

```

/*=====*/
/*
Programmer:   R. VALABOJU
Date:        03-31-20xx
Purpose:      Macro to check uniqueness across library/studies
Requirements:
*/
/*=====*/

%macro chkstudyid;

proc sql NOPRINT;
    select distinct studyid into :study from sdtm.dm;
quit;
%let study=&study;

PROC DATASETS;
    Contents data=sdtm._all_ out=test NOPRINT;
RUN; QUIT;

proc sql noprint;
    select distinct memname into :memn1-:memn99 from test
    where name='DOMAIN';
quit;
%let totmem=&sqllobs;

%do i=1 %to &totmem;
proc freq data=sdtm.&&memn&i NOPRINT;
    tables studyid*domain / list out=&&memn&i (drop=percent);
    where studyid^="&study";
run;
%end;

proc sql noprint;
    select distinct memname into :memx1-:memx99 from test
    where name='RDOMAIN';
quit;
%let totmemx=&sqllobs;

%do i=1 %to &totmemx;
proc freq data=sdtm.&&memx&i NOPRINT;
    tables studyid*rdomain / list out=&&memx&i (rename=(RDOMAIN=DOMAIN) drop=percent);
    where studyid^="&study";
run;
%end;

```

```

proc sql noprint;
    create table out1 (
        issue char(100),
        studyid char(10) FORMAT $10.,
        domain char(10) FORMAT $10.,
        count num(8));
quit;

proc sql noprint;
    select distinct memname into :mem1-:mem999 from sashelp.vcolumn
    where libname='WORK'
    and memname not in ('DM', 'OUT1', 'TEST', 'DRALL1');
quit;
%LET TOTWRK=&SQLOBS;

%DO K=1 %TO &TOTWRK;

proc sql noprint;
    select distinct studyid into :id from &&mem&k;
quit;

%let id=&id;

data &&mem&k;
    length ISSUE $100;
    LENGTH STUDYID $10;
    LENGTH DOMAIN $10;
set &&mem&k;
    ISSUE="StudyID &id does not match value &study in DM domain";
run;

proc append base=out1
    data=&&mem&k force;
run; quit;

%END;

proc datasets nolist library=work memtype=data;
    save out1 DRALL1;
run; quit;

%mend chkstudyid;

%chkstudyid;

```

```

/*=====*/

/* TEST #2-1
1. source=CDISC
2. crtfile
3. codevalue
4. ctlist      single control term - use lower or uppercase
5. ctexclude   single control term - use lower or uppercase
*/

%chk_control_terms(
source=CDISC,
ctfile=SDTM_controlled_terms_20xx,
codevalue=CDISC_Submission_Value,
ctlist=lbtest,          /* Macro parameters without single quotes */
ctexclude=vstestcd);    /* Macro parameters without single quotes */

/*=====*/

%macro chkcts;
option compress=yes;

PROC IMPORT OUT=sdtmtrms
    DATAFILE="P:\Medical\SRSProd\utilities\standards\SDTM_controlled_terms_20xx.xls"
    DBMS=EXCEL2000 REPLACE;
    SHEET="CDISC_controlled_terminology";
    GETNAMES=YES;

RUN;
options symbolgen;
proc sql noprint;
    select distinct memname into :mem1-:mem999 from sashelp.vcolumn
    where libname='SDTM';
%LET TOTMEM=&SQLOBS;          /* total # of datasets in SDTM folder */

%DO J=1 %TO &TOTMEM;          /* outer loop - Process datasets */

    select name into :var1-:var999 from sashelp.vcolumn
    where libname='SDTM'
    and memname="&MEM&J"
    and name in (select distinct varname from sdtmtrms);
%LET TOTVAR=&SQLOBS;

%DO i=1 %TO &totvar;          /* inner loop - Process variables */

```

```

/* create datasets for each variable with controlled term
values */
create table &mem&j&&var&i (COMPRESS=YES) as
  select "&&var&i" as varname length=15,
    &&var&i as value label="value" length=200,
    case
      when "&&var&i" in
        (select CDISC_Submission_Value from sdtmtrms
         where varname in ('DOMAIN','RDOMAIN')
         and Codelist_Extensible='Yes') then 'Exact'
      when &&var&i in
        (select CDISC_Submission_Value from sdtmtrms
         where varname="&&var&i") then 'Exact'
      when upcase(&&var&i) in
        (select upcase(CDISC_Submission_Value) from sdtmtrms
         where varname="&&var&i") then 'Case'
      when "&&var&i" in
        (select varname from sdtmtrms
         where varname="&&var&i" and Codelist_Extensible='Yes') then 'Extensible'
      when &&var&i not in
        (select CDISC_Submission_Value from sdtmtrms
         where varname="&&var&i" and Codelist_Extensible='No') then 'None'
      else '' end as checkct length=10, count(*) as frequency
  from SDTM.&MEM&J;

/* assign flag value for each variable */

%END;
/* end of inner loop */
%END;
/* end of outer loop */

create table allcts
  (varname char(8) format $8.,
   value char(200) format $200.,
   checkct char(10),
   frequency num(8)
  );

/* create table to append all datasets */

select distinct memname into :mem1-:mem999 from sashelp.vcolumn
where libname='WORK'
and memname not in ('ALLCTS','SDTMTRMS','CT_EXTENSIBLE','CT_NOMATCH');

%LET TOTWRK=&SQLOBS;

%DO K=1 %TO &TOTWRK;

alter table &mem&k
modify value char(200) format $200.;

