**Common SDTM Variables**

Crash course in CDISC SDTM variables

**Rules**

1. **All raw variables**
2. **Add any required SDTM compliance variables, ex. Topic**
3. **Use VDT as metadata for variable attributes**
4. **No user defined formats applied**
5. **Apply formula for common SDTM variables**
6. **Most all datasets, except DM, are vertical structure**
7. **Common key variables : USUBJID, XXTESTCD, XXDTC, XXCOD**

**Hierarchy of Grouping Variables**

**STUDYID/DOMAIN**

**XXCAT, ex. Behavior Rating**

**XXSCAT, ex. Parent Form**

**USUBJID**

**XXGRPID**

**XXTESTCD, ex. BRFP111**

**XXTEST, ex. ITEM1**

**STUDYID/USUBJID/DOMAIN – PKV016/XXXXXX-XXXX-XXXX/QS**

studyid = 'XYZ123';

domain = 'LB';

usubjid = strip(uniqueid);

**XXSEQ – SEQ # by key variables**

proc sort data=ex;

by usubjid exstdtc;

run;

data ex;

set ex;

by usubjid exstdtc;

if first.usubjid then exseq=1;

else exseq + 1;

run;

**XXTESTCD/XXTEST – Coded value/Formatted value [$XXTESTCD., $XXTEST.]**

lbtest = put(labtest, $lbtest\_labs\_labtest.);

lbtestcd = put(labtest, $lbtestcd\_labs\_labtest.);

**XXCAT/XXSCAT – Category/Subcategory**

lbcat = put(labcat,$lbcat\_labs\_labcat.);

**XXORRES/XXSTRESC/XXSTRESN – Original result/$ Standard Unit/ N Standard Unit [BEST.]**



\*\*\*\* store original results;

lborres = strip(put(nresult, best.));

lborresu = strip(colunits);

lbornrlo = strip(put(lownorm, best.));

lbornrhi = strip(put(highnorm, best.));

\*\*\*\* create standardized results;

lbstresc = lborres;

lbstresn = nresult;

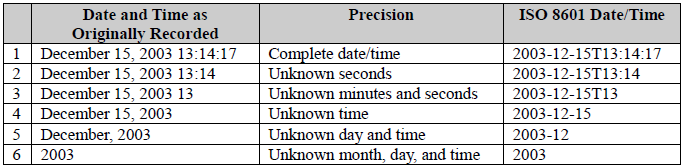
lbstresu = lborresu;

lbstnrlo = lownorm;

lbstnrhi = highnorm;

**XXDTC – Date/Time [YYYY-MM-DD, ISO 8601]**

**Process ISO8601 Date/Time as a SAS date variable – INPUT(SVSTDTC, E8601DA.)**



if (&second ne .) then

&dtcdate = put(&year,z4.) || "-" || put(&month,z2.) || "-"

|| put(&day,z2.) || "T" || put(&hour,z2.) || ":"

|| put(&minute,z2.) || ":" || put(&second,z2.);

else if (&minute ne .) then

&dtcdate = put(&year,z4.) || "-" || put(&month,z2.) || "-"

|| put(&day,z2.) || "T" || put(&hour,z2.) || ":"

|| put(&minute,z2.);

else if (&hour ne .) then

&dtcdate = put(&year,z4.) || "-" || put(&month,z2.) || "-"

|| put(&day,z2.) || "T" || put(&hour,z2.);

else if (&day ne .) then

&dtcdate = put(&year,z4.) || "-" || put(&month,z2.) || "-"

|| put(&day,z2.);

else if (&month ne .) then

&dtcdate = put(&year,z4.) || "-" || put(&month,z2.);

else if (&year ne .) then

&dtcdate = put(&year,z4.);

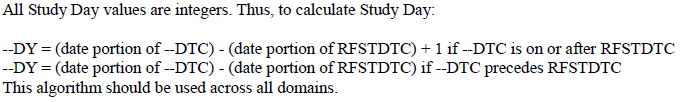
else if (&year = .) then

&dtcdate = "";

\*\* remove duplicate blanks and replace space with a dash;

&dtcdate = translate(trim(compbl(&dtcdate)),'-',' ');

**XXDY – Study Day based on first dose date, if visit date is after dose date then add 1, else do not.**



**Option # 1**

case

when a.qsdtn > . and b.fdosedt > . and a.qsdtn >= fdosedt then (datepart(qsdtn)-datepart(fdosedt) + 1)

when a.qsdtn > . and b.fdosedt > . and a.qsdtn < fdosedt then (datepart(qsdtn)-datepart(fdosedt))

else .

end as qsdy

**Option # 2**

if length(&date) >= 10 and length(&refdate) >= 10 then

do;

if input(substr(%substr("&date",2,%length(&date)-3)dtc,1,10),yymmdd10.) >=

input(substr(%substr("&refdate",2,%length(&refdate)-3)dtc,1,10),yymmdd10.) then

%upcase(%substr("&date",2,%length(&date)-3))DY = input(substr(%substr("&date",2,%length(&date)-3)dtc,1,10),yymmdd10.) -

input(substr(%substr("&refdate",2,%length(&refdate)-3)dtc,1,10),yymmdd10.) + 1;

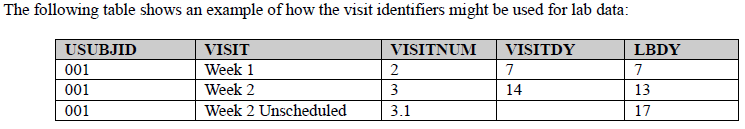
else

%upcase(%substr("&date",2,%length(&date)-3))DY = input(substr(%substr("&date",2,%length(&date)-3)dtc,1,10),yymmdd10.) -

input(substr(%substr("&refdate",2,%length(&refdate)-3)dtc,1,10),yymmdd10.);

end;

