

The Inevitable Use of RETAIN Statement in Multiple Scenarios with Examples for Creating Efficacy Parameters

Prabhakara Rao Burma, Acerta Pharma, CA, USA
Venkata N Madhira, Shionogi Inc, NJ, USA

ABSTRACT

Imputation techniques and deriving analysis flags plays an indispensable role while deriving analysis datasets. By following study SAP, for a given parameter, a value to be assigned to the missing AVAL (Analysis value). Retaining previous observation (analysis value) to the current observation is one of the recommended methods for imputing a value for missing analysis value. In this paper few examples have been presented where the retain statement is inevitable for populating missing analysis value by conditionally imputing the collected value and deriving analysis flags.

INTRODUCTION

In any clinical trial irrespective of therapeutic area, sponsor look for the safety as well as efficacy outcomes such as primary and secondary objective results to get their drug approved. To assess the efficacy outcomes of the study at different visits for all treated subjects should have values for the efficacy parameters in their corresponding visits. But it is most common for subjects having missing analysis values at different time points. There have been multiple imputation rules (based on the study SAP) to populate analysis value at missing visits to support the corresponding analysis. Some analysis flags are also to be derived for the first occurrence of an event in consecutive occurrences etc... "RETAIN" is the most powerful statement to populate missing analysis values by following multiple imputation rules. The inevitable use of RETAIN statement is discussed in this paper with examples by taking multiple challenging scenarios.

EXAMPLE-I

```
data cars;
  set sashelp.cars;
  by make;
  if first.make then visit_n=.;
  VISITN+1;
  VISIT= "Visit"||strip(put (visitn, best.));
  if first.make then id+1;
  SUBJID=strip(put(id,z3.));
  AVAL=MPG_Highway; PARAM="PARAM1";
  keep SUBJID AVAL visit visitn PARAM subjid;
run;

/*---Pick up records for consecutive AVAL ge 25 in each SUBJID for 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 visitn visit;
run;

data _2;
  set _1;
  by SUBJID descending visitn 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 changed here for the number of continuous occurrences of
analysis value---*/ *if ret1 ge 3;
keep SUBJID aval visitn visit anl01fl param;
run;

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

```

EXAMPLE-II

Imputing missing value with 1, if the previous and next visits have value 1 otherwise 0. Where 1 (=yes) and 0 (=no) are the response values for AVAL.

```
/*-----Imputing missing value as 1 if previous and next visit has value 1-----*/
```

```

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

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

data _2a1;
  set _2a;
  by SUBJID descending VISITN;
  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;
  by SUBJID VISITN;
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 VISITN IMP01FL;
run;

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

```

EXAMPLE-III

Pick the first observation from which sustained response occurred. Where 1 (=yes) and 0 (=no) are the response values for AVAL.

```

/*---Pick the first observation at which Sustained Response occurred (where AVAL=1
RESPONSE=YES, AVAL=0 RESPONSE=NO---*/

```

```

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=1 ;output ;
SUBJID=4; AVAL=1 ;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 VISITN =.;
  VISITN+1;
run;

proc sort data=sustain out=sustain1;
  by subjid descending visitn;
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 helps programmer to understand the use of RETAIN statement in multiple scenarios.

REFERENCES

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

Prabhakara Rao Burma
E-mail: prsas85@gmail.com
Lead SAS Programmer,
Consultant at Acerta Pharma, CA, USA

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