```

```

insert into allcts (varname, value, checkct, frequency)
select varname, value, checkct, frequency from &mem&k;
drop table &mem&k;

/* append all datasets */

%END;

/* create summary report of all variables and controlled
terms */
create table summary_report_&study as select distinct varname, value, checkct, count(*) as frequency from allcts
group by 1,2,3
order by 1,2,3,4;

drop table allcts;

/* additional reports */
create table ct_extensible_&study as select * from summary_report_&study
where checkct='Extensible'
order by 1;

create table ct_nomatch_&study as select * from summary_report_&study
where checkct='None'
order by 1;
quit;

%mend chkcts;

%chkcts; /* macro execution */

```

```

/*=====*/
/*
Programmer:  R. VALABOJU
Date:        03-31-20xx
Purpose:     Check duplicates across SDTM database according to established sort keys
Requirements: Standards excel spreadsheet
*/
/*=====*/
**Specify study name;
%let study=poa1001;

options nodate ps=45 ls=132 nocenter missing='.' nofmterr maautosource noserror nolabel
        NOSYMBOLGEN MPRINT MLOGIC FORMCHAR = ' |----|+|---+=|-/<>'
        sasautos=('P:\Medical\SRSPProd\maclib\test',
                  'P:\Medical\SRSPProd\maclib\prod',
                  'P:\Medical\dbadmin\macros',
                  'P:\Medical\dbadmin\tools',
                  );
%let common=P:\Medical\SRSPProd\utilities\standards\;
%let raw=P:\Medical\SRSPProd\test\standards\output\;
%let macro=P:\Medical\SRSPProd\mac_bsp\test\MACROS;
%let sdtm=P:\Medical\SRSPProd\test\standards\testdata;
%let sdtmwork=P:\Medical\SRSPProd\poa\poa1001\core\forbsp\sdtmwork;

libname sdtmwork 'P:\Medical\SRSPProd\poa\poa1001\core\forbsp\sdtmwork\';
libname sdtm 'P:\Medical\SRSPProd\test\standards\testdata';
run;

/* POA Studies*/
libname poa1001 'P:\Medical\SRSPProd\poa\poa1001\core\forbsp';
libname poa1004 'P:\Medical\SRSPProd\poa\poa1004\core\forbsp';
run;

/*Create test data*/

%macro didups_v1;
option compress=yes symbolgen;

PROC IMPORT OUT=dmnsvars
        DATAFILE="P:\Medical\SRSPProd\utilities\standards\SDTM_312_domains_and_varnames.xls"
        DBMS=EXCEL2000 REPLACE;
        SHEET="domains";
        GETNAMES=YES;

RUN;

```

```

DATA dmnsvars;
set dmnsvars;
  DO I = 1 TO 99 until (length=0);
    CALL SCANQ(dup_reckey,I,POSITION,LENGTH);
  END;
  NUM_WORDS = I-1;
  DROP POSITION LENGTH I;
run;

proc sql;
create table duprecs
  ( domain char(10),
    sortkeys char(200),
    count num(8)
  );
quit;

proc sql noprint;
select distinct upcase(domain) into :dmn1-:dmn99 from dmnsvars
where domain in
(select distinct memname from sashelp.vcolumn where libname='SDTM');
quit;
%let tot=&sqllobs;

%do i=1 %to &tot;
proc sql noprint;
select dup_reckey into :key from dmnsvars
where domain="&&dmn&i";

select num_words into :nkeys from dmnsvars
where domain="&&dmn&i";
quit;

%let nkeys=&nkeys;

data all&&dmn&i;
length key $200;
length domain $10;
length keynum 3;
key=' ';
domain=' ';
keynum=.;
run;

%do j=1 %to &nkeys;

```

```

%let k&j=%qscan(&key,&j,' ');

data &&k&j;
length key $200;
length domain $10;
length keynum 3;
  dsid=open("sdm.&&dmn&i");
  check=varnum(dsid,"&&k&j");
  if check=0 then key=" ";
  if check^=0 then key="&&k&j";
  domain="&&dmn&i";
  keynum="&j";
  drop dsid check;
run;

proc append base=all&&dmn&i data=&&k&j;
run;

proc sort data=all&&dmn&i;
by keynum;
run;

proc sql noprint;
drop table &&k&j;
quit;

%end;

%let keys=;

proc sql noprint;
select distinct key into :keys separated by ' ' from all&&dmn&i
where key^=' ';
quit;
%put &keys;

proc sql noprint;
select distinct key into :keys2 separated by ',' from all&&dmn&i
where key^=' ';
quit;
%put &keys2;

%if %length(&keys)>0 %then %do;
proc sort data=SDTM.&&dmn&i out=nodup&&dmn&i nodupkey;

```



```

by &keys;
run;

proc sql noprint;
create table dup&&dmn&i as
select distinct &keys2, count(*) as count from SDTM.&&dmn&i
group by &keys2
having count(*)>1
order by &keys2;
quit;

*drop table nodup&&dmn&i;
quit;
%end;

%if %length(&keys)>0 %then %do;
proc sql noprint;
insert into duprecs (domain, sortkeys, count)
select distinct "&&dmn&i" as domain, "&keys" as sortkeys,
count(count) as count from dup&&dmn&i;

*drop table dup&&dmn&i, all&&dmn&i;
quit;
%end;

%end;
/*
proc datasets library=work nolist;
save duprecs;
run;quit;
*/
%mend didups_v1;

%didups_v1;