**VISITNUM/VISIT/VISITDY – Visit #/Label/Day [$VISLIST]**



visitnum = month;

visit = put(month, visit\_labs\_month.);

**XXBLFL – Baseline flag [N, Y, ] based on visits before or on first dose date, i.e. study day <= 1**

**Option # 1**

\* Sort by study variable, LBDY, at the end;

proc sort data=lb4;

by usubjid lbcat lbtestcd lbspec lbmethod visitnum **lbdy**;

run;

\* Create phase variable based on acceptable study day values;

data lb5;

set lb4;

by usubjid lbcat lbtestcd lbspec lbmethod visitnum lbdy;

**if lbdy <= 1 then phase = 0; else phase = 1;**

run;

\* Sort by study variable, LBDY, and PHASE at the end;

proc sort data=lb5;

by usubjid lbcat lbtestcd lbspec lbmethod visitnum **lbdy** **phase**;

run;

Add new baseline flag variable, LBBLFL, based on LAST.PHASE and PHASE=0;

data lb6;

set lb5;

by usubjid lbcat lbtestcd lbspec lbmethod visitnum lbdy phase;

if last.phase and phase = 0 then lbblfl = 'Y'; else lbblfl='';

drop phase;

run;

**Option # 2a**

\* Sort with the study day variable, QSDY at the end;

proc sort data=adhdt;

by usubjid qscat qsscat qstestcd **qsdy**;

run;

\* Group without the study day variable, QSDY. Create baseline flag variable, QSBLFL;

proc sql;

create table bflag as

select usubjid, qscat, qsscat, qstestcd, min(qsdy) as bflagdy, **‘Y’ as qsblfl**

from adhdt where qsorres ne '' and . < qsdy <=1

group by usubjid, qscat, qsscat, qstestcd;

quit;

proc print data=bflag (obs=200);

where usubjid='PKU016-0123-1004'; run;

proc freq data=bflag;

tables usubjid\*qscat\*qsscat\*qstestcd\*qsdy\*bflagdy/list missing;

run;

\* Many to One join with the study day variable, QSDY, and the baseline flagged study day variable, BFLAGDY. Add new baseline flag variable, QSBLFL;

proc sql;

create table adhd2 as

select a.\*, b.qsblfl

from adhdt as a left join bflag as b

on a.usubjid=b.usubjid and a.qscat=b.qscat and a.qsscat=b.qsscat and a.qstestcd=b.qstestcd and **a.qsdy=b.bflagdy**;

quit;

**Option # 2b**

\* Sort with the study day variable, QSDY at the end;

proc sort data=adhdt;

by usubjid qscat qsscat qstestcd **qsdy**;

run;

\* Group without the study day variable, QSDY.

proc sql;

create table bflag as

select usubjid, qscat, qsscat, qstestcd, min(qsdy) as bflagdy

from adhdt where qsorres ne '' and . < qsdy <=1

group by usubjid, qscat, qsscat, qstestcd;

quit;

proc print data=bflag (obs=200);

where usubjid='PKU016-0123-1004'; run;

proc freq data=bflag;

tables usubjid\*qscat\*qsscat\*qstestcd\*qsdy\*bflagdy/list missing;

run;

\* Create baseline flag variable, QSBLFL;

\* Many to One join. If study day variable, QSDY, matches the baseline flagged study day variable, BFLAGDY, then set the new baseline flag variable, QSBLFL, to ‘Y’. All other non-matching records are set to missing;

proc sql;

create table adhd2 as

select a.\*,

case

when qsdy = b.bflagdy then 'Y'

else ''

end as **qsblfl**

from adhdt as a left join bflag as b

on a.usubjid=b.usubjid and a.qscat=b.qscat and a.qsscat=b.qsscat and a.qstestcd=b.qstestcd;

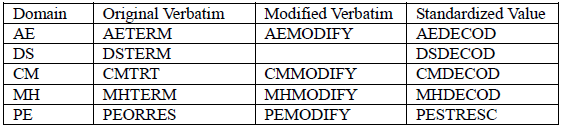
quit;

**Option # 3**

if visit = 'Baseline' then lbblfl = 'Y';

else lbblfl = ' ';

**XXTERM/XXDECOD – Original Verbatim/Standard Value**



**Common SUPPXX SDTM Variables**

**Common key variables: RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM**

**STUDYID/USUBJID/RDOMAIN – PKV016/XXXXXX-XXXX-XXXX/QS**

**IDVARVAL = XXSEQ, used to link with XXSEQ in XX dataset**

**IDVAR = ‘XXSEQ’, name of variable to link with XXSEQ variable [$IDVAR.]**

**QNAM = Coded name [$LBQNAM.]**

**QLABEL = Formatted Label [$LBQLABEL.]**

**QVAL = Original Value**

**QORIG = Source of data (CRF, Derived, Assigned) [$QORIG.]**

**QEVAL = Subjective Results [$QEEVAL.]**

data suppex;

length idvar $8 idvarval $3 qnam $8 qlabel $40 qval $200 qorig $10 qeval $30;

set suppex;

idvar='EXSEQ';

idvarval=strip(put(EXSEQ, best6.));

qorig='CRF';

qeval='INVESTIGATOR';

if exdoswt ne . then do;

qnam='EXDOSWT';

qlabel='Dosing weight (kg)';

qval=strip(put(exdoswt, best12.));

output;

end;

run;

