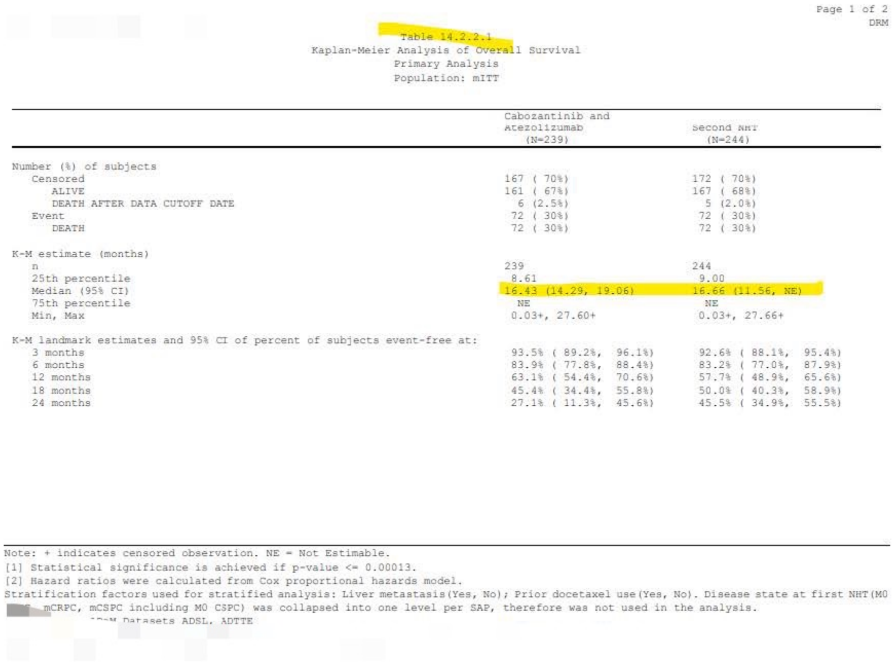
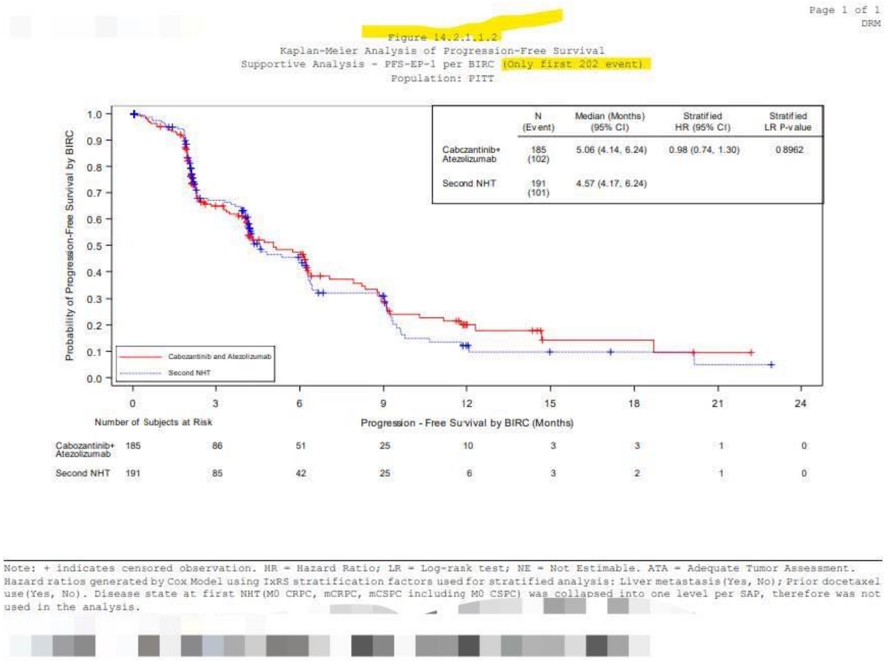
SAS TFL



```
                                                                                                    Page 1 of 2
                                                                                                            DRM
                                              Table 14.2.2.1
                                  Kaplan-Meier Analysis of Overall Survival
                                            Primary Analysis
                                            Population: mITT

------------------------------------------------------------------------------------------------------------
                                                                Cabozantinib and
                                                                Atezolizumab              Second NHT
                                                                   (N=239)                  (N=244)
============================================================================================================

Number (%) of subjects
   Censored                                                     167 ( 70%)               172 ( 70%)
      ALIVE                                                     161 ( 67%)               167 ( 68%)
      DEATH AFTER DATA CUTOFF DATE                                6 (2.5%)                 5 (2.0%)
   Event                                                         72 ( 30%)                72 ( 30%)
      DEATH                                                      72 ( 30%)                72 ( 30%)

K-M estimate (months)
   n                                                            239                      244
   25th percentile                                               8.61                     9.00
   Median (95% CI)                                              16.43 (14.29, 19.06)     16.66 (11.56, NE)
   75th percentile                                               NE                       NE
   Min, Max                                                      0.03+, 27.60+            0.03+, 27.66+

K-M landmark estimates and 95% CI of percent of subjects event-free at:
   3 months                                                      93.5% ( 89.2%,  96.1%)   92.6% ( 88.1%,  95.4%)
   6 months                                                      83.9% ( 77.8%,  88.4%)   83.2% ( 77.0%,  87.9%)
   12 months                                                     63.1% ( 54.4%,  70.6%)   57.7% ( 48.9%,  65.6%)
   18 months                                                     45.4% ( 34.4%,  55.8%)   50.0% ( 40.3%,  58.9%)
   24 months                                                     27.1% ( 11.3%,  45.6%)   45.5% ( 34.9%,  55.5%)


--------------------------------------------------------------------------------------------------------------
Note: + indicates censored observation. NE = Not Estimable.
[1] Statistical significance is achieved if p-value <= 0.00013.
[2] Hazard ratios were calculated from Cox proportional hazards model.
Stratification factors used for stratified analysis: Liver metastasis(Yes, No); Prior docetaxel use(Yes, No). Disease state at first NHT(M0
[redacted] mCRPC, mCSPC including M0 CSPC) was collapsed into one level per SAP, therefore was not used in the analysis.
        Datasets ADSL, ADTTE
```