```

```

/*=====*/
/*
Programmer:      R. VALABOJU
Date:           04-06-20xx
Purpose:        Create data view of AE, EX, DS data for subjects with AE
Requirements:
*/
/*=====*/

PROC PRINTTO LOG='P:\Medical\SRSPProd\test\standards\output\T_DVAEEXDS1.log' ;
RUN;

PROC PRINTTO PRINT='P:\Medical\SRSPProd\test\standards\output\T_DVAEEXDS1.lst';
RUN;

/*
proc printto log=log;
run;

proc printto print=print;
run;
*/

%macro dvaeexds_v1;

%let ds1=AE;
%let ds2=EX;
%let ds3=DS;

%do i=1 %to 3;

proc sql noprint;
create table &ds&i as select * from sdtm.&ds&i
/*where usubjid in (select usubjid from sdtm.AE)*/;
quit;

proc sort data=&ds&i;
by usubjid &ds&i..seq ;
run;

data &ds&i;
set &ds&i;
count + 1;

```

```

    by usubjid;
    if first.usubjid then count = 1;
run;
%end;

data view_AEEXDS;
merge ae (rename=(domain=domain1))
      ex (rename=(domain=domain2))
      ds (rename=(domain=domain3));
by usubjid count;
drop count;
run;

proc datasets library=work nolist;
save view_AEEXDS;
run; quit;

%mend dvaeexds_v1;

%dvaeexds_v1;

/* QA queries */
/*Count of distinct usubjid in program output dataset*/
proc sql;
title "Count of distinct subjects in program dvaeexds1 output";
select count(distinct usubjid) as count from dvaeexds1;
quit;

/*Count of all usubjid values in program output dataset*/
proc sql;
title "Count of all subjects in program dvaeexds1 output";
select count(usubjid) as count from dvaeexds1;
quit;

/*Count of distinct usubjid in program dvaeexds_v1 output */
proc sql;
title "Count of distinct subjects in program view_aeexds output";
select count(distinct usubjid) as count from view_aeexds;
quit;

/*Count of all usubjid values in program dvaeexds_v1 output */
proc sql;
title "Count of all subjects in program view_aeexds output";
select count(usubjid) as count from view_aeexds;
quit;

```

```
/* Cross check usubjid not matching between output datasets */
proc sql number;
title "Cross check usubjid not matching between output datasets";
select distinct usubjid from dvaeexds1
where usubjid not in (select usubjid from view_aeexds);
quit;

/* Cross check usubjid not matching between output datasets */
proc sql number;
title "Cross check usubjid not matching between output datasets";
select distinct usubjid from view_aeexds
where usubjid not in (select usubjid from dvaeexds1);
quit;

/* Proc compare of output from two programs */
title "Proc Compare of output datasets from both programs";
proc compare base=dvaeexds1
              compare=view_aeexds brief;
run;
```

```

/*=====*/
/*
Programmer:      R. VALABOJU
Date:            04-06-20xx
Purpose:         Create data view of DS, EX, XP data.
Requirements:
*/
/*=====*/

%macro dvdsexp_v1;

%let ds1=DM;
%let ds2=DS;
%let ds3=EX;
%let ds4=XP;

%do i=1 %to 4;

proc sql noprint;
create table &ds&i as select * from sdtm.&ds&i
order by usubjid ;
quit;

%end;

%do i=2 %to 4;
proc sort data=&ds&i;
by usubjid &ds&i..seq ;
run;
%end;

%do i=1 %to 4;

data &ds&i;
set &ds&i;
count + 1;
by usubjid;
if first.usubjid then count = 1;
run;

%end;

data view_DSEXXP;
merge dm (keep=USUBJID ARM ARMCD RFSTDTC RFENDTC COUNT )

```

```

        ds (keep=USUBJID DSTERM DSDECOD  EPOCH  VISIT VISITNUM COUNT rename=(VISIT=VISITDS VISITNUM=VISITNUMDS
EPOCH=EPOCHDS))
        ex (keep=USUBJID EXDOSE EXSTDTC EXSTDY VISITNUM VISIT COUNT )
        xp (keep=USUBJID XPTEST XPTESTCD XPCAT XPSCAT XPSTRESN VISIT VISITNUM  XPDTC COUNT
            rename=(VISIT=VISITXP VISITNUM=VISITNUMXP));
by usubjid count;
rename count=seq ;
run;

proc datasets library=work nolist;
save view_DSEXXP;
run; quit;

%mend dvdsexxp_v1;
%dvdsexxp_v1;

%dvdsexxp1;

```

```

/*=====*/
/*
Programmer:      R. VALABOJU
Date:            04-29-20xx
Purpose:         Create data view of DS, EX, XP data.
Requirements:
*/
/*=====*/

PROC PRINTTO LOG='P:\Medical\SRSProd\test\standards\output\T_DVDSEXXP1.log' ;
RUN;

PROC PRINTTO PRINT='P:\Medical\SRSProd\test\standards\output\T_DVDSEXXP1.lst';
RUN;

/*
proc printto log=log;
run;

proc printto print=print;
run;
*/

*****
Program Name:
autoexec.sas
*****;

options mprint mlogic symbolgen;

%macro v_DSEXXP1(xptestcd);
option compress=yes;

%macro exists;
%global exists;
DATA _null_;
if 0 then set SDTM.XP;
stop;
run;
%if &SYSERR=0 %then %let exists=YES;
%ELSE %LET exists=NO;
%mend exists;
%exists;

