

```
/*=====
=====*
```

```
| Covance Study Number   : 000000106331          |
| Program Name           : d_adeq.sas             |
| Purpose                 : Create ADEG dataset    |
| Input Data              : SDTM.EG SDTM.SUPPEG ADAM.ADSL          |
| Output Data             : ADAM.ADEG             |
| Macros Called           : m_printto, m_logchk, m_attrib_adam    |
| Originally Performed by : kpothuri              |
| Date                    : 16March2015           |
|                          |
```

```
|=====
=====|
```

```
| Modification History          |
|-----|
| Modified by                   : kpothuri          |
| Modification Date             : 5/26/15           |
| Modification Description      : ABLFL derivation, shift1 derivation
|                                     6/29/15 - remove leading zeros from basec
and avalc
```

```
+=====
=====*/
```

```
options validvarname=upcase;
```

```
libname adam "&base2/datasets/adam/cleaned_adam";
```

```
libname sdtm "/cvn/projects/prj/data/000000106331/datasets/sdtm/sdtmx";
```

```
%m_printto(route=YES);
```

```
*****,
```

```
* bring in ADSL ;
```

```
*****,
```

```
data adsl;
```

```
    set adam.adsl;
```

```
    drop studyid;
```

```
run;
```

```
proc sort data=adsl; by usubjid; run;
```

```
*****,
```

```
* pick up SUPPEG      ;
```

```
*****,
```

```
data suppeg;
```

```
length egclsig $3;
```

```
    set sdtm.suppeg;
```

```
    /*EGCLSIG*/
```

```
    if QNAM="EGCLSIG" then do;
```

```
        EGCLSIG=QVAL;
```

```
        egseq=input(idvarval,best.);
```

```
        output suppeg;
```

```
    end;
```

```

        keep usubjid idvarval egclsig egseq;

run;

*****.

* pick up EG      ;

*****.

data hr;

    set sdtm.eg(where=(egtestcd='HRMEAN') rename=(egstresn=hr));

run;

data qt;

    set sdtm.eg(where=(egtestcd='QTMEAN') rename=(egstresn=qt));

run;

proc sort data=hr (drop=STUDYID DOMAIN EGSEQ EGTESTCD EGTEST EGCAT EGPOS EGORRES
EGORRESU EGSTRESC EGSTRESU
EGSTAT EGREASND EGMETHOD EGBLFL VISIT VISITDY EPOCH EGDTC EGDY EGTPPT);

    by usubjid visitnum egtptnum;

run;

proc sort data=qt (drop=egmethod egcat egpos);

    by usubjid visitnum egtptnum;

run;

*****.

* calculate QTcF ;

```

```

*****.
data qtcf(drop=hr qt);
length egtest $200;
    merge hr qt;
    by usubjid visitnum egtptnum;
    egtestcd = 'QTcf';
    egtest = "QTcf - Fridericia's Correction Formula (msec)";
    egstresn = (qt/((60/hr)**(1/3)));
    egstresc = left(trim(put(egstresn,5.)));
    egorres = trim(egstresc);
    egseq=.;
run;

```

```

*****.

```

```

* Add to EG;

```

```

*****.

```

```

data ELENGTH;

```

```

    set SDTM.EG;

```

```

run;

```

```

data eg;

```

```

length egtest $200;

```

```

    set ELENGTH qtcf;

```

```

run;

```

```

proc sort data=eg; by usubjid egseq; run;

```

```
data eg1;

length avalc $200 param $50;

set eg;

/*Parameters*/

paramcd = trim(egtestcd);

if egtestcd = 'HRMEAN' then do;

    paramn = 1;

    param="Heart Rate (Beats/min)";

end;

else if egtestcd = 'PRMEAN' then do;

    paramn = 2;

    param="PR Duration (msec)";

end;

else if egtestcd = 'QRS DUR' then do;

    paramn = 6;

    param="QRS Duration (msec)";

end;

else if egtestcd = 'QTMEAN' then do;

    paramn = 3;

    param="QT Duration (msec)";

end;

else if egtestcd = 'QTCB' then do;

    paramn = 4;
```

```

        param="QTcB - Bazett's Correction Formula (msec)";
    end;

    else if egtestcd = 'INTP' then do;

        paramn = 7;

        param="Interpretation";

    end;

    else if egtestcd = 'INTPCM' then do;

        paramn = 8;

        param="INTERPRETATION  COMMENTS";

    end;

    else if egtestcd = 'QTCF' then do;

        paramn = 5;

        param="QTcF - Fridericia's Correction Formula (msec)";

    end;

else if egtestcd='EGALL' then do;

    paramn=9;

    param="All ECG Examinations";

end;

else put 'USER WARN' 'ING: unidentified parameters: ' egtestcd = egtest =;

/*Paramtyp, Dtype*/

if egtestcd = 'QTCF' then do;

    paramtyp = 'DERIVED';

    dtype = 'FUNCTION';

end;

