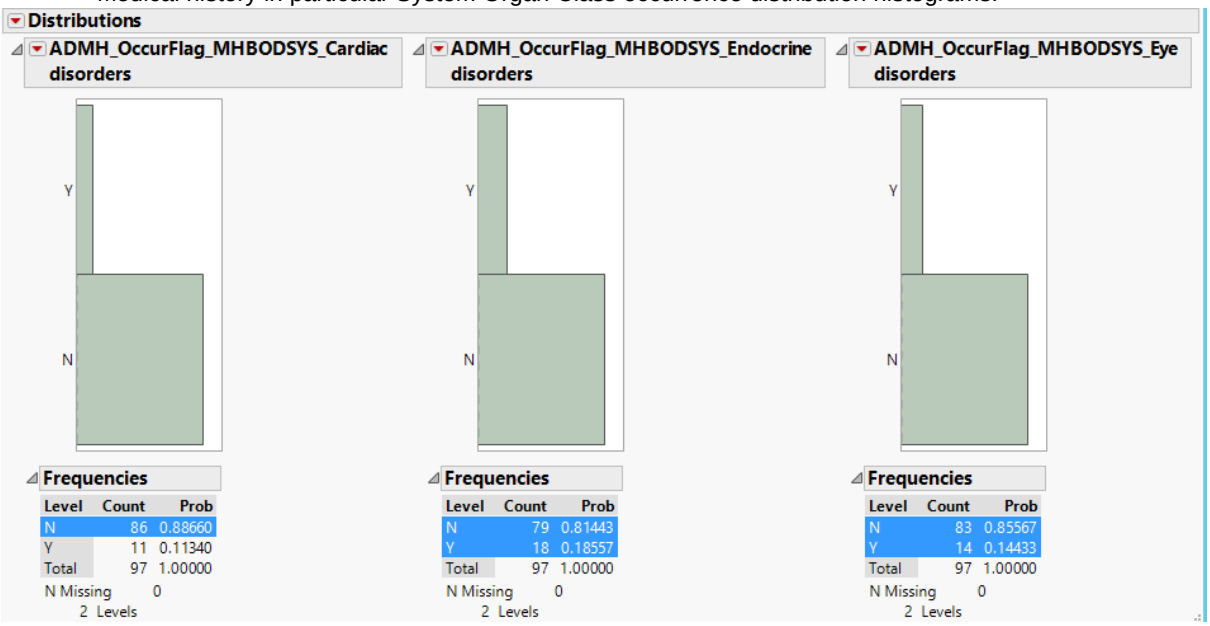
EXAMPLES OF DATA VISUALIZATION AND MODELING BASED ON AXDM

When the AXDM and/or AXDMV in the structure described above are created and stored in the format readable by the data mining or data visualization tool that is to be used, it should be quite straightforward to use it for exploratory analyses.
Below are some screenshots of data visualization performed in SAS JMP® using AXDM:

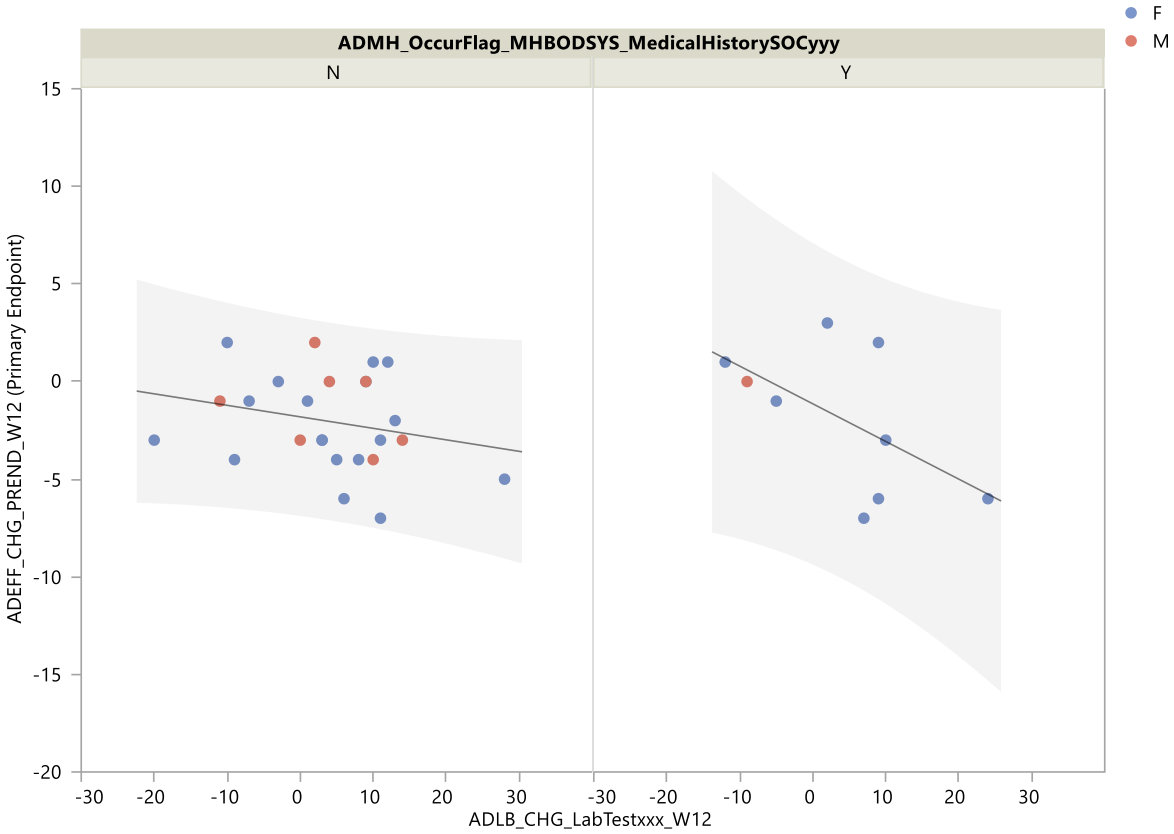
- Baseline laboratory parameters distribution histograms:



- Medical history in particular System Organ Class occurrence distribution histograms:



- Primary endpoint vs change from baseline in laboratory test xxx with coloring by gender:



The convenience here is that all the potentially interesting variables are at hand.

Also modeling with variable selection algorithms can be applied if the action of a study medication on different populations is not yet studied very well.
As an example the following code applies a model with effect selection in the framework of general linear models which is supposed to identify if age, sex, race, medical history event involving any particular body system or baseline creatinine level can explain change from baseline to visit 8 in quality of life score:

```
proc glmselect data=AXDM(where = (RANDFL = "Y" and TRT01P ne "Placebo"));
  class ADMH FL MHBODSYS :;
  model ADEF_EQ5DTOT_V8_CHG = AGE SEX RACE ADLB_CREAT_BSL ADMH_FL_MHBODSYS_: /
  details=all;
run;
```

CREATION OF AXDM

Having defined the types of variables for AXDM it is clear that the SAS code for creation of AXDM would be very similar for different trials. The difference would be only in by-groups and variables to process. Therefore a macro for creation of such a dataset could be one approach. However, a metadata-driven programming approach was deemed to be more flexible in case slightly different processing of the data or additional derivations or manipulations are required. The proposed approach consists of 3 steps:

- creation of metadata for a trial,
- automatic generation of a SAS program based on the metadata,
- manual fine-tuning the SAS program if required and running it.

CREATION OF METADATA FOR A TRIAL

The first step is the creation of metadata for a trial. The metadata that needs to be collected for every ADaM/SDTM dataset are where-clauses (e.g. RANDFL = 'Y' and ANL01FL = 'Y'), by-variables (usually USUBJID), id-variables (e.g. PARAMCD AVISITN), occurrence category variables (for occurrence datasets only e.g. AEDECOD). For subject-level datasets only the where-clause and by-variable(s) are required, for the transposed from BDS –

everything except occurrence categories is required, for the occurrence flags and counts everything except id variables is required. In addition to id-variables, id-label-variables metadata (see IDLABEL below) is used to assign labels to the variables of AXDM.

An example of how the metadata for a trial is collected in an excel spreadsheet is shown below:

MEMNAME	BY	ID	IDLABEL	OCCUR	WHERE	VAR
ADSL	USUBJID				RANDFL = 'Y'	
ADQS	USUBJID	PARAMCD AVISITN	PARAM AVISIT		RANDFL = 'Y' & PARAMCD = 'PAINNOW' & ANL01FL='Y'	CHG
ADVS	USUBJID	PARAMCD AVISITN	PARAM AVISIT		RANDFL = 'Y' & not missing (AVAL)	AVAL
ADMH	USUBJID			MHBODSYS	PARCAT2 = 'MEDICAL HISTORY' & not missing (MHBODSYS)	
ADLB	USUBJID	PARAMCD AVISITN	PARAM AVISIT		RANDFL = 'Y' & not missing (AVAL) & PARCAT2 = 'CENTRAL LABORATORY'	AVAL

This metadata can either be entered into the spreadsheet manually, or a SAS program can be used to scan the contents of the ADaM library and provide a best guess for the metadata (e.g. if there is a PARAMCD variable in a dataset, then add it to id variables, if there is ANLxxFL variables, then add those to the where-clause etc, if there is a xxDECOD/xxBODSYS variable, then add those to the occurrence-variables). After the best guess metadata is created, the user can edit the excel spread sheet to adjust as needed.