```

```

%IF &exists=YES %THEN %DO;

%let ds1=DM;
%let ds2=DS;
%let ds3=EX;
%let ds4=XP;

    %do i=1 %to 4;
    proc sql noprint;
    create table &&ds&i as select * from sdtm.&&ds&i
    order by usubjid ;
    quit;
    %end;

%if &xptestcd= %then %do;
%let xptestcd="P10AVG14","P10AVG24";
%let xptestcd2=P10AVG14,P10AVG24;
%end;
%else %if %upcase(&xptestcd)=P10AVG14 P10AVG24 %then %do;
%let xptestcd="P10AVG14","P10AVG24";
%let xptestcd2=P10AVG14,P10AVG24;
%end;
%else %if %upcase(&xptestcd)=P10AVG24 P10AVG14 %then %do;
%let xptestcd="P10AVG14","P10AVG24";
%let xptestcd2=P10AVG14,P10AVG24;
%end;
%else %if %upcase(&xptestcd)=P10AVG14 %then %do;
%let xptestcd="P10AVG14";
%let xptestcd2=P10AVG14;
%end;
%else %if %upcase(&xptestcd)=P10AVG24 %then %do;
%let xptestcd="P10AVG24";
%let xptestcd2=P10AVG24;
%end;

proc sql;
create table v_temp1 as
select * from ds(keep=USUBJID DOMAIN DSSEQ DSTERM DSDECOD DSDTC DSSTDTC EPOCH VISIT VISITNUM
                rename=(VISIT=VISITDS VISITNUM=VISITNUMDS EPOCH=EPOCHDS DSSEQ=SEQ))
outer union corr
select * from ex(keep=USUBJID DOMAIN EXSEQ EXDOSE EXSTDTC EXENDTC EXSTDY EXENDY VISITNUM VISIT EPOCH
                rename=(VISIT=VISITEX VISITNUM=VISITNUMEX EPOCH=EPOCHEX EXSEQ=SEQ))

```



```

outer union corr
select * from xp(
  where=(xptestcd in (&xptestcd))
  keep=USUBJID DOMAIN XPSEQ XPTST XPTSTCD XPCAT XPSCAT XPSTRESN VISIT VISITNUM XPTC EPOCH
  rename=(VISIT=VISITXP VISITNUM=VISITNUMXP EPOCH=EPOCHXP XPSEQ=SEQ));

quit;

proc sql nowarn;
create table v_temp2 as select b.*,
case when b.domain="DS" then coalesce(b.seq) else . end as DSSEQ,
case when b.domain="EX" then coalesce(b.seq) else . end as EXSEQ,
case when b.domain="XP" then coalesce(b.seq) else . end as XPSEQ,
case when b.xptestcd="P10AVG14" then coalesce(xpstresn) else . end as P10AVG14,
case when b.xptestcd="P10AVG24" then coalesce(xpstresn) else . end as P10AVG24
from v_temp1 as b
order by usubjid, DOMAIN, SEQ;
quit;

proc sql nowarn;
create table v_dsexxp1 as
select usubjid,dsseq,epochds,dsterm,dsdecod,visitds,visitnumds,dsdtc,dsstdtc,exseq,epochex,visitex,
visitnumex,exdose,exstdtc,exstdy,exendtc,exendy,xpseq,epochxp,visitxp,visitnumxp,xptdc,xptestcd,
xptest,xpcat,xpstat,&xptestcd2 from v_temp2
order by usubjid,domain,seq;
quit;

%END;
%ELSE %DO;
DATA _NULL_;
FILE PRINT;
PUT "THE DATASET XP DOES NOT EXIST IN THIS STUDY";
%END;
RUN;

%mend v_DSEXXP1;

%v_DSEXXP1;
%dvdsexxp1;

proc compare base=dvdsexxp1
              compare=v_DSEXXP1 brief;

run;

```

```

/*=====*/
/*
programmer:      r. valaboju
date:           04-29-20xx
purpose:        create data view of ds, ex, xp ae data where d/c reason is LOTE (lack of Tx Efficacy).
requirements:
*/
/*=====*/
/*
proc printto log='p:\medical\srsprod\test\standards\output\t_dvdsexxpael.log' ;
run;

proc printto print='p:\medical\srsprod\test\standards\output\t_dvdsexxpael.lst';
run;

/*
proc printto log=log;
run;

proc printto print=print;
run;
*/

options mprint mlogic symbolgen;

%macro v_dsexxpael(xptestcd=p10avg14 p10avg24);
option compress=yes;

%macro exists;
%global exists;
data _null_;
if 0 then set sdtm.xp;
stop;
run;
%if &syserr=0 %then %let exists=yes;
%else %let exists=no;
%mend exists;
%exists;

%if &exists=yes %then %do;

%let ds1=dm;
%let ds2=ds;
%let ds3=ex;

