```
libname adam_raw '/home/u58485303/adam_raw';
libname adam_out '/home/u58485303/adam_output';
libname a_base '/home/u58485303/adam_base';

options validvarname=upcase;

/* merged lb dataset with adsl */

data lb1;
length APHASE $50;
merge adam_raw.lb (in=a) adam_out.adsl (in=b);
by usubjid;
if a and b;

TRTP= TRT01P;
TRTPN= TRT01PN;
TRTA= TRT01A;
TRTAN= TRT01AN;

/* ADT ADTM ATM */
if LBDTC ne '' and length (LBDTC)> 10 then ADTM=
input(LBDTC,??is8601dt.);
ADT= input(scan(LBDTC,1,'T'),??is8601da.);
ATM= timepart(ADTM);

format ADT date11. ADTM datetime16. ATM time5.;

/* ADY */
if ADT>= TRTSDT then ADY= ADT-TRTSDT+1;
else ADY= ADT-TRTSDT;

/* AVISIT AVISITN */
AVISIT= propcase(VISIT);
AVISITN= VISITNUM;

/* PARAM PARAMCD PARCAT1 PARCAT2 AVAL AVALC */
if lbstresu ne '' then PARAM= strip(lbtest)||'
('||strip(lbstresu)||')';
else if lbstresu eq '' then PARAM= strip(lbtest);

PARAMCD= LBTESTCD;

PARCAT1= LBCAT;
PARCAT2= LBSCAT;

AVAL= LBSTRESN;
```

```
AVALC= LBSTRESC;

/* ANRIND ANRINDN ANRLO ANRHI */
ANRIND= LBNRIND;

if ANRIND= 'NORMAL' then ANRINDN=1;
else if ANRIND= 'ABNORMAL' then ANRINDN=2;

ANRLO= LBSTNRLO;
ANRHI= LBSTNRHI;

/* APHASE */
if adt>= trtsdt > . then APHASE= 'On-Treatment';
if . < adt < trtsdt  then APHASE= 'Pre-Treatment';
If ADT > TRTEDT > . then APHASE= 'Post-Treatment';

run;

/* ABLFL BASE BASEC */
proc sort data= lb1 out=pre_base;
by usubjid /* parcat1 */ paramcd adt adtm;
where (adt ne . and trtsdt ne . and aval ne .) and  (adt < trtsdt);
run;

data base;
set pre_base;
by usubjid /* parcat1 */ paramcd adt adtm;
if last.paramcd;
ABLFL= 'Y';
BASE= AVAL;
BASEC= AVALC;
keep USUBJID /* parcat1 */ PARAMCD LBSEQ ABLFL BASE BASEC;
run;

proc sort data= base;
by usubjid /* parcat1 */ paramcd lbseq;
run;

proc sort data= lb1;
by usubjid /* parcat1 */ paramcd lbseq;
run;

/* merged to get ablfl */
data lb2;
merge lb1 (in=a) base (in=b keep= usubjid /* parcat1 */ paramcd lbseq
ablfl);
```

```
by usubjid /* parcat1 */ paramcd lbseq;
if a;

run;

/* merged to get base basec */
data lb3;
merge lb2 (in=a) base (in=b keep= usubjid /* parcat1 */ paramcd base
basec);
by usubjid /* parcat1 */ paramcd;
if a;

if ADY>= 1 /* and ABLFL ne 'Y' */ then do;
chg= aval- base;
if chg ne 0 and base ne 0 then pchg= (chg/base)*100;
end;

run;

/* derived records of min aval for each paramcd and appended to master
dataset */
proc sort data= lb3 out=min;
by usubjid parcat1 paramcd aval adt adtm;
run;

data lb4;
set min;
by usubjid parcat1  paramcd aval adt adtm;
if first.paramcd;
AVISIT= 'Minimum';
AVISITN= 9999;
DTYPE= 'MIN';

call missing (lborres, lborresu, lbstresc, lbstresn, lbstresu, visit,
visitnum, lbdtc, epoch, ady, ablfl);
run;

data lb5;
set lb3 lb4;
run;

/* SHIFT1 SHIFT1N */
proc sort data=lb5;
by usubjid paramcd avisitn adt;
run;
```

```
data shift;
set lb5;
by usubjid paramcd;
retain BIND;
if ablfl= 'Y' then BIND= ANRIND;

if parcat1= 'URINALYSIS' and (ADY ge 1 or ADY=.) then do;
       if ANRIND= BIND then do; shift1= 'To No Change'; shift1n=2;
end;
       else if ANRIND ne BIND then do;
              if ANRIND= 'NORMAL' then do; shift1= 'To Normal';
shift1n=1; end;
              else if ANRIND= 'ABNORMAL' then do; shift1= 'To
Abnormal'; shift1n=3; end;
       end;
end;

/* anl01fl */
if find(visit,'unsch','i') then anl01fl='';
else anl01fl='Y';

/* anl02fl */
if ANRIND in ('ABNORMAL', 'NORMAL') then anl02fl='Y';
else anl02fl='';

run;

proc sort data= shift;
by usubjid paramcd avisitn avisit adt;
run;

%let keepvar= STUDYID USUBJID SUBJID SITEID TRTP TRTPN TRTA TRTAN
SCRNFL ENRLFL ITTFL SAFFL PKFL ADT ADTM ATM ADY
       VISIT VISITNUM AVISIT AVISITN PARAM PARAMCD PARCAT1 PARCAT2
AVAL AVALC DTYPE ABLFL BASE BASEC CHG ANRIND ANRINDN
       ANRLO ANRHI EPOCH LBDTC LBSEQ LBORRES LBORRESU LBSTRESN
LBSTRESU LBSTRESC LBSTAT LBNAM LBSPEC LBNRIND LBSTNRLO
       LBSTNRHI LBORNRLO LBORNRHI SHIFT1 SHIFT1N APHASE ANL01FL
ANL02FL;

data adam_out.adlb;
retain &keepvar;
set shift;

keep &keepvar;
run;
```

```
proc compare base=a_base.adlb comp=adam_out.adlb outcompare outbase
outnoequal outdiff method=exact out=difference;
/* ods listing file= '/home/u58485303/output/adlb.lst'; */
run;
```