

```

*****
*****;
*      CLIENT: ModernaTX, Inc.
*      PROTOCOL: mRNA-1273-P301
*      PURPOSE: Create analysis dataset ADTTRE2
*      INPUT FILES: ADB.ADSL, ADB.ADEFF3
*      OUTPUT FILES: ADTTRE2.sas7bdat
*      USAGE NOTES:
*****
*****;
* Copyright 2021 Pharmaceutical Product Development, Inc.
* All Rights Reserved.
*****
*****;

```

```

proc datasets lib=work kill nolist;
quit;
options noquotelenmax;

```

```

*options source2;
*%include "madam.sas";

```

```

/* Assign global macro variable DSETNAME to reflect the name of the final ADaM
dataset */

```

```

%global DSETNAME;
%LET DSETNAME = ADTTRE2;
%LET ADSLVAR = /*STUDYID*/ USUBJID TR01SDT DTHDT EOSDT CUTOFFDT;

```

```

proc contents data=adb.adsl varnum;
run;
proc contents data=adb.adeff3 varnum;
run;

```

```

data adb_adsl;
  set adb.adsl;
run;

```

```

data adsl;
  set adb_adsl;
  keep &ADSLVAR.;
run;

```

```

proc sort data=adsl nodupkey;
  by usubjid;* subjid;
run;

```

```

data adb_adeff3;
  set adb.adeff3;
run;

```

```

data adefff3;
  set adb_adeff3;
  where ANL01FL='Y' and PARCAT1N=1 and FASFL='Y';
run;

proc freq data=adeff3;
  tables paramcd*param / list missing nopercnt nocum;
run;

data all;
  merge adsl (in=a)
        adefff3(in=b drop=TR01SDT); /* Ask (b) (6): TR01SDT from ADEFF3 or ADSL? */
  by usubjid; *subj;
  if b;
  PARAMCD = 'TOTSCORE';
  PARAM    = 'Total COVID-19 Symptoms Score';
run;

/**** Derive EVENTSEQ - Prep data ****/

proc summary data=all nway;
  by usubjid;
  var AVAL;
  output out=all1 sum=sum_aval_subj;
  where PPROTFL='Y';
run;

proc summary data=all1(keep=sum_aval_subj) nway;
  var sum_aval_subj;
  output out=all1a min=min_sum_aval q1=q1_sum_aval median=med_sum_aval
q3=q3_sum_aval max=max_sum_aval;
run;

data all2;
  retain sum_aval_subj;
  merge all
        all1(keep=usubjid sum_aval_subj);
  by usubjid;
run;

data all2a;
  if _n_=1 then set all1a(keep=min_sum_aval q1_sum_aval med_sum_aval q3_sum_aval
max_sum_aval);
  set all2;
run;

/**** Select record of date of First Symptom Occurrence for each subject ****/
/**** If subject does not have flagged above, then select first date ****/

```

```

proc sort data=all2a;
  by usubjid paramcd descending FASYMPFL adt;
run;

data all3;
  set all2a;
  by usubjid paramcd descending FASYMPFL adt;
  if first.usubjid;
run;

/**** Derive EVENTSEQ ****/

data all4;
  set all3;
  if sum_aval_subj=0 or (0 < sum_aval_subj < q1_sum_aval) then do;
    EVENTSEQ=1;
    output;
  end;
  else if (q1_sum_aval <= sum_aval_subj < med_sum_aval) then do;
    EVENTSEQ=1;
    output;
    EVENTSEQ=2;
    output;
  end;
  else if (med_sum_aval <= sum_aval_subj < q3_sum_aval) then do;
    EVENTSEQ=1;
    output;
    EVENTSEQ=2;
    output;
    EVENTSEQ=3;
    output;
  end;
  else if (q3_sum_aval <= sum_aval_subj) then do;
    EVENTSEQ=1;
    output;
    EVENTSEQ=2;
    output;
    EVENTSEQ=3;
    output;
    EVENTSEQ=4;
    output;
  end;
run;

/**** Derive CNSR ****/

data all5;
  set all4;
  if EVENTSEQ=1 then do;
    if (sum_aval_subj = 0) then CNSR=1;

