

PhUSE US Connect 2019

Paper CT15

Implementation of Imputation Rules and Creating Analysis Flags Following CDISC Guidelines

Venkata N Madhira, Shionogi Inc, Florham Park, USA

Harish Yeluguri, Shionogi Inc, Florham Park, USA

Prabhakara Rao Burma, Acerta Pharma, CA, USA

Abstract

Deriving variables for creating Analysis datasets has always been challenging tasks for programmers, since these derived variables involves complicated algorithms such as implementation of imputation rules from study SAP and deriving analysis flags. These challenging tasks can be easily achieved using some sophisticated SAS® functions. In this paper few examples are given along with code (following CDISC conventions) for derived variables. The examples given in this paper will be useful for programmers to derive some variables in their analysis datasets.

Introduction

Analysis datasets consists of direct copied variables from SDTM datasets and derived variables. Some of the derived variables are Analysis Start Date, Analysis End Date, and Analysis Flags etc... It is most common for raw datasets to have partial dates which will be carried to SDTM datasets. In SDTM level no imputations will be done for timing variables. But in analysis data sets, partial dates will be imputed using some imputation rules based on the study SAP and following CDISC guidelines. The given example in this paper will be helpful for programmer to understand the imputation rules (generic) for the partial start and end dates for adverse events as well as deriving analysis flags for the analysis purpose following CDISC guidelines.

Example I

Imputation rules (generic) for Partial adverse events start date

- (i) If the day part of the adverse event start date is missing, then there would be two scenarios.
 - (a) If the year and month part of adverse event start date is identical with the year and month part of treatment start date (and treatment start date should be less than or equal to adverse event end date if complete end date is available. If day part of adverse event end date is missing and adverse event end year is identical with treatment start date then the month part of treatment start date should be less than or equal to month part of adverse event end date.), then the missing day part of adverse event start date to be imputed with day part of treatment start date.
 - (b) Otherwise, missing day part of adverse event start date can be imputed with first day of the month (YYYY-MM-01).

Note: As per the CDISC guidelines, ASTDTF (Analysis Start Date Imputation Flag) variable has to be populated with value "D" for the observations where the day part of the start date has been imputed.

- (ii) If the month and day part of the adverse event start date is missing, then there would be two scenarios.
 - (a) If the year part of adverse event start date is identical with year part of treatment start date (and the date part of the treatment start date should be less than or equal to date part of adverse event end date if complete end date is available. Else if the day part of the adverse end date is missing and year part of adverse event end date is identical with year part of treatment start date then the month part of treatment start date should be less than or equal to month part or adverse end date.) , then the missing month and day part of adverse event start date to be imputed with month and day part of treatment start date.

- (b) Otherwise missing month and day part of adverse event start date to be imputed with first month and first day of the year (YYYY-01-01).

Note: As per the CDISC guidelines, ASTDTF (Analysis Start Date Imputation Flag) variable has to be populated with value "M" for the observations where the month and day part of the start date has been imputed.

Imputation rules (generic) for Partial adverse events end date

- (i) If the day part of adverse event end date is missing then impute the last day of the month for missing day part of adverse event end date.

Note: As per the CDISC guidelines, AENDTF (Analysis End Date Imputation Flag) variable has to be populated with value "D" for the observations where the day part of the end date has been imputed.

- (ii) If the month and day part of the adverse event end date is missing, then impute last month and last day of the year (YYYY-12-31)

Note: As per the CDISC guidelines, AENDTF (Analysis End Date Imputation Flag) variable has to be populated with value "M" for the observations where the month and day part of the end date has been imputed.

