

Proceedings

First SUGCDSC Conference

August 31, 2018

Silver Spring Civic Building at Veterans Plaza
8525 Fenton Street, Silver Spring, MD 20910
Washington DC Area, USA



SAS User Group for Clinical Data Standards and Codes (SUGCDSC)

PO Box 1032, Rockville, MD 20849, USA

<https://sugcdsc.org>

SUGCDSC 2018

Thanks for Visiting: Tips and Tricks for Creating and Using the Subject Visits Domain

David Van't Hof, QST Consultations, LTD., Allendale, MI

ABSTRACT

The pooling of clinical trial data for the creation of the Subject Visits domain presents many challenges. Clinical trials often collect multiple assessment dates and/or datetimes at each visit which should be checked for accuracy as well as the need for the creation of date ranges for visits. In addition, EDC systems often restrict forms available at scheduled visits to planned assessments and require the creation of an unscheduled visit to submit unplanned assessments that occur at scheduled visits. This leads to assessments completed on the same date being tied to both scheduled and unscheduled visits. Another challenge with clinical trial data is the pooling of vendor data, such as laboratory and electrocardiogram transfers, with EDC data. Vendor data often uses differently terminology for visits and has limitations on the visit values that are exported.

This paper will detail how to pool the data together and account for the challenges described above. The paper will include methods to account for a singular date that is tied to multiple visits, visits with date ranges, unscheduled assessments at scheduled visits, and differences between vendor and EDC data. The paper will also describe validity checks on visit dates as well as how to use the Subject Visits domain when processing by visit domains to ensure more accurate data submitted to SDTM.

INTRODUCTION

Clinical trials generally have multiple sites and multiple users entering data into the electronic data capture (EDC) system. In addition, multiple assessment dates and/or datetimes are often collected at each visit. The data entry process ultimately involves entering multiple date fields that should be consistent, but we are bound to have some data entry errors anytime you require users to enter dates multiple times. Adding to the complexity of the data entry errors is the inconsistent handling of common situations between multiple users. A common error of this style would be when an assessment is missed at one visit, completed at the following, and entered into the system with the date of the later visit, but entered on the CRF page of the earlier visit. Clinical trials also involve submitting data from vendor transfers for assessments such as laboratory tests and electrocardiograms. These vendor transfers often have limitations on how the visit field can be populated. The most common scenario is where the visit field can only be populated with visits where the assessment was planned and everything else is indicated as unscheduled. This paper explains how to best create the Subject Visits (SV) domain, apply checks in the programming of the SV domain to help clean the data, and utilize the SV domain in the programming of findings domains.

BUILDING THE SV DOMAIN

To best utilize the SV domain for efficiency and accuracy in programming findings domains, it should be programmed prior to programming any of the findings domains. The first step in programming the SV domain is to pool all EDC and vendor datasets that contain a visit indication and a date of assessment or visit. The datasets that contain visit information are pooled using a SET statement. There are often "not done" records in the assessment datasets that do not have the date of assessment populated. We do not want to use the records where the date of assessment is not populated when pooling all visit information. We can restrict our pooled dataset to records where the date of visit or assessment is populated by renaming the date variables in the SET statement and using a WHERE clause to eliminate the records where the date variables are missing. The SET statement should also make use of the IN dataset option to track the source of the visit records as we process the dataset. The variables created by the IN dataset option are temporary Boolean variables that are automatically dropped by the data step, but you will want to keep the source location for later data steps. Two arrays can be used to populate a source location variable based on the IN dataset Boolean variables. The first array (inloc) represents the IN dataset Boolean variables and the second array (dloc) is a character array initialized to the names of the