SAS TFL

Subtitle should be “Dec12 2022 Cut off date”.
Paramcl from PFSEP1A change to PFSEP1C according to changed on cut of date.



[redacted]

Page 1 of 1
DRM

Figure 14.2.1.1.2
Kaplan-Meier Analysis of Progression-Free Survival
Supportive Analysis - PFS-EP-1 per BIRC (Only first 202 event)
Population: PITT

	N (Event)	Median (Months) (95% CI)	Stratified HR (95% CI)	Stratified LR P-value
Cabczantinib+ Atezolizumab	185 (102)	5.06 (4.14, 6.24)	0.98 (0.74, 1.30)	0.8962
Second NHT	191 (101)	4.57 (4.17, 6.24)		

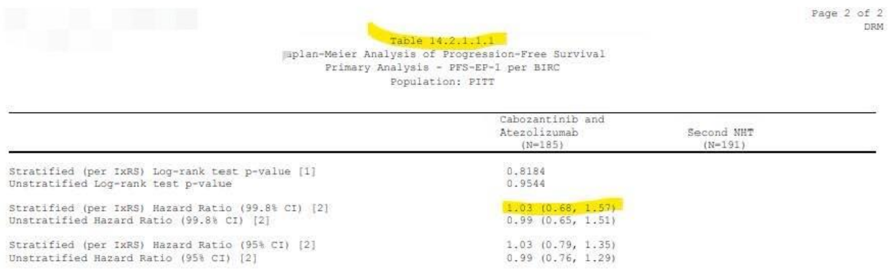
Number of Subjects at Risk									
Cabozantinib+ Atezolizumab	185	86	51	25	10	3	3	1	0
Second NHT	191	85	42	25	6	3	2	1	0

Note: + indicates censored observation. HR = Hazard Ratio; LR = Log-rank test; NE = Not Estimable. ATA = Adequate Tumor Assessment. Hazard ratios generated by Cox Model using IxRS stratification factors used for stratified analysis: Liver metastasis(Yes, No); Prior docetaxel use(Yes, No). Disease state at first NHT(M0 CRPC, mCRPC, mCSPC including M0 CSPC) was collapsed into one level per SAP, therefore was not used in the analysis.

[redacted]

Overall analysis

SAS TFL

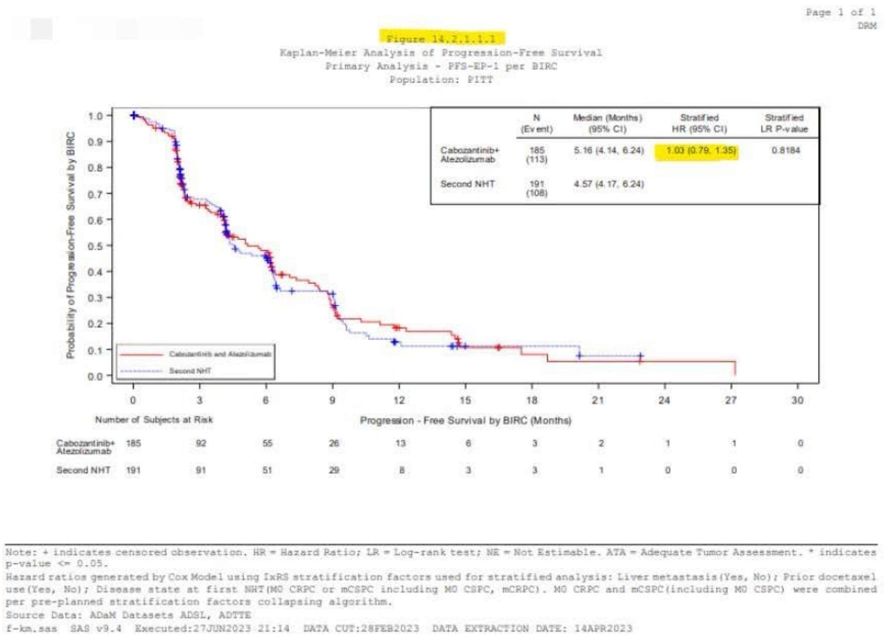


[redacted]