AUTOMATIC GENERATION OF THE SAS PROGRAM BASED ON THE METADATA

At the second step another SAS program reads in the metadata file and interprets it into SAS code. The program works with character variables containing sas code, repopulates it with the variables and statements from the metadata file and writes the code into axdm.sas file. Optionally the generated SAS code can be directly submitted, e.g. to depict that the variables are created as expected.

Example of the generated SAS code:

```
data ADSL;
      set temp.ADSL(where =(RANDFL = 'Y'));
run;
%get_dups(ds=ADSL, by= , id=%str());

data ADQS;
      set temp.ADQS(where =(RANDFL = 'Y' & PARAMCD = 'PAINNOW' & ANL01FL='Y'));
      attrib _all_ label=" ";
      length id $40 idlabel $200;
      id=catx("_",trim("ADQS"), "CHG", trim(PARAMCD), trim(AVISITN));
      idlabel=catx("_",trim("ADQS"), "CHG", trim(PARAM), trim(AVISIT));
run;
%get_dups(ds=ADQS, dsout=ADQS_CHG, by= , id=PARAMCD AVISITN,bdupsby=PARAMCD);

proc transpose data = ADQS_CHG_dups (where = (bdups=0)) out=t_ADQS_CHG(drop = _name_);
      by USUBJID;
      id id;
      idlabel idlabel;
      var CHG;
run;
...
```

It is expected that the combination of by-variables and id-variables uniquely identify a record in the parent domain. The SAS program therefore can also identify violations of this assumption which are stored in another output dataset. In case of violations the user should consider updating the source metadata e.g. by adding id-variables or adapting the where-clause and then run the program again.

FINE-TUNING THE AXDM SAS PROGRAM IF REQUIRED AND RUNNING IT

In case some additional derivations or manipulations with the input data are required or the output data has to be converted to a certain format (e.g. csv file), the user can edit the generated axdm.sas code directly in a SAS editor.

After that the axdm.sas program has to be submitted and the AXDM dataset would be ready for use.

EXAMPLE OF THE GENERATED SAS CODE

POSSIBLE ISSUES

- CDISC restriction of variable name length (8 characters) cannot be applied to AXDM

It would be very difficult to follow the standard CDISC restriction of 8 characters for a variable name length for a large wide dataset and also not handy to use afterwards. So the recommendation would be to fit into 32 characters length (limitation for SAS variable names). This would not allow to use XPT transport files for the dataset, other transport formats have to be considered.

- Too many id variables

If too many variables are assigned as id variables - the variable name length in AXDM may exceed 32 characters and then we'll need to handle shortening the length of the variable name. However in a proper CDISC-compliant ADaM dataset PARAMCD has to correspond to only 1 PARCATx and therefore PARCATx might be skipped from id-variables. Also numeric equivalents of the variables (e.g. PARAMN, AVISITN may be chosen instead of the character variables as id variables.

- Multiple record in a by/id group

If multiple rows in one by/id group appear in the input dataset some variables in AXDM will not be created. Ideally this should not happen in a quality CDISC-compliant dataset and thoughtfully developed metadata file. However if such cases occur the user will have to work with the metadata file or with the generated axdm.sas code directly to handle these records. Sometimes the issues occur if the unscheduled visits are not handled properly or the uniqueness of PARAMCD-PARCATx rule is not followed.

CONCLUSION

The described dataset structure is considered by the author to be convenient to do easily customizable data visualization and high level data mining. Also the dataset can be more understandable for people with minimal CDISC standard experience, since all the data for a subject would appear in 1 row. The contents of a certain variable should be clear from the variable name/label since the PARAM, VISIT, ATPT variables are used as id variables when transposing the data.

The data structure will not cover all the data stored in ADaM/SDTM, but covers most of the analysis-enabling data.

The metadata file has to be prepared by a statistical programmer or a biostatistician with good understanding of the particular trial ADaM/SDTM design to be able to correctly identify the parameters and deal with any issues if they arise. However the process is semi-automated and does not take much time since generation of the best guess metadata based on ADaM/SDTM library contents is introduced and the pre-final code for the dataset is generated automatically from the metadata.

REFERENCES

1. SAS Institute Australia Pty Limited (Aug 03, 2016). Data Visualisation The What, Why and How. Whitepaper.
2. https://en.wikipedia.org/wiki/Data_mining
3. https://en.wikipedia.org/wiki/Metadata

CONTACT INFORMATION

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

Author Name: Alexey Kuznetsov
Company: Grünenthal GmbH
Address: Zieglerstr. 6
City / Postcode: Aachen, Germany / 52078
Work Phone: +49-241-569-3424
Email: alexey.kuznetsov@grunenthal.com