```

```

/*Avalc, Aval*/

if egtestcd = "INTP" and egstresc = "NORMAL" then do;

    avalc = "Normal";

    aval = 0;

end;

else if egtestcd = "INTP" and index(egstresc,"ABNORMAL") then do;

    avalc = "Abnormal";

    aval = 1;

    desc=propcase(scan(EGSTRESC,2,'-'),'');

end;

else if PARAMTYP='DERIVED' then do;

    AVAL=ROUND(EGSTRESN,0.0000000000000001);

    AVALC=STRIP(PUT(AVAL,BEST16.));

    end;

    else do;

        aval = egstresn;

        avalc = trim(egstresc);

    end;

/*Avalu*/

    avalu = trim(egstresu);

    *Timepoints;

    atpt = egtpt;

```

```

atptn = egtptnum;

*Dates;

if not missing(egdtc) then adt = input(egdtc,yymmdd10.);

    format adt date9.;

run;

*****;

* Combine ADSL and EG data *;

*****;

data eg2;

    merge adsl eg1(in=a);

    by usubjid;

    if a;

/*ADAY*/

if not missing (adt) and not missing (trtsdt) then aday = adt - trtsdt + 1;

*Visits;

if COMPLFL = "Y" then do;

    avisit = propcase(visit);

    avisitn = visitnum;

end;

else if find(DISCCAT,"Discontinued", 'l')>0 and visit = "DAY 91/DISCHARGE AMBULATORY" then
do;

    if 7<ADAY<31 then do;

```



```

        AVISIT="DAY 30";

        AVISITN=130;

/*        ATPT="";*/
/*        ATPTN=.;*/

        ATPT="DAY 30";

        ATPTN=2;

end;

else if 32<ADAY<61 then do;

        AVISIT="DAY 60";

        AVISITN=160;

/*        ATPT="";*/
/*        ATPTN=.;*/

        ATPT="DAY 60";

        ATPTN=3;

end;

else do;

        AVISIT=VISIT;

        AVISITN=VISITNUM;

end;

end;

else do;

        AVISIT=VISIT;

        AVISITN=VISITNUM;

end;

```

```

TRTP=TRT01P;

TRTPN=TRT01PN;

TRTA=TRT01A;

TRTAN=TRT01AN;

run;

*ablfl;

proc sort data=eg2; by paramcd usubjid avisitn adt; run;

data base1 base2 base2_;

    set eg2;

    by paramcd usubjid avisitn adt;

    if armcd in ("MCC", "THS 2.2M", "SMABST") then do;

        if not missing(adl) and adl<trtsdt and avisitn <= 101 then do;

            ablfl_="Y";

            output base1;

        end;

        else output base2;

    end;

    else do;

        output base2_;

    end;

run;

data base3;

    set base1;

    by paramcd usubjid avisitn adt;

```

```

        if last.usubjid then ablfl="Y";

run;

data ablfl;

        set base2 base2_ base3;

run;


*****;

* Combine EG and SUPPEG data *;

*****;

proc sort data=ablfl; by usubjid egseq; run;

proc sort data=suppeg; by usubjid egseq; run;

data comb;

        merge ablfl suppeg;

        by usubjid egseq;

run; /*7,288*/


*****;

* Calculate changes from baseline (Screening) ;

*****;

proc sort data = comb; by usubjid paramn avisitn; run;


*baseline, change;

data base (rename=(adt=adt_ avalc=basec aval=base));

        set comb;