Page 2 of 2
DRM

Table 14.2.1.1.1
[redacted]plan-Meier Analysis of Progression-Free Survival
Primary Analysis - PFS-EP-1 per BIRC
Population: PITT

	Cabozantinib and Atezolizumab (N=185)	Second NHT (N=191)
Stratified (per IxRS) Log-rank test p-value [1]	0.8184	
Unstratified Log-rank test p-value	0.9544	
Stratified (per IxRS) Hazard Ratio (99.8% CI) [2]	1.03 (0.68, 1.57)	
Unstratified Hazard Ratio (99.8% CI) [2]	0.99 (0.65, 1.51)	
Stratified (per IxRS) Hazard Ratio (95% CI) [2]	1.03 (0.79, 1.35)	
Unstratified Hazard Ratio (95% CI) [2]	0.99 (0.76, 1.29)	



[redacted]

Page 1 of 1
DRM

Figure 14.2.1.1.1
Kaplan-Meier Analysis of Progression-Free Survival
Primary Analysis - PFS-EP-1 per BIRC
Population: PITT

	N (Event)	Median (Months) (95% CI)	Stratified HR (95% CI)	Stratified LR P-value
Cabozantinib+ Atezolizumab	185 (113)	5.16 (4.14, 6.24)	1.03 (0.79, 1.35)	0.8184
Second NHT	191 (108)	4.57 (4.17, 6.24)		

Number of Subjects at Risk

	0	3	6	9	12	15	18	21	24	27	30
Cabozantinib+ Atezolizumab	185	92	55	26	13	6	3	2	1	1	0
Second NHT	191	91	51	29	8	3	3	1	0	0	0

Note: + indicates censored observation. HR = Hazard Ratio; LR = Log-rank test; NE = Not Estimable. ATA = Adequate Tumor Assessment. * indicates p-value <= 0.05.
Hazard ratios generated by Cox Model using IxRS stratification factors used for stratified analysis: Liver metastasis(Yes, No); Prior docetaxel use(Yes, No); Disease state at first NHT(M0 CRPC or mCSPC including M0 CSPC, mCRPC). M0 CRPC and mCSPC(including M0 CSPC) were combined per pre-planned stratification factors collapsing algorithm.
Source Data: ADaM Datasets ADSL, ADTTE
f-km.sas SAS v9.4 Executed:27JUN2023 21:14 DATA CUT:28FEB2023 DATA EXTRACTION DATE: 14APR2023

Table 14-2-1-1-1 KM-pfs-ep1

For stratified trt cabo+atez HR 95% CI, figure range should be (0.68,1.57).