corresponding datasets. A DO loop is used to process through each of the possible locations to populate the source location variable (loc). The final task in the data step is to populate the visit number (VISITNUM) variable. VISITNUM should be populated as the chronological order of the visit for scheduled visits. To help with sorting, VISITNUM's for unscheduled visits should be temporarily set to a value exceeding those for scheduled visits, but these values are changed later in the code.

```

data allvisits1;
  length usubjid $40. visit $200. dtc $19.;
  set i_ape (in=ape rename=(pedt=dtc)) i_appl (in=appl rename=(exstdt=dtc))
  i_chem (in=chem rename=(lbdtd=dtc)) i_dov (in=dov rename=(visdt=dtc))
  i_egss (in=egss rename=(egssdt=dtc)) i_hem (in=hem rename=(lbdtd=dtc))
  i_udov (in=udov rename=(visdt=dtc)) i_vs (in=vs rename=(vsdt_dts=dtc))
  edt_lb (in=EDT rename=(lbcoldt=dtc));

  where dtc ne '';

  /*identifying the source data location of each record*/
  array inloc{9} ape appl chem dov egss hem udov vs edt;
  array dloc{9} $10. ("APE", "Appl", "Chem", "DOV", "EGSS", "Hem", "UDOV",
                    "VS", "EDT");

  do i=1 to 9;
    if inloc{i}=1 then loc=dloc{i};
  end;

  /*standardizing subject and visit variables from EDC datasets*/
  if EDT=0 then do;
    usubjid="%studyid.-"||strip(subjectnumber);
    visit=visitid;
    if visit='UNS' then visit='UNSCHEДУLED';
  end;

  /*standardizing subject, visit, and date variables from EDT lab
  dataset*/
  else if EDT then do;
    usubjid="%studyid.-"||strip(subjectid);
    visit=visitname;
    dtc=scan(dtc, 1, 'T');
  end;

  if visit='SCREENING' then visitnum=1;
  else if visit='BASELINE' then visitnum=2;
  else if visit='WEEK 4' then visitnum=3;
  else if visit='WEEK 8' then visitnum=4;
  else if visit='WEEK 12' then visitnum=5;
  else if visit='UNSCHEДУLED' then visitnum=10;
  /*confirming there are no unexpected visit values in any of the
  datasets*/
  else put 'QST ERR' 'OR: Visit not accounted for ' usubjid= loc= visit= ;

  keep usubjid loc visit visitnum dtc;
run;

```

The next step is to sort the dataset by subject, date, visit number, and location. After sorting, the dataset is processed to get one record per subject per date per visit number. As the dataset is processed down to this unique set of records, the source data locations should be concatenated together into a single variable to aid with presenting data issues later on. A common issue to be concerned with in this data step, are dates that are identified with more than one visit. Unscheduled visits may arise on the same

date as scheduled visits due to unplanned assessments completed during scheduled visits. EDC systems frequently restrict forms available at scheduled visits to planned assessments, and this leads to unplanned assessments being submitted under unscheduled visits on the same date as scheduled visits. It is also common for vendor transfers to have limitations on the visit field that only allow the field to be populated with visits where the assessments were planned or the field is populated as unscheduled. Therefore when unscheduled visits are in the data on the same day as scheduled visits, it is common practice to remove the unscheduled visits and submit only the scheduled visits. This data step also includes a check for multiple scheduled visits that are indicated on the same day. The code below prints a message to the log indicating multiple scheduled visits on the same date with information on the subject, date, visits, and sources of the records. Once we have addressed the dates with multiple visit indications, we can start to process our proper set of observations. Unscheduled visits are processed to use visit numbers and names that provide information as to when they occurred. For example, unscheduled visits that occur prior to baseline are considered an extension of screening and mapped to the screening visit. Unscheduled visits that occur after the baseline visit will have visit number set to the prior visit number plus .1, and the visit description is "UNSCHEDULED VISIT x" where x represents the visit number. This data step also includes a check for visits that occur out of order. Visits often appear out of order when assessments are missed at visits, completed at a follow up visit, but entered with the earlier visit indication. A visit order check can identify these issues, so they can be addressed with queries and updates to the EDC data.

```

proc sort data=allvisits1;
  by usubjid dtc visitnum loc;
run;

data allvisits2;
  length locs pvisit plocs pdtc $200.;
  set allvisits1;
  by usubjid dtc visitnum loc;
  retain pvisit pvisitn plocs pdtc locs;
  /*only processing the first record for each combination of subject, date,
  visit, and location*/
  if first.loc;
  if first.usubjid then do;
    pvisit='';
    pvisitn=.;
    plocs='';
    pdtc='';
  end;
  /*creating a variable with all locations for each combination of subject,
  date, and visitnum*/
  if first.visitnum then locs=loc;
  else locs=strip(locs)||'|'; '||loc;
  if last.visitnum then do;
    /*Identifying dates that indicated for multiple visits*/
    if dtc=pdtc then do;
      /*Unscheduled visits on the same date as scheduled visits will be
      removed*/
      if visit='UNSCHEDULED' then delete;
      /*Printing an error message to the log for all dates that are
      indicated for multiple scheduled visits*/
      else put 'QST ERR' 'OR: Date indicated for multiple visits ' usubjid=
        dtc= pvisit= plocs= visit= locs= ;
    end;
    if visitnum<pvisitn then put 'QST ERR' 'OR: visit is out of order for '