```

```

%let ds4=xp;
%let ds5=ae;

    %do i=1 %to 5;
    proc sql noprint;
    create table &&ds&i as select * from sdtm.&&ds&i
    order by usubjid ;
    quit;
    %end;

%if %upcase(&xptestcd)=%upcase(p10avg14 p10avg24) %then %do;
%let xptestcd=%upcase("p10avg14","p10avg24");
%let xptestcd2=p10avg14,p10avg24;
%end;
%else %if %upcase(&xptestcd)=%upcase(p10avg24 p10avg14) %then %do;
%let xptestcd=%upcase("p10avg14","p10avg24");
%let xptestcd2=p10avg14,p10avg24;
%end;
%else %if %upcase(&xptestcd)=%upcase(p10avg14) %then %do;
%let xptestcd=%upcase("p10avg14");
%let xptestcd2=p10avg14;
%end;
%else %if %upcase(&xptestcd)=%upcase(p10avg24) %then %do;
%let xptestcd=%upcase("p10avg24");
%let xptestcd2=p10avg24;
%end;

proc sql;
create table v_temp as
select * from ds(rename=(visit=visitds visitnum=visitnumds epoch=epochds dsseq=seq))
where usubjid in (select usubjid from ds where dsterm="lack of therapeutic effect")
outer union corr
select * from ex(rename=(visit=visitex visitnum=visitnumex epoch=epochex exseq=seq))
where usubjid in (select usubjid from ds where dsterm="lack of therapeutic effect")
outer union corr
select * from xp(rename=(visit=visitxp visitnum=visitnumxp epoch=epochxp xpseq=seq))
where xptestcd in (&xptestcd)
and usubjid in (select usubjid from ds where dsterm="lack of therapeutic effect")
outer union corr
select * from ae (drop=studyid rename=(aeseq=seq))
where usubjid in (select usubjid from ds where dsterm="lack of therapeutic effect");
quit;

proc sql nowarn;
create table v_temp1 as select b.*,

```

```

case when b.domain="ds" then coalesce(b.seq) else . end as dsseq,
case when b.domain="ex" then coalesce(b.seq) else . end as exseq,
case when b.domain="xp" then coalesce(b.seq) else . end as xpseq,
case when b.domain="ae" then coalesce(b.seq) else . end as aeseq,
case when b.xptestcd="p10avg14" then coalesce(xpstresn) else . end as p10avg14,
case when b.xptestcd="p10avg24" then coalesce(xpstresn) else . end as p10avg24,
case when b.domain="ds" then 1
      when b.domain="ex" then 2
      when b.domain="xp" then 3
      when b.domain="ae" then 4
      else . end as domain_n
from v_temp as b
order by usubjid, domain_n, seq;
quit;

```

```

proc sql nowarn;
create table v_dsexxpael as
select
usubjid,dsseq,epochds,dsterm,dsdecod,visitds,visitnumds,dsdtc,dsstdtc,exseq,epochex,visitex,visitnumex,exdose,exstdtc,e
xstdy,exendtc,exendy,xpseq,epochxp,visitxp,visitnumxp,xpdtc,xptestcd,xptest,xpcat,xpscat,&xptestcd2,aeseq,aespid,aeterm
,aemodify,aedecod,aebodsys,aesev,aeser,aeacn,aeacnoth,aerel,aeout,aescong,aesdisab,aesdth,aeshosp,aeslife,aesmie,aecont
rt,aestdtc,aeendtc,aestdy,aeendy,aestrf,aeenrf
from v_temp1
order by usubjid, domain_n, seq;
quit;

```

```

/*
proc datasets library=work nolist;
save v_dvdsexxpael;
run; quit;
*/

```

```

%end;
%else %do;
data _null_;
file print;
put "dataset xp does not exist in this study";
%end;
run;

```

```

%mend v_dsexxpael;

```

```

/* compare 1 */
%v_dsexxpael(xptestcd=p10avg14 p10avg24);

```

```
%dvdsexxpael(xptestcd=p10avg14 p10avg24);  
  
title "test #1 proc compare output";  
proc compare base=dvdsexxpael  
              compare=v_dsexxpael brief;  
run;  
title;
```

```

/*=====*/
/*
Programmer:      R. VALABOJU
Date:           05-03-20xx
Purpose:         Generic program to Create view of Datasets with xxTESTCD, xxSTRESN, variables. Shows difference
                  from prior visit values as well as change from baseline values.

Requirements:
*/
/*=====*/

```

Purpose: Checks the test results against range and difference values input and displays subject results meeting the input criteria
 For subjects identified as having at least one outlier, displays all results and indicates outliers where applicable.
 An outlier includes a baseline to endpoint check which is always displayed

sample macro call

```
%dvalldiff1(dsname=VS, vstestcd=DIABP, diffamt=20, highflag=140, lowflag=100, unit=mmHG)
```

displays all DIABP results for subjects that have at least one occurrence of

flags any difference >= 20 from the previous collection

flags low under 100

flags upper over 140

displays the baseline to endpoint result if a baseline record is flagged 'Y' in --BLFL