/******Implementation of above date Imputation rules *****/

```
data _1;
  length SUBJID $10 AETERM $ 15 AESTDTC AEENDTC $10;
  subjid="001"; aeterm="Nausea"; aestdte="2018-02-16"; aeendte="2018-02-16"; TRTSDT="15FEB2018"d; output;
  subjid="001"; aeterm="cough"; aestdte="2018-02" ; aeendte="2018-02-20"; trtsdt="15FEB2018"d; output;
  subjid="001"; aeterm="Nausea"; aestdte="2018-02-21"; aeendte="2018-02-22"; trtsdt="15FEB2018"d; output;
  subjid="001"; aeterm="Nausea"; aestdte="2018-02-21"; aeendte="2018-02" ; trtsdt="15FEB2018"d; output;
  subjid="001"; aeterm="fever"; aestdte="2018-02-25"; aeendte="2018-02-26"; trtsdt="15FEB2018"d; output;
  subjid="001"; aeterm="fever"; aestdte="2018-03-14"; aeendte="2018-03-16"; trtsdt="15FEB2018"d; output;
  subjid="001"; aeterm="cough"; aestdte="2018-03" ; aeendte="2018-04-03"; trtsdt="15FEB2018"d; output;
  subjid="001"; aeterm="ache"; aestdte="2018" ; aeendte="2018-03-05"; trtsdt="15FEB2018"d; output;
  subjid="001"; aeterm="ache"; aestdte="2018" ; aeendte="2018-02-15"; trtsdt="15FEB2018"d; output;
  subjid="002"; aeterm="ache"; aestdte="2018-03" ; aeendte="2018-03-15"; trtsdt="12MAR2018"d; output;
  subjid="002"; aeterm="ache"; aestdte="2018-03-17"; aeendte="2018-03" ; trtsdt="12MAR2018"d; output;
  subjid="002"; aeterm="ache"; aestdte="2018-10-15"; aeendte="2019-02" ; trtsdt="12MAR2018"d; output;
  subjid="002"; aeterm="fever"; aestdte="2018-03-30"; aeendte="2018-04" ; trtsdt="12MAR2018"d; output;
  subjid="002"; aeterm="fever"; aestdte="2018-05-20"; aeendte="2018" ; trtsdt="12MAR2018"d; output;
  subjid="002"; aeterm="cough"; aestdte="2018-03" ; aeendte="2018" ; trtsdt="12MAR2018"d; output;
  subjid="003"; aeterm="fever"; aestdte="2018-04" ; aeendte="2018-04" ; trtsdt="11APR2018"d; output;
  subjid="003"; aeterm="fever"; aestdte="2018-06-15"; aeendte="2018-06-20"; trtsdt="11APR2018"d; output;
  subjid="003"; aeterm="fever"; aestdte="2018-07" ; aeendte="2018-08-10"; trtsdt="11APR2018"d; output;
  subjid="004"; aeterm="Nausea"; aestdte="2018-04" ; aeendte="" ; trtsdt="15APR2018"d; output;
  subjid="004"; aeterm="Nausea"; aestdte="2019" ; aeendte="" ; trtsdt="15APR2018"d; output;
  format trtsdt date9.;
run;

data _1;
  set _1;
  st_y=input(scan(AESTDTC, 1, "-"), best.);
  st_m=input(scan(AESTDTC, 2, "-"), best.);
```

```

st_d=input(scan(AESTDTC, 3, "-"), best.);

en_y=input(scan(AEENDTC, 1, "-"), best.);
en_m=input(scan(AEENDTC, 2, "-"), best.);
en_d=input(scan(AEENDTC, 3, "-"), best.);

if length(aeendtc) eq 10 then aendt_num=input(AEENDTC, yymmdd10.);
else if length(aeendtc) eq 7 then aendt_num=input(strip(AEENDTC)||"-"||"01", yymmdd10.);
format aendt_num date9.;

/*--- Missing day part of Start Date Imputation---*/

if nmiss(st_y,st_m) eq 0 and missing(st_d) then do;
    if (st_y eq year(trtsdt)) and (st_m eq month(trtsdt)) and (ifn(length(aeendtc) eq 10,
trtsdt <= aendt_num, ifn(length(AEENDTC) eq 7, year(trtsdt) <= en_y and month(trtsdt) <= en_m,
    ifn(length(AEENDTC) eq 4, year(trtsdt) <= en_y, missing(en_y)))) ) then do;

        st_d=day(trtsdt); ASTDTF="D";
    end;
else do;
    st_d=1; ASTDTF="D";
end;
end;

/*--- Missing Month and Day part of Start Date Imputation---*/

if nmiss(st_m, st_d) eq 2 and ^missing(st_y) then do;
    if (st_y eq year(trtsdt)) and (ifn(length(aeendtc) eq 10, trtsdt <= aendt_num,
ifn(length(AEENDTC) eq 7, year(trtsdt) <= en_y and month(trtsdt) <= en_m,
    ifn(length(AEENDTC) eq 4, year(trtsdt) <= en_y, missing(en_y)))) ) then do;
        st_m=month(trtsdt); st_d=day(trtsdt); ASTDTF="M";
    end;
else do;
    st_m=1; st_d=1; ASTDTF="M";
end;
end;

ASTDT=input(catx("-", put(st_y, best.), put(st_m, z2.), put(st_d,z2.)), yymmdd10.);

/*--- Missing Month and/or Day part of End date Imputation ---*/

if missing(en_d) and nmiss(en_y, en_m) eq 0 then do;
    AENDT=intnx("month", intnx("month", aendt_num, -1, "e"), +1, "e");
    AENDTF="D";
end;

else if nmiss(en_m, en_d) eq 2 and ^missing(en_y) then do;
    AENDT=input(strip(put(en_y,best.)||"-"||"12-31", yymmdd10.);
    AENDTF="M";
end;

if missing(AENDT) and length(AEENDTC) eq 10 then AENDT=input(AEENDTC, yymmdd10.);