```

```

usubjid= pvisitn= visitnum= locs= ;
/*processing unscheduled visits*/
if visit='UNSCHEDULED' then do;
  /*assuming all unscheduled visits prior to baseline will be an
  extension of screening*/
  if pvisit in ('SCREENING', '') then do;
    visit='SCREENING';
    visitnum=1;
  end;
else do;
  /*identifying unscheduled visits relative to the last scheduled
  visit*/
  visitnum=pvisitn+.1;
  visit='UNSCHEDULED VISIT '||strip(visitnum);
end;
end;
output;
pvisit=visit;
pvisitn=visitnum;
plocs=locs;
pdtc=dtc;
end;
drop p:;
run;

```

The last step in building the SV domain is to process the data to create one record *per* subject *per* visit. There are some visits such as the screening visit where it is possible to span multiple dates. Visits that are completed over multiple dates should have the start date of the visit (SVSTDTC) set to the earliest date of the visit and the end date of the visit (SVENDTC) set to the latest date of the visit. The start and end dates for these visits are calculated with a RETAIN statement and some BY processing rules.

```

proc sort data=allvisits2;
  by usubjid visitnum dtc;
run;

data sv;
  length svstdtc svendtc $19.;
  set allvisits2;
  by usubjid visitnum dtc;
  retain svstdtc;
  if first.visitnum then svstdtc=dtc;
  if last.visitnum then do;
    svendtc=dtc;
  end;
  output;
end;
run;

```

UTILIZING THE SV DOMAIN

Once the SV domain is finished, it can be used to efficiently populate the visit variables in findings domains. The SV domain was programmed by pooling all EDC and vendor data, so any visit mapping that will need to happen in a findings domain was already completed in the programming of SV. Since it is possible for the SV domain to contain ranged visits, it first needs to be converted to a dataset with one record *per* subject *per* visit *per* possible date of visit. This can be accomplished by processing through a DO loop on SVSTDTC and SVENDTC in the final SV dataset.

```

data visits_by_date;
  set sv;
  _start=input(svstdtc, yymmdd10.);
  _end=input(svendtc, yymmdd10.);
  do i=_start to _end;
    dtc=put(i, yymmdd10.);
    output;
  end;
  keep usubjid dtc visit;;
run;

```

After the SV domain has been converted it can then be merged into any findings domain by subject and date to populate the standard SDTM visit variables VISIT, VISITNUM, and VISITDY. In the sample code below, the converted SV dataset (visits_by_date) is merged into a vital signs dataset (vs1) to populate the visit variables. Two possible issues to keep in mind are: 1) dates variable in the findings dataset that can include a time component and 2) any variables in the findings dataset that use a standard SDTM visit variable name. Findings datasets that include a time component in the date variable should have an additional variable created that contains only date information. If the findings dataset contains a variable that uses a standard SDTM visit variable name, then it should be renamed or dropped prior to merging with the converted SV dataset. The findings dataset should be sorted by subject and date and merged with the converted SV dataset. When merging with converted SV dataset, make use of the IN dataset option to only keep records for dates in the findings dataset.

```

proc sort data=findings1;
  by usubjid dtc;
run;

data findings2;
  merge findings1 (in=A) visits_by_date;
  by usubjid dtc;
  /*restricting to records found in the findings dataset*/
  if A;
run;

```

CONCLUSION

Clinical trial data is prone to inconsistencies with date and visit information. Programmers can easily address these issues by following the steps described in this paper. This in turn will result in more accurate data and more efficient programs.

REFERENCES

Study Data Tabulation Model Implementation Guide: Human Clinical Trials Version 3.2
<https://www.cdisc.org/standards/foundational/sdtmig>.

CONTACT INFORMATION

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

David Van't Hof, M.S.
 QST Consultations, Ltd.
 DVantHof@qstconsultations.com
 www.qstconsultations.com

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.