displays the baseline record flag

```
%macro v_dvalldiff1(dsname=, testcd=, diffamt=, highflag=, lowflag=, unit=);
```

```
data &dsname;
```

```
set sdtm.&dsname (keep=usubjid &dsname.seq &dsname.testcd &dsname.stresn &dsname.stresu &dsname.blfl &dsname.dtc);
count+1;
```

```
by usubjid;
```

```
where upcase(&dsname.testcd)=upcase("&testcd");
```

```
where also upcase(&dsname.stresu)=upcase("&unit");
```

```
if first.usubjid then count=1;
```

```
run;
```

```
proc sql;
```

```
create table bl as select usubjid, &dsname.stresn, &dsname.blfl from &dsname
```

```
where &dsname.blfl="Y";
```

```
create table &dsname.bl as select a.*, b.&dsname.stresn as blstresn from &dsname as a left join bl as b
```

```
on a.usubjid=b.usubjid
```

```
order by a.usubjid;
```

```
quit;
```

```

proc sort data=&dsname.bl;
by usubjid &dsname.seq count;
run;

data &dsname;
set &dsname.bl;
by usubjid;
if &dsname.blfl^="Y" and &dsname.stresn=. then delete;
run;
proc sort data=&dsname;
by usubjid &dsname.seq count;
run;

data &dsname;
set &dsname;
by usubjid;
if &dsname.stresn<input("&lowflag",8.) then result_hilow=&dsname.stresn-input("&lowflag",8.);
if &dsname.stresn>input("&highflag",8.) then result_hilow=&dsname.stresn-input("&highflag",8.);
if not first.usubjid and abs(dif(&dsname.stresn))>=input("&diffamt",8.) then diffflag=1;
result_change=dif(&dsname.stresn);
run;
proc sort data=&dsname;
by usubjid &dsname.seq count;
run;

data &dsname.out;
set &dsname;
by usubjid;
if diffflag=. then result_change=.;
if last.usubjid and &dsname.stresn^=. then bl_to_endpt=&dsname.stresn-blstresn;
drop diffflag blstresn;
run;

proc sql;
create table &dsname.tmp as select distinct usubjid, result_change, result_hilow, bl_to_endpt from &dsname.out
where (result_change^=. or result_hilow^=. or bl_to_endpt^=.);

delete from &dsname.tmp
where (result_change=. and result_hilow=.)
and abs(bl_to_endpt)<abs(input("&diffamt",8.));

delete from &dsname.out
where usubjid not in (select usubjid from &dsname.tmp);
quit;
%mend v_dvalldiff1;

```

```

/*****
Sample table program -
*****/

libname r "P:\Medical\SRSPProd\otr\otr1021\core\forbsp";
run;

/* Copy EX dataset to work folder and assign single letter treatment codes */
data ex;
length treatc $1;
set r.ex(keep=studyid usubjid exseq exdostxt exstdtc exendtc epoch visitnum);
if EXDOSTXT="FINELY CRUSHED 10 MG OTR" then treatc="A";
if EXDOSTXT="COARSELY CRUSHED 10 MG OTR" then treatc="B";
if EXDOSTXT="FINELY CRUSHED 10 MG OC" then treatc="C";
drop exdostxt;
run;

/* Population counts by treatment assigned to macro variables */
proc sql noprint;
select count(distinct usubjid) into :trta from ex where treatc="A";
select count(distinct usubjid) into :trtb from ex where treatc="B";
select count(distinct usubjid) into :trtc from ex where treatc="C";
select count(distinct usubjid) into :overall from ex where treatc in ("A","B","C");
quit;

/* Re-assign Population count macro variables to remove blanks */
%let trta=&trta;
%let trtb=&trtb;
%let trtc=&trtc;
%let overall=&overall;

/* Transpose ex dataset to get start and end dates of exposure by treatment */
proc transpose data=ex out=trex(drop=_name_);
by usubjid;
id treatc ;
var exstdtc;
run;

/* Transpose ex dataset to get start and end dates of exposure by treatment */
proc transpose data=ex out=trex2(drop=_name_ rename=(a=paendt b=pbendt c=pcendt));
by usubjid;
id treatc ;
var exendtc;
run;

```



```

/* Add :00 to the ae start and end dates to make use of the is8601 function later on the variables */
data ae;
set r.ae;
aestdtc2=trim(trim(aestdtc)||trim(":00"));
aeendtc2=trim(trim(aeendtc)||trim(":00"));
drop aestdtc aeendtc;
rename aestdtc2=aestdtc aeendtc2=aeendtc;
run;

/* Merge ae and transposed ex datasets to get start and stop dates of exposure for all treatments for every subject ae
record */
data aeexst;
merge ae trex trex2;
by usubjid;
run;

/* Create numeric dates using is8601 function and datetime20 format for later use */
data aeexst;
set aeexst;
array dates(8) aestdtc aeendtc a b c paendt pbendt pcendt;
array numdates(8) _aestdtc _aeendtc _a _b _c a _b _c _;
do i = 1 to 8;
if dates(i) ^= '' then
numdates(i) = input(dates(i), IS8601DT.);
end;
drop i;
format _aestdtc _aeendtc _a _b _c a _b _c _ datetime20.;
run;

/* Create numeric dates using is8601 function and datetime20 format for later use */
data dm;
set r.dm;
if rfstdtc ^= '' then do;
rfstdt=input(rfstdtc, IS8601DT.);
rfendtc=input(rfendtc, IS8601DT.);
end;
format rfstdt rfendtc datetime20.;
run;

/* Merge dm data with rfstdt and rfendtc variables with merged ae and ex dataset from above */
data aed;
merge aeexst dm(keep=usubjid rfstdt rfendtc);
by usubjid;
run;