```

```

format ASTDT AENDT date9.;
run;

proc sql noprint;
    create table _2 as
    select SUBJID, AETERM, AESTDTC, AEENDTC, TRTSDT, ASTDT, ASTDTF, AENDT, AENDTF
    from _1 ;
quit;

```

```

/*****

```

Example II

Flag for first occurrence of an event in a row of consecutive occurrences.

```

data cars;
    set sashelp.cars;
    by make;
    if first.make then AVISITN=.;
    AVISITN+1;
    VISIT= "Visit" || strip(put (AVISITN, best.));
    if first.make then id+1;
    SUBJID=strip(put(id,z3.));
    AVAL=MPG_Highway;
    PARAM="PARAM1";
    keep SUBJID AVAL visit AVISITN PARAM subjid;
run;
/*pick up records for consecutive AVAL ge 25 in each SUBJID within PARAM */
data _1;
    set cars;
    by SUBJID;
    if first.SUBJID then x1=.;
    if AVAL ge 25 then x1+1; /*-----AVAL value can be changed as per the requirement-----*/
    else x1=0;
run;
proc sort data=_1;
    by SUBJID descending AVISITN visit;
run;
data _2;
    set _1;
    by SUBJID descending AVISITN visit;
    retain c2 ;
    if x1=0 then c1=0;
    else c1=1;
    if first.SUBJID then c2=.;
    c2=sum(c2,x1)*c1;
    if x1 gt 0 and x1 ne c2 then c3+x1;
    else c3=0;
    if c1= 1 then ret1=c2-c3;
    if x1 eq 1 and ret1 ge 3 then ANL01FL="Y"; /*----RET1 variable value can be changed here for the number
of continuous occurrences of analysis value-----*/
    *if ret1 ge 3;
    keep SUBJID aval AVISITN visit anl01fl param;
run;

```

```
proc sql noprint;
    create table _3 as
    select subjid,param, AVISITN, visit, aval, anl01fl
    from _2
    order by SUBJID, AVISITN, visit ;
quit;
```

Example III

Impute missing value with 1, if the previous and next visit (with respect to the visit with missing analysis value) have value 1 otherwise 0. Where 1(=yes) and 0 (=no) are the response values for AVAL.

```
data x;
    SUBJID=1; AVAL=1;output;
    SUBJID=1; AVAL=1;output;
    SUBJID=1; AVAL=.;output;
    SUBJID=1; AVAL=1;output;
    SUBJID=1; AVAL=0;output;
    SUBJID=1; AVAL=0;output;
    SUBJID=1; AVAL=.;output;
    SUBJID=1; AVAL=0;output;
    SUBJID=1; AVAL=0;output;
    SUBJID=1; AVAL=.;output;
    SUBJID=1; AVAL=1;output;
    SUBJID=1; AVAL=.;output;
    SUBJID=1; AVAL=1;output;
    SUBJID=1; AVAL=.;output;
    SUBJID=1; AVAL=.;output;
    SUBJID=1; AVAL=1;output;
    SUBJID=2; AVAL=1;output;
    SUBJID=2; AVAL=1;output;
    SUBJID=2; AVAL=.;output;
    SUBJID=2; AVAL=.;output;
    SUBJID=2; AVAL=1;output;
    SUBJID=2; AVAL=0;output;
    SUBJID=2; AVAL=0;output;
    SUBJID=2; AVAL=.;output;
    SUBJID=2; AVAL=0;output;
    SUBJID=2; AVAL=.;output;
    SUBJID=2; AVAL=.;output;
    SUBJID=2; AVAL=1;output;
    SUBJID=2; AVAL=1;output;
    SUBJID=2; AVAL=.;output;
    SUBJID=2; AVAL=1;output;
run;

proc sort data=x out=_1;
    by SUBJID;
run;

data _1a;
    set _1;
    by SUBJID;
    retain x ;
    if first.SUBJID then do; x=.; seq1=.;AVISITN=.;end;
```

```

        if AVAL ne . then x=AVAL;
        if AVAL eq . then seq1+1; else seq1=.;
        AVISITN+1;
        *AVISITN=_n_;
run;

proc sort data=_1a out=_2a;
    by SUBJID descending AVISITN;
run;

data _2a1;
    set _2a;
    by SUBJID descending AVISITN;
    retain y ;
    if first.SUBJID then do; y=.; seq2=.;end;
    if AVAL ne . then y=AVAL;
    if AVAL eq . then seq2+1; else seq2=.;
run;

proc sort data=_2a1;* out=_2a2;
    by SUBJID AVISITN;
run;

data _2a1;
    set _2a1;
    if AVAL eq . and x eq 1 and y eq 1 and seq1 eq 1 and seq2 eq 1 then imp=1;
    if AVAL eq . and imp ne . then do;AVAL = imp; IMP01FL="Y"; end;
    else if AVAL eq . and imp eq . then do;AVAL = 0; IMP01FL="Y"; end;
    keep SUBJID AVAL AVISITN IMP01FL;
run;

proc sql noprint;
    create table _3 as
    select SUBJID, AVISITN, AVAL, IMP01FL
    from _2a1;
quit;

/*-----Generating same results using SQL-----*/

data sql1;
    set _1a;
    keep SUBJID AVAL AVISITN;
run;

proc sql noprint;
    create table sql2 as
    select a.*, (select AVAL as lagval from sql1 where (AVISITN=a.AVISITN-1) and (SUBJID eq a.SUBJID)),
    (select AVAL as leadval from sql1 where (AVISITN=a.AVISITN+1) and (SUBJID eq a.SUBJID))
    from sql1 as a;
quit;

data sql3;
    set sql2;