Table 14-2-1-1-2

SAS Writing sample

```
**** 20230612 Mike;
proc sort data=anadata.adsl out=adsl0;
 by usubjid;
 where mittfl="Y";
run;

data adsl0;
     set adsl0;
     trtn=trt01pdn;
     trtc=trt01pd;
run;

******* ODSDATA (method 1);
*** End of treatment (reason);
proc sort data=odsdata.etcdb out=etcdb0 ;
   by studyid usubjid ;
   *where ETDCST ne " ";
run;

*** Radiographic follow-up;
proc sort data=odsdata.ercdb out=ercdb0 ;
   by studyid usubjid ;
   *where ETDCST ne " ";
run;

*** Survival Follow-up;
proc sort data=odsdata.escdb out=escdb0 ;
   by studyid usubjid ;
   *where ETDCST ne " ";
run;

data etcdb0;
     merge etcdb0(in=ina) adsl0(in=inb keep=usubjid trtn trtc trt01pdn trt01pd);
     by usubjid;
     if ina and inb;
run;

data ercdb0;
     merge ercdb0(in=ina) adsl0(in=inb keep=usubjid trtn trtc trt01pdn trt01pd);
     by usubjid;
     if ina and inb;
run;

data escdb0;
     merge escdb0(in=ina) adsl0(in=inb keep=usubjid trtn trtc trt01pdn trt01pd);
     by usubjid;
     if ina and inb;
run;

proc freq data=etcdb0 noprint;
  tables trtn*trtc*PAGEID*etdcst/list missing out=_et_f;;
run;

proc freq data=ercdb0 noprint;
  tables trtn*trtc*esradrs/list missing out=_er_f;;
run;

proc freq data=escdb0 noprint;
  tables trtn*trtc*esdcfu/list missing out=_es_f;;
run;
*********************************************;

******** (method 2);
proc sort data=sdtmdata.ds out=ds0 ;
  by usubjid;
run;

data ds0;
```

```
     merge ds0(in=ina) adsl0(keep=usubjid trtn trtc trt01pdn trt01pd);
     by usubjid;
     if ina;
run;

proc freq data=ds0 noprint;
   tables trtn*trtc*dscat*dsscat*dsterm/list missing out=_ds_f;
run;

data _ds_f1;
     set _ds_f;
     if dsscat="END OF STUDY TREATMENT, ATEZOLIZUMAB";
run;

proc freq data=adsl0;
  tables mittfl*trtn*trtc/list missing;
  tables mittfl*ittfl*trtn*trtc/list missing;
run;

data _findat;
  set findat;
  keep usubjid mittfl ittfl ;
run;

proc freq data=findat;
  tables ittfl/list missing;
run;


****** Part II Subjects treated;
/*ADSL:TRTCSDT and ADSL:TRTASDT non-missing*/
/*(ADSL:TRTBSDT and ADSL:TRTPSDT  non-missing) or ADSL:TRTESDT non-missing*/
;

data adsl0;
     set adsl0;
     if saffl="Y" then do;
        if trtcsdt>. and trtasdt>. then flag21=1;
        if (trtbsdt>. and trtpsdt>. ) or trtedt>. then flag22=1;
     end;
run;

proc freq data=adsl0;
     tables trtn*trtc*flag21/list missing;
     tables trtn*trtc*flag22/list missing;
     tables trtn*trtc*flag21*flag22/list missing;
run;


***** Part III: subject on study treatment at data cutoff
**Ensure MSAFFL=’Y’ for this section:
ADSL: EOTDCDT >= ADSL:CUTDT or ADSL: EOTDCDT missing
For trt01ad= Cabozantinib+Atezolizumab: ADSL: EOCODCDT >= ADSL:CUTDT or ADSL: EOCODCDT missing
For trt01ad= Cabozantinib+Atezolizumab: ADSL: EOAODT >= ADSL:CUTDT or ADSL: EOAODT missing
For trt01ad= Second NHT  and TRTNHT=Abi+Pred:( (ADSL: EOBODCDT >= ADSL:CUTDT or ADSL: EOBODCDT missing) and (EOPODCDT >= ADSL:CUTDT or ADSL: EOPODCDT missing) )
For trt01ad= Second NHT and TRTNHT=Enza: EOEODCDT >= ADSL:CUTDT or ADSL: EOEODCDT missing;



**** Part IV: subjects discontinued any study treatment ;
/*ADSL:DCTREASC or ADSL:DCTREASA or ADSL:DCTREASB or ADSL:DCTREASP or ADSL:DCTREASE or ADSL:EOCDCDT or ADSL:EOADCDT or ADSL:EOBDCDT or ADSL:EOPDCDT   or ADSL:EOEDCDT  */
/*ADSL:DCTREASC or ADSL:EOCDCDT non-missing */
/*ADSL:DCTREASA or  ADSL:EOADCDT non-missing*/
/*(ADSL:DCTREASC and ADSL:DCTREASA) or (ADSL:EOCDCDT and ADSL:EOADCDT) non-missing*/
```

```
*** steven solutions ;
%let suffix=mitt;
data adsl1;
set anadata.adsl;
/*if &suffix.fl eq "Y" and trt01a=trt01ad and trt01ad ne ' '; */
/*if trt01adN = . then delete; */

if &suffix.fl eq "Y" and trt01p=trt01pd and trt01pd ne ' ';