```

```

/* Delete non ae records from the merged dataset */
proc sql;
delete from aed
where usubjid not in (select usubjid from r.ae);
quit;

/* Assign treatment groups to each AE record depending on when AE occurred - used ae start dates in calculation */
proc sql;
create table aed2 as select *,
case when _aestdtc<=rfstdt then "Z"
when (_aestdtc between _a and _a+48*60*60) or (_aestdtc between _a and _a+48*60*60) then "A"
when (_aestdtc between _b and _b+48*60*60) or (_aestdtc between _b and _b+48*60*60) then "B"
when (_aestdtc between _c and _c+48*60*60) or (_aestdtc between _c and _c+48*60*60) then "C"
else "" end as treatc
from aed;
quit;

options mprint mlogic symbolgen;

/* Macro to prepare final report dataset by creating horizontal and vertical datasets and merging */
%macro aeprep;

%let trt1=%upcase(a);
%let trt2=%upcase(b);
%let trt3=%upcase(c);
%let trt4=%upcase(z);
%let trt5=overall;

%let str=%nrstr(put(count(*),3.0) ||compbl(" (N="||put(count(distinct usubjid),3.0)||")"));
/* Vertical dataset "overall" to get overall numbers - no grouping */
proc sql ;
create table &trt5 as
(select "Any Adverse Event" as AE_GRP, %unquote(&str) as Overall, "1" as _order from aed2)
union all
(select distinct(propcase(aesev)) as AE_GRP, %unquote(&str) as Overall, "2" as _order from aed2
group by 1)
union all
(select "Possible, Probably, or Definitely Related Adverse Event" as AE_GRP,
%unquote(&str) as Overall, "3" as _order from aed2
where aerel in ("POSSIBLY","PROBABLY","DEFINITELY"))
union all
(select "Adverse Event Leading to Study Discontinuation" as AE_GRP,

```

```

        %unquote(&str) as Overall, "4" as _order from aed2
        where aeacnoth in ("WITHDRAWN FROM STUDY","TREATMENT GIVEN AND WITHDRAWN FROM STUDY"))
union all
        (select "Serious Adverse Event" as AE_GRP, %unquote(&str) as Overall, "5" as _order from aed2
        where aeser="Y");
quit;

/* Vertical datasets "Pre-Study Drug", "A", "B", "C" - grouping by treatment and pre-study */
%do i=1 %to 4;
proc sql ;
        create table &&trt&i as
        (select "Any Adverse Event" as AE_GRP, %unquote(&str) as &&trt&i, "1" as _order from aed2
        where treatc="&&trt&i")
union all
        (select distinct(propcase(aesev)) as AE_GRP, %unquote(&str) as &&trt&i, "2" as _order from aed2
        where treatc="&&trt&i"
        group by 1)
union all
        (select "Possible, Probably, or Definitely Related Adverse Event" as AE_GRP,
        %unquote(&str) as &&trt&i, "3" as _order from aed2
        where treatc="&&trt&i"
        and aerel in ("POSSIBLY","PROBABLY","DEFINITELY"))
union all
        (select "Adverse Event Leading to Study Discontinuation" as AE_GRP,
        %unquote(&str) as &&trt&i, "4" as _order from aed2
        where treatc="&&trt&i"
        and aeacnoth in ("WITHDRAWN FROM STUDY","TREATMENT GIVEN AND WITHDRAWN FROM STUDY"))
union all
        (select "Serious Adverse Event" as AE_GRP, %unquote(&str) as &&trt&i, "5" as _order from aed2
        where treatc="&&trt&i"
        and aeser="Y");
quit;
%end;

/* Create row for dataset Z where proc sql did not create a row for SEVERE ae category */
data z;
set z;
if _n_=3 then do;
output;
ae_grp="SEVERE";
_order="2";
z="---";
end;
output;
run;

```

```

/* include ae group labels and order variable values */
%do i=1 %to 5;
data &&trt&i;
set &&trt&i;
if _n_=6 and ae_grp='' then do;
ae_grp="Adverse Event Leading to Study Discontinuation";
_order="4";
end;
if _n_=7 and ae_grp='' then do;
ae_grp="Serious Adverse Event";
_order="5";
end;
if _order="2" then do;
ae_grp=" "||ae_grp;
end;
run;
proc sort data = &&trt&i;
by _order;
run;
%end;

/* Create final report dataset by merging all vertical datasets created above */
data final;
merge z a b c overall;
by _order;
run;

proc datasets library=work;
save final;
run; quit;

%mend aeprp;

%aeprp; /* Macro execution step to create vertical datasets */

/* Create blank rows after each ae group category */
data final;
set final;
output;
by _order;
if _order^=1 and last._order;
array allchar [*] _character_ ;
drop i;
do i=1 to dim(allchar); allchar{i}=' ' ; end;

```

```

output; /* Output blank observation */
run;

/* Format values per SAP specifications */
proc format;
    value $col 30. ' 0 (N=0) '=' --- '
                  ' 0 (N= 0) '=' --- '
;
run;

/*Create macro variable for sysdate system macro variable in format specified in shells */
data _null_;
today = put(date(),date9.);
call symput('dtnull',today);
run;
%let dtnull2=%substr(&dtnull,1,2)-%substr(&dtnull,3,3)-%substr(&dtnull,6,4);

/* Report procedure to export rtf file */
dm "odsresults; clear;";
options nonumber nodate orientation=landscape;
ods listing close;
ods rtf file="\nacl03\users_ab\valabojr\Documents\OTR1021\reports\14-3-1-1a_new.rtf" bodytitle style=Styles.custom;
proc report
    data = final
    nowd
    ls=122
    ps=30
    headline
    headskip
    split='*'
    style(report)={font_face=arial font_size=2.5 bordercolor=black}
    style(column)={just=center font_face=arial font_size=2.5 bordercolor=black}
    style(header)={font_face=arial font_size=2.5 foreground=black bordercolor=black}
;

columns (ae_grp('Incidences (Number of Distinct Subjects)*'
               '
               z a b c overall _order));

define ae_grp /display " " style={protectspecialchars=off cellwidth=30% just=1 asis=on} flow;
define z /display center "Pre-Study*Drug" format=$col. style={protectspecialchars=off cellwidth=13% just=c};
define a /display center "A*(N=&trta)" format=$col. style={protectspecialchars=off cellwidth=13% just=c};
define b /display center "B*(N=&trtb)" format=$col. style={protectspecialchars=off cellwidth=13% just=c};
define c /display center "C*(N=&trtc)" format=$col. style={protectspecialchars=off cellwidth=13% just=c};