```

```

if missing(AVAL) and _TEMA001 eq 1 and _TEMA002 eq 1 then Impval=1;
if AVAL eq . and impval eq 1 then do; AVAL=impval; IMP01FL="Y";end;
else if AVAL eq . then do;AVAL=0; IMP01FL="Y";end;
keep SUBJID AVISITN AVAL IMP01FL;

run;

```

Example IV

Pick the first response from sustained response (until end of the study). Where 1 =yes and 0 =no are the response values for AVAL.

```

data sustain;
  SUBJID=1; AVAL=1;output;
  SUBJID=1; AVAL=1;output;
  SUBJID=1; AVAL=.;output;
  SUBJID=1; AVAL=1;output;
  SUBJID=1; AVAL=0;output;
  SUBJID=1; AVAL=0;output;
  SUBJID=1; AVAL=0;output;
  SUBJID=2; AVAL=1;output;
  SUBJID=2; AVAL=1;output;
  SUBJID=2; AVAL=.;output;
  SUBJID=2; AVAL=.;output;
  SUBJID=2; AVAL=1;output;
  SUBJID=2; AVAL=1;output;
  SUBJID=2; AVAL=1;output;
  SUBJID=3; AVAL=0;output;
  SUBJID=3; AVAL=0;output;
  SUBJID=3; AVAL=1;output;
  SUBJID=3; AVAL=1;output;
  SUBJID=3; AVAL=1;output;
  SUBJID=3; AVAL=1;output;
  SUBJID=3; AVAL=1;output;
  SUBJID=3; AVAL=1;output;
  SUBJID=3; AVAL=1;output;
  SUBJID=4; AVAL=1;output;
  SUBJID=4; AVAL=0;output;
  SUBJID=4; AVAL=1;output;
  SUBJID=4; AVAL=0;output;
  SUBJID=4; AVAL=0;output;
  SUBJID=4; AVAL=1;output;
  SUBJID=4; AVAL=1;output;
  SUBJID=4; AVAL=1;output;

run;

data sustain;
  set sustain;
  by subjid;
  if first.subjid then AVISITN =.;
  AVISITN+1;

run;

proc sort data=sustain out=sustain1;
  by subjid descending AVISITN;

run;

```

```

data sustain2;
    set sustain1;
    retain x;
    by subjid;
    if first.subjid then x=1;
    x=aval*x;
    if x=1;
run;

```

```

data sustain2;
    set sustain2;
    by subjid x;
    if last.x;
run;

```

CONCLUSION

The examples given in this paper are helpful for the programmers to understand the imputation rules (generic) for adverse events (start and end) partial dates, some of the analysis flags derivations following CDISC guidelines for the submission purpose.

REFERENCES

Analysis Data Model Implementation Guide Version 1.1
 SAS ®Certification Prep Guide Advanced Programming for SAS ®9
<https://www.lexjansen.com/wuss/2012/28.pdf>
<https://www.phusewiki.org/docs/2006/P011.pdf>
https://www.sas.com/storefront/aux/en/splongitudinal/58176_excerpt.pdf

ACKNOWLEDGMENTS

I would like to thank Malla Reddy Boda, Associate Director, Shionogi Inc, for his review and providing invaluable comments.

CONTACT INFORMATION

Venkata N Madhira
 E-mail: venkatanmadhira@gmail.com
 Sr. Statistical Programmer, Consultant at Shionogi Inc, NJ, USA

Harish Yeluguri
 E-mail: harish.yeluguri@gmail.com
 Sr. Statistical Programmer, Consultant at Shionogi Inc, NJ, USA

Prabhakara Rao Burma
 E-mail: prsas85@gmail.com
 Principal Clinical SAS Analyst, Acerta Pharma, CA, USA

SAS ® and all other SAS ® Institute Inc. product or service names are registered trademarks or trademarks of SAS ® Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.