```

```

        else if (sum_aval_subj > 0) then CNSR=0;
    end;
    else if EVENTSEQ in (2 3 4) then CNSR=0;
run;

/** Derive EVNTDESC **/

data all6;
    set all5;
    length EVNTDESC $100;
    if sum_aval_subj=0 then EVNTDESC = 'No
symptom';
    else if (0 < sum_aval_subj < q1_sum_aval) then EVNTDESC = '0 < Total
COVID-19 Symptoms Score < Q1';
    else if (q1_sum_aval <= sum_aval_subj < med_sum_aval) then EVNTDESC = 'Q1 <=
Total COVID-19 Symptoms Score < Median';
    else if (med_sum_aval <= sum_aval_subj < q3_sum_aval) then EVNTDESC = 'Median <=
Total COVID-19 Symptoms Score < Q3';
    else if (q3_sum_aval <= sum_aval_subj) then EVNTDESC = 'Q3 <=
Total COVID-19 Symptoms Score';
run;

/** Derive STARTVAL based on on the date of first symptom occurrence per subject
***/

data all7;
    set all6;
    if EVENTSEQ=1 then STARTVAL=0;
    else if FASYMPFL='Y' then do;
        if EVENTSEQ=2 and cmiss(adt,tr01sdt)=0 then STARTVAL = (adt - tr01sdt);
        else if EVENTSEQ=3 and cmiss(adt,tr01sdt)=0 then STARTVAL = (adt - tr01sdt) +
(1*10**(-10));
        else if EVENTSEQ=4 and cmiss(adt,tr01sdt)=0 then STARTVAL = (adt - tr01sdt) +
(2*10**(-10));
    end;
run;

/** Derive AVAL based on on the date of first symptom occurrence per subject ***/

data all8;
    set all7(drop=aval:);
    if sum_aval_subj=0 then do;
        if EVENTSEQ=1 then do;
            minvalue = min(DTHDT,EOSDT,CUTOFFDT);
            if nmiss(minvalue,tr01sdt)=0 then AVAL = (minvalue - tr01sdt);
        end;
    end;
    else if sum_aval_subj>0 then do;
        if FASYMPFL='Y' then do;
            if EVENTSEQ=1 then AVAL = (adt - tr01sdt);

```

```

        else if EVENTSEQ=2 and cmiss(adt,tr01sdt)=0 then AVAL = (adt - tr01sdt) +
(1*10**(-10));
        else if EVENTSEQ=3 and cmiss(adt,tr01sdt)=0 then AVAL = (adt - tr01sdt) +
(2*10**(-10));
        else if EVENTSEQ=4 and cmiss(adt,tr01sdt)=0 then AVAL = (adt - tr01sdt) +
(3*10**(-10));
    end;
end;
format minvalue date9.;
run;

```

```

data &DSETNAME.;
    set all8;
    drop min_sum_aval q1_sum_aval med_sum_aval q3_sum_aval max_sum_aval minvalue
sum_aval_subj;
run;

```

```

options mprint mlogic symbolgen;
%ADAM_DATASET_UPDATE
(ds          = &DSETNAME,
 libin       = work,
 libout      = output,
 ADSLlib     = adb,
 addcomvar   = Y,
 addseq      = ,
 dropinfmt   = Y,
 mapspecfile = &ADAM_SPEC,
 maploc      = &ADAM_SPEC_LOC,
 debug       = N
);
options nomprint nomlogic nosymbolgen;

```

```

proc contents data=output.adttre2 varnum;
run;

```

**** Generate Value Level Metadata values for the ad<xx> dataset **;**

```

%* DOD_VLM
(type      = ADAM,
 selmems   = &DSETNAME,
 excmems    = ,
 specloc   = &ADAM_SPEC_LOC,
 specname  = &ADAM_SPEC,
 debug     = N
);

```

**** Generate Enhanced Controlled Terminology for the ad<xx> dataset **;**

```

%* DOD_ENHCD_CT
(type      = ADAM,

```

```
    selmems      = &DSETNAME,  
    excmems      = ,  
    specloc      = &ADAM_SPEC_LOC,  
    specname     = &ADAM_SPEC,  
    map_ct_nm    = ,  
    map_ct_loc   = ,  
    debug        = N  
);  
  
proc datasets lib=work kill nolist;  
quit;  
libname _all_ clear;
```