```

```

define overall /display center "Overall*(N=&overall)" format=$col. style={protectspecialchars=off cellwidth=17%
just=c};
define _order /order=internal noprint;

compute before;
line @1 ' ';
endcomp;

title1 j=1 "OTR1021" Table 14.3.1-1a
DATE:&dtnull12 TIME:&systime";
title2 j=c "Summary of All Adverse Events";
title3 j=c "(Page 1 of 1)";
*title3 j=c "Page ^{thispage} of ^{lastpage}";
title4;
title5 j=1 "Population: Enrolled";
footnote j=1 "NOTE: Treatment A = Finely crushed 10 mg OTR; Treatment B = Coarsely crushed 10 mg OTR; Treatment C
= Finely crushed 10 mg OC.";
footnote2 j=1 "Source: Listing 16.2.7-1a";
run;quit;
ods listing;
ods rtf close;
/*****/

```

```
/***** ADHOC PROGRAM WRITTEN TO CALCULATE MORPHINE EQUIVALENT DOSAGE OF CONMEDS*****/
```

```
PROGRAMMER: R. VALABOJU
```

```
DT: 04-20xx
```

```
PURPOSE: conmeds dataset structure had single rows for multiple days of dosage. additional records created between  
start and end dates and average dosage calculated per day. subjects identified that fall out of per day dose limits.
```

```
/*****/
```

```
libname temp 'P:\Medical\SRSProd\buprenorphine\submission_2009\bup3015\cor_data' ;  
libname temp2 'P:\Medical\SRSProd\buprenorphine\submission_2009\bup3015\ana_data';  
run;
```

```
options nolabel;
```

```
proc sql;  
create table cm as  
select distinct subjid, randpoc, cmseq, safetyc,opflag,trtgrp,pause,  
cmstest, cmenest, pastdt, paendt, cmdsmegn, cmdurpa, cmavpads,  
cmavtot,cmdecod2 from temp2.a_conmed  
where safety=1  
and opflag=1  
and pause='Yes'  
order by subjid,cmseq;  
quit;
```

```
%put &sqlobs;
```

```
data cm;  
set cm;  
if cmstest<pastdt then cmstest=pastdt;  
if cmstest>paendt then delete;  
if cmenest>paendt then cmenest=paendt;  
if cmenest<pastdt then delete;  
run;
```

```
proc sort data=cm;  
by subjid cmstest;  
run;
```

```
data cm2;  
set cm;  
by subjid cmstest;  
do extended=cmstest to cmenest;
```

```

if cmdurpa>1 then do;
cmstest=cmstest+1;
end;
output;
end;
run;

data cm2;
set cm2;
if cmdurpa>1 then do;
cmstest=cmstest-1;
end;
run;

proc sql;
create table cm3 as select *, abs(sum(cmdsmeqn)) as sumday from cm2
group by subjid, cmstest;
quit;

proc sql;
create table test as select * from cm3
where sumday=.
and cmdurpa>1;
quit;
/*
proc sql;
create table test2 as select * from cm2
where subjid in (select subjid from test);
quit;
*/
proc sql;
create table cm4 as
select distinct subjid, randpoc, trtgrp, cmstest,
pastdt, paendt, sumday from cm3 group by subjid;
quit;

proc means data=cm4;
class subjid;
var sumday;
run;

proc means data=cm4;
var sumday;
run;

```



```
data cm4;
length flag 3;
set cm4;
flag=0;
if sumday<30 or sumday>80 then flag=1;
run;
```

```
data cm5;
set cm4;
by subjid;
if cmstest<(paendt-6) then delete;
run;
```

```
proc sql;
create table cm6 as select *, count(*) as count from cm5
group by subjid;
quit;
```

```
proc sql;
create table cm7 as select distinct subjid, count(flag) as flagcount from cm5
where flag=1
group by subjid
order by subjid;
quit;
```

```
data cm7;
merge cm6 cm7;
by subjid;
if flagcount=. then flagcount=0;
run;
```

```
proc sql;
select * from cm7
where sumday=.;
quit;
```

```
proc sql;
create table meq_pd as select distinct subjid, randpoc, trtgrp, pastdt, paendt, count, sumday, flagcount,
case when count>=7 and flagcount>3 then 'Y'
      when count=6 and flagcount>3 then 'Y'
      when count=5 and flagcount>3 then 'Y'
      when count=4 and flagcount>2 then 'Y'
      when count=3 and flagcount>2 then 'Y'
      when count=2 and flagcount>0 then 'Y'
```

```
        when count=1 and flagcount>0 then 'Y'  
        else '' end as pdflag
```

```
from cm7  
order by subjid;  
quit;
```

```
proc sql;  
create table meq_pd_final as  
select * from meq_pd  
where subjid in (select distinct subjid from meq_pd  
where pdflag='Y');  
quit;
```

```
data MEQ_PDS;  
merge meq_pd_final(in=a) cm5(in=b);  
by subjid;  
if a;  
run;
```

```
proc sql;  
select count(distinct subjid) from cm7  
where sumday=.;  
quit;
```

```
/******end******/
```