```

```
where ABLFL='Y';

keep usubjid paramcd adt avalc aval;

run;

proc sort data=BASE; by usubjid paramcd; run;

proc sort data=comb; by usubjid paramcd; run;

data new;

    merge comb BASE;

    by usubjid paramcd;

run;

data change;

*length shift1 $50;

    set new;

    if ADT<ADT_ then do;

        BASEC="";

        BASE=.;

    end;

    else if egstat="NOT DONE" then do;

        BASEC="";

        BASE=.;

    end;

    if adt>adt_ then do;

        if not missing (aval) and not missing (base) then chg = aval - base;

    end;
```

```
run;
```

```
PROC SORT DATA=change; BY USUBJID PARAMN AVISITN ADT; RUN;
```

```
data eg3 eg4;
```

```
    set change;
```

```
    if egstat="NOT DONE" then output eg3;
```

```
        else output eg4;
```

```
run;
```

```
data eg5;
```

```
    set eg4;
```

```
    by usubjid paramn avisitn ADT;
```

```
    /*Anl01fl*/
```

```
    if SAFAFL="Y" then do;
```

```
        if adt_ ne . and adt>=adt_ and first.avisitn then ANL01FL = "Y";
```

```
            else if ablfl ne "Y" and TRTSDT ne . and adt>=TRTSDT and first.avisitn then ANL01FL =  
"Y";
```

```
    end;
```

```
    if INDEX(UPCASE(AVISIT),'UNSCHEDULED') ^= 0 then ANL01FL="";
```

```
run;
```

```
/*Shift1*/
```

```
data shift (rename=(egclsig=clig_base));
```

```
    set eg5;
```

```
    where ablfl="Y";
```

```

        keep usubjid paramcd egclsig;

run;

proc sort data=shift; by usubjid paramcd; run;

proc sort data=eg5; by usubjid paramcd; run;

data shift_1;

        merge shift eg5;

        by usubjid paramcd;

run;


data shift_2;

length shift1 $50;

        set shift_1;

        if ADT > ADT_ and avalc ne "" and basec ne "" then do;

                if basec="Abnormal" and avalc="Abnormal" then shift1 = trim(basec) || ', ' || trim(egclsig) || '
to ' || trim(avalc) || ', ' || trim(egclsig);

                        else if clig_base ne "" and basec="Abnormal" then shift1 = trim(basec) || ', ' ||
trim(clig_base) || ' to ' || trim(avalc);

                                else if avalc="Normal" and basec="Normal" then shift1 = trim(basec) || ' to '
|| trim(avalc);

                                        else if not missing(egclsig) then shift1 = trim(basec) || ' to ' || trim(avalc) || ', ' ||
trim(egclsig);

                                                end;

run;


data final;

        set shift_2 eg3;

```

```

/*ASPER*/

if AVISITN < 101 then do;

    ASPER=1;

    ASPERC="Pre-Randomization Period";

end;

else if 101<=AVISITN<=106 then do;

    ASPER=2;

    ASPERC="Confinement Period";

end;

else if 106<AVISITN<=191 then do;

    ASPER=3;

    ASPERC="Ambulatory Period";

end;

else if AVISITN>191 then do;

    ASPER=4;

    ASPERC="Safety Follow-up Period";

end;

run;

data final_1;

    set final;

    *for Enrolled not randomized;

    if trtpn in (97,98) then do;

```

```

        ASPERC="Pre-Randomization Period";

        ASPER=1;

    end;

run;

*****.

* create output dataset ;

*****.

*options replace;

data ADEG (drop=PARAM EGCLSIG avalc basec rename=(PARAM_=PARAM EGCLSIG_=EGCLSIG
avalc_1=avalc basec_1=basec));

length PARAM_ $50 EGCLSIG_ $3;

    set final_1;

    PARAM_=PARAM;

    EGCLSIG_=EGCLSIG;

    AVISIT=propcase(avisit);

    ATPT=propcase(atpt);

    if paramcd ^= "INTP" and not missing(avalc) then do;

        if int(input(avalc,best.))=0 and substr(avalc,1,1)="0" then avalc_1=substr(avalc,2);

        else avalc_1=avalc;

    end;

    else avalc_1=avalc;

```



```
if paramcd ^= "INTP" and not missing(basec) then do;

    if int(input(basec,best.))<0 and substr(basec,1,1)="0" then basec_1=substr(basec,2);

    else basec_1=basec;

end;

else basec_1=basec;

run;


%m_attrib_adam(dset=ADEG);


proc sort data=adeg out=adam.adeg(label = 'ECG Analysis Dataset');

    by USUBJID AVISITN ATPTN PARAMCD;

run;


%m_logchk;
```