if trt01pdN = . then delete;
     trtn=trt01pdn;
     trtc=trt01pd;
     if mittfl="Y";
run;

***1: ITT Population ;
proc freq data=adsl1;
 tables trtn*trtc/list missing;
run;

proc freq data=adsl1;
 tables trtn*trtc*ittfl/list missing;
run;

*** 2: subjets treated with ;
data adsl2;
    set adsl1;
     if saffl="Y" then do;
        if trtcsdt>. and trtasdt>. then flag21=1;
        if (trtbsdt>. and trtpsdt>. ) or trtedt>. then flag22=1;
     end;
run;
proc freq data=adsl2;
 tables trtn*trtc*flag21/list missing;
 tables trtn*trtc*flag22/list missing;
run;

*** 3 subjects on study treatment at cut-off ;
data adsl3;
     set adsl2;
     if (EOCOdcDT >= CUTDT or EOCOdcDT eq .) then flag31=1;
     if (EOAODT >= CUTDT or EOAODT eq . ) then flag32=1;
     if (EOBODCDT >= CUTDT or EOBODCDT eq .) then flag33_1=1;
     if (EOPODCDT >= CUTDT or EOPODCDT eq .) then flag33_2=1;
     if (EOEODCDT >= CUTDT or EOEODCDT eq .a) then flag33_3=1;
     if (flag33_1=1 and flag33_2=1) or flag33_3=1 then flag33=1;
run;

proc freq data=adsl3;
 tables trtn*trtc*flag31/list missing;
 tables trtn*trtc*flag32/list missing;
 tables trtn*trtc*flag33/list missing;
run;


*** 4  subjects discontinued any study treatment;
data adsl4;
     set adsl2;
     if  (^missing(DCTREASC) or ^missing(EOCDCDT))  then flag41=1;
     if  (^missing(DCTREASA) or ^missing(EOADCDT))  then flag42=1;
     if flag41=1 and flag42=1 then flag43=1;
     if  (^missing(DCTREASB) or ^missing(EOBDCDT))  then flag44_1=1;
     if  (^missing(DCTREASP) or ^missing(EOPDCDT))  then flag44_2=1;
     if  (^missing(DCTREASE) or ^missing(EOEDCDT))  then flag44_3=1;
     if  (flag44_1=1 and flag44_2=1) or flag44_3=1 then flag44=1;
     if flag41=1 or flag42=1 or flag43=1 or flag44=1 then flag40=1;
     *if  (^missing(DCTREASC) and  ^missing(DCTREASA)) or (^missing(EOCDCDT) or
     ^missing(EOADCDT)) then flag43=3;
     *if ( (^missing(EOBDCDT) and  ^missing(EOPDCDT)  or ^missing(EOEDCDT)) or
        ( ^missing(DCTREASB) and  ^missing(DCTREASP) or ^missing(DCTREASE)) then
        flag44=4;
run;

proc freq data=adsl4;
 tables trtn*trtc*flag40/list missing;
 tables trtn*trtc*flag41/list missing;
 tables trtn*trtc*flag42/list missing;
 tables trtn*trtc*flag43/list missing;
 tables trtn*trtc*flag44/list missing;
run;


***6 primary reason for discontinuation from study treatment ;
/*DCTREASA  Reason for Discont of Atez*/
/*DCTREASC  Reason for Discont of Cabo*/
/*DCTREASB  Reason for Discont of Abi*/
/*DCTREASP  Reason for Discont of Pred*/
/*DCTREASE  Reason for Discont of Enza*/
/*DS.DSDECOD where DS.DSSCAT='END OF STUDY TREATMENT, ATEZOLIZUMAB'*/
/*DS.DSDECOD where DS.DSSCAT='END OF STUDY TREATMENT, CABOZANTINIB'*/
/*DS.DSDECOD where DS.DSSCAT='END OF STUDY TREATMENT, ABIRATERONE'*/
/*DS.DSDECOD where DS.DSSCAT='END OF STUDY TREATMENT, PREDNISONE'*/
/*DS.DSDECOD where DS.DSSCAT='END OF STUDY TREATMENT, ENZALUTAMIDE'*/

proc freq data=adsl4;
 tables DCTREASA/list missing;
run;

data adsl6;
     length DOCTREAS $200.;
     set adsl4;
     if DCTREASA>'' then do;
            DOCTREAS=strip(DCTREASA);
            _trtn=0;
            output;
     end;
     if DCTREASC>'' then do;
            DOCTREAS=strip(DCTREASC);
            _trtn=0;
            output;
     end;
     if DCTREASB>'' then do;
            DOCTREAS=DCTREASB;
            _trtn=1;
            output;
     end;
         if DCTREASP>'' then do;
            DOCTREAS=DCTREASP;
            _trtn=1;
            output;
     end;
         if DCTREASE>'' then do;
            DOCTREAS=DCTREASE;
            _trtn=1;
            output;
     end;

run;

proc sort data=adsl6 nodupkey;
 by _trtn trtn trtc usubjid DOCTREAS;
run;

proc freq data=adsl6;
 tables _trtn*trtn*DOCTREAS/list missing;
run;

```

```
/**************************************************
*H*
*H* PROGRAM:           v_a_adsl.sas
*H*
*H* USAGE:                         QC adsl for 312 CSR
*H*
*H* REQUIRES anadata:
*H* REQUIRES odsdata:
*H* REQUIRES sdtmdata:  AE   CD           CM           DD           DM           DS           EX           IE           PR           QS           SU
                SUPPDM  SUPPDS   SV           VS
*H* REQUIRES macros:    dtimpute merge_supp probdes psort getQS labels
*H*
*H* PRODUCES anadata:   adsl (in QC folder)
*H*
*H* REVISION HISTORY:
*H*     20220320 SC    315
*H*     20230515 SC    pittfl population changed from 324 to 400
*H* NOTES:
*H*
*H* $Id:$
*************************************************/


* libname sdtmdata "S:\stat\xl184\184315\dev\sdtmdata\cro\20230221";
* libname anadata "S:\stat\xl184\184315\dev\anadata\cro\20220725";



%macro attrib_adsl;
attrib STUDYID         length=$8.          format=$8.               label="Study Identifier";
attrib USUBJID         length=$16.         format=$16.              label="Unique Subject Identifier";
attrib SUBJID          length=$8.          format=$8.               label="Subject Identifier for the Study";
attrib SITEID          length=$4.          format=$4.               label="Study Site Identifier";
attrib COUNTRY         length=$3.          format=$3.               label="Country";
attrib CNTRYGR1        length=$40.         format=$40.              label="Pooled Country Group 1";
attrib STRATAR         length=$200.        format=$200.             label="Strata Used for Randomization";
attrib STRATARN        length = $200.            label="Strata Used for Randomization (N)";
attrib REGIONI         length=$10.         format=$10.              label="Geographic Region, IxRS";
attrib REGNIGR1        length=$10.         format=$10.              label="Pooled Geog Region, IxRS Group 1";
attrib REGION1                length=$40.         format=$40.              label="Geographic Region 1";
attrib SCRNFFL               length=$1.          format=$1.               label="Reason Arm and/or Actual Arm is Null";
attrib ETIOIXRS        length=$200.        format=$200.             label="Disease Etiology, IxRS";
attrib DSIXRS          length=$200.        format=$200.             label="Pres of Extrahep Dis or Macro Inv, IxRS";
attrib STRATAV         length=$200.        format=$200.             label="Strata from Verification Source ";
attrib STRATAVN              format=best20.            label="Strata from Verification Source (N)";
attrib REGIONC         length=$10.         format=$10.              label="Geographic Region, CRF";
attrib ETIOCRF         length=$200.        format=$200.             label="Disease Etiology, CRF";
attrib DSCRF           length=$200.        format=$200.             label="Pres of Extrahep Dis or Macro Inv, CRF";
attrib PROTVS          length=$20.         format=$20.              label="Protocol Version Subject Enrolled";
attrib AGE                   format=best20.            label="Age";
attrib AGEU            length=$10.         format=$10.              label="Age Units";
attrib AGEGR1          length=$10.         format=$10.              label="Pooled Age Group 1";
attrib AGEGR2          length=$10.         format=$10.              label="Pooled Age Group 2";
attrib AGEGR3          length=$10.         format=$10.              label="Pooled Age Group 3";
attrib SEX             length=$1.          format=$1.               label="Sex";
attrib RACE            length=$200.        format=$200.             label="Race";
attrib RACEOTH         length=$200.        format=$200.             label="Race Other Specify ";
attrib RACEMULT        length=$200.        format=$200.             label="Race Multiple";
attrib RACEGR1         length=$40.         format=$40.              label="Pooled Race Group 1";
attrib ETHNIC          length=$22.         format=$22.              label="Ethnicity";
attrib SAFFL           length=$1.          format=$1.               label="Safety Population Flag";
attrib ITTFL           length=$1.          format=$1.               label="Intent-To-Treat Population Flag";
attrib PITTFL          length=$1.          format=$1.               label="Primary PFS Population Flag";
attrib ELIGIBFL        length=$1.          format=$1.               label="Met All Eligibility Criteria Flag";
attrib ARM            length=$200.      format=$200.          label="Description of Planned Arm";
attrib ACTARM         length=$200.      format=$200.          label="Description of Actual Arm";
attrib TRTSEQP        length=$200.      format=$200.          label="Planned Sequence of Treatments";
attrib TRT01P         length=$25.       format=$25.           label="Planned Treatment for Period 01";
attrib TRT01PN                  format=best20.          label="Planned Treatment for Period 01 (N)";
attrib TRT01A         length=$25.       format=$25.           label="Actual Treatment for Period 01 ";
attrib TRT01AN                  format=best20.          label="Actual Treatment for Period 01 (N)";
attrib TRT02P         length=$25.       format=$25.           label="Planned Treatment for Period 02";
attrib TRT02PN                  format=best20.          label="Planned Treatment for Period 02 (N)";
attrib TRT02A         length=$25.       format=$25.           label="Actual Treatment for Period 02 ";
attrib TRT02AN                  format=best20.          label="Actual Treatment for Period 02 (N)";
attrib XVRDT                    format=date9.           label="Crossover Date/Time";
attrib PGICYN         length=$2.        format=$2.            label="Consent to Pharmacogenetic Blood Sample?";
attrib RFICDT                   format=date9.           label="Date of Informed Consent";
attrib RFIC2DT                  format=date9.           label="Date of Informed Consent 2";
attrib RANDDT                   format=date9.           label="Date of Randomization";
attrib TRTSDT                   format=date9.           label="Date of First Exposure to Treatment";
attrib TRTSDTM              format=datetime19.      label="Datetime of First Exposure to Treatment";
attrib TRTASDT                  format=date9.           label="Date of First Exposure to Atezo, Core";
attrib TRTCSDT                  format=date9.           label="Date of First Exposure to Cabo, Core";
attrib TRTSSDT                  format=date9.           label="Date of First Exposure to Sora, Core";
attrib TRTAXSDT                 format=date9.           label="Date of First Exposure to Atezo, XVR";
attrib TRTCXSDT                 format=date9.           label="Date of First Exposure to Cabo, XVR";
attrib RFXENDTC       length=$19.       format=$19.           label="Date/Time of Last Study Treatment";
attrib TRTEDT                   format=date9.           label="Date of Last Exposure to Treatment";
attrib TRTEDTF        length=$1.        format=$1.            label="Date of Last Exposure Imput. Flag";
attrib TR01SDT                  format=date9.           label="Date of First Exposure in Period 01";
attrib TR01STM                  format=time5.           label="Time of First Exposure in Period 01";
attrib TR01SDTM             format=datetime19.      label="Datetime of First Exposure in Period 01";
attrib TR01EDT                  format=date9.           label="Date of Last Exposure in Period 01";
attrib TR01ETM                  format=time5.           label="Time of Last Exposure in Period 01";
attrib TR01EDTM             format=datetime19.      label="Datetime of Last Exposure in Period 01";
attrib TR02SDT                  format=date9.           label="Date of First Exposure in Period 02";
attrib TR02STM                  format=time5.           label="Time of First Exposure in Period 02";
attrib TR02SDTM             format=datetime19.      label="Datetime of First Exposure in Period 02";
attrib TR02EDT                  format=date9.           label="Date of Last Exposure in Period 02";
attrib TR02ETM                  format=time5.           label="Time of Last Exposure in Period 02";
attrib TR02EDTM             format=datetime19.      label="Datetime of Last Exposure in Period 02";
attrib TRTAEDT                  format=date9.           label="Date of Last Exposure to Atezo, Core";
attrib TRTAEDTF       length=$1.        format=$1.            label="Date Last Exp Atez Core Imput. Flag";
attrib TRTCEDT                  format=date9.           label="Date of Last Exposure to Cabo, Core";
attrib TRTCEDTF       length=$1.        format=$1.            label="Date Last Exp Cabo Core Imput. Flag";
attrib TRTSEDT                  format=date9.           label="Date of Last Exposure to Sora, Core";
* attrib TRTSEDTF     length=$1.        format=$1.            label="Date Last Exp Sora Core Imput. Flag";
attrib TRTAXEDT                 format=date9.           label="Date of Last Exposure to Atezo, XVR";
attrib TRTCXEDT                 format=date9.           label="Date of Last Exposure to Cabo, XVR";
attrib LTRTOGDT                 format=date9.           label="Date of Last Dose Ongoing";
attrib EOADCDT                  format=date9.           label="Date of Decision to Discont. Atez";
attrib EOCDCDT                  format=date9.           label="Date of Decision to Discont. Cabo";
attrib EOSDCDT                  format=date9.           label="Date of Decision to Discont. Sora";
attrib EOXADCDT                 format=date9.           label="Date of Dec to Discont. XVR Atez";
attrib EOXCDCDT                 format=date9.           label="Date of Dec to Discont. XVR Cabo";
attrib EOTDCDT                  format=date9.           label="Date of Decision to Discont. Core Tx";
attrib EOTXDCDT                 format=date9.           label="Date of Dec to Discont. XVR Tx";
attrib EOTDCODT                 format=date9.           label="Date of Last Dose Decision Ongoing, Core";
attrib EOXDCODT                 format=date9.           label="Date of Last Dose Decision Ongoing, XVR";
attrib EOAODCDT                 format=date9.           label="Date of Last Dose Dec Ongo (Atez, Core)";
attrib EOAODT                   format=date9.           label="Date of Last Dose Ongo (Atez, Core)";
attrib EOCODCDT                 format=date9.           label="Date of Last Dose Dec Ongo (Cabo, Core)";
attrib EOSODCDT                 format=date9.           label="Date of Last Dose Dec Ongo (Sora, Core)";
attrib WDCFDT                   format=date9.           label="Date of Withdrawal of Full Consent";
attrib DSEVTADT                 format=date9.           label="Date of Disposition Event (Atez, Core)";
attrib DSEVTCDT                 format=date9.           label="Date of Disposition Event (Cabo, Core)";
attrib DSEVTSDT                 format=date9.           label="Date of Disposition Event (Sora, Core)";
attrib DSEVXADT                 format=date9.           label="Date of Disposition Event (Atez, XVR)";
attrib DSEVXCDT     format=date9.     label="Date of Disposition Event (Cabo, XVR)";
attrib LSTALVDT     format=date9.     label="Date Last Known Alive";
attrib LSALVDTF     length=$1.     format=$1.     label="Date Last Known Alive Imp. Flag";
attrib DTHDT     format=date9.     label="Date of Death";
attrib DTHDTF     length=$1.     format=$1.     label="Date of Death Imputation Flag";
attrib DTHDTC     length=$19.     format=$19.     label="Date/Time of Death";
attrib DTHUNFDT     format=date9.     label="Date of Death Unfiltered";
attrib DTHUNDTF     length=$1.     format=$1.     label="Date of Death Unf Imp Flag";
attrib DTHDY     format=best20.     label="Relative Day of Death";
attrib DTHSL0DY     format=best20.     label="Day of Death Rel to Last Non-Zero Dose";
attrib DTHPER     length=$50.     format=$50.     label="Death Period (30 Days)";
attrib DTHAESFL     length=$1.     format=$1.     label="Death After Ext Safety Obs Per Flag";
attrib DTHAEAFL     length=$1.     format=$1.     label="Death After Ext Atezo Saf Obs Per Flag";
attrib DTHAECFL     length=$1.     format=$1.     label="Death After Ext Combo Saf Obs Per Flag";
attrib EOSOBSDT     format=date9.     label="Date of Start of SOP (Standard, Core)";
attrib EOSSXSDT     format=date9.     label="Date of Start of SOP (Standard, XVR)";
attrib EOECASDT     format=date9.     label="Date of Start of SOP (Ext, Core, Atezo)";
attrib EOECCSDT     format=date9.     label="Date of Start of SOP (Ext, Core, Combo)";
attrib EOECMSDT     format=date9.     label="Date of Start of SOP (Ext, Core, Mono)";
attrib EOEXASDT     format=date9.     label="Date of Start of SOP (Ext, XVR, Atezo)";
attrib EOEXCSDT     format=date9.     label="Date of Start of SOP (Ext, XVR, Combo)";
attrib EOSOBEDT     format=date9.     label="Date of End of SOP (Standard, Core)";
attrib EOSSXEDT     format=date9.     label="Date of End of SOP (Standard, XVR)";
attrib EOECAEDT     format=date9.     label="Date of End of SOP (Ext, Core, Atezo)";
attrib EOECCEDT     format=date9.     label="Date of End of SOP (Ext, Core, Combo)";
attrib EOECMEDT     format=date9.     label="Date of End of SOP (Ext, Core, Mono)";
attrib EOEXAEDT     format=date9.     label="Date of End of SOP (Ext, XVR, Atezo)";
attrib EOEXCEDT     format=date9.     label="Date of End of SOP (Ext, XVR, Combo)";
attrib CUTDT     format=date9.     label="Date of Data Cutoff Date";
attrib SYSACTDT     format=date9.     label="First Systemic Non-Rad Therapy Date";
attrib SYSACTWK     format=best20.     label="Time to First Systemic Non-Rad (Week)";
attrib FSBACTDT     format=date9.     label="First Subsequent Anticancer Therapy Date";
attrib TRCRDURD     format=best20.     label="Total Treatment Duration, Core (Days)";
attrib TRCRDURW     format=best20.     label="Total Treatment Duration, Core (Weeks)";
attrib TRCRDURM     format=best20.     label="Total Treatment Duration, Core (Months)";
attrib TRTXDURD     format=best20.     label="Total Treatment Duration, XVR (Days)";
attrib TRTXDURW     format=best20.     label="Total Treatment Duration, XVR (Weeks)";
attrib TRTXDURM     format=best20.     label="Total Treatment Duration, XVR (Months)";
attrib TRTDURD     format=best20.     label="Total Treatment Duration (Days)";
attrib TRTDURW     format=best20.     label="Total Treatment Duration (Weeks)";
attrib TRTDURM     format=best20.     label="Total Treatment Duration (Months)";
attrib TRTADURD     format=best20.     label="Total Atezolizumab Duration (Days)";
attrib TRTCDURD     format=best20.     label="Total Cabozantinib Duration (Days)";
attrib TRTCDURM     format=best20.     label="Total Cabozantinib Duration (Months)";
attrib TRTSDURD     format=best20.     label="Total Sorafenib Duration (Days)";
attrib TRTSDURM     format=best20.     label="Total Sorafenib Duration (Months)";
attrib TROCDURD     format=best20.     label="Total Ongoing Cabo Duration (Days)";
attrib TROCDURM     format=best20.     label="Total Ongoing Cabo Duration (Months)";
attrib TROSDURD     format=best20.     label="Total Ongoing Sora Duration (Days)";
attrib TROSDURM     format=best20.     label="Total Ongoing Sora Duration (Months)";
attrib DCTREASA     length=$200.     format=$200.     label="Reason for Discont of Atez, Core";
attrib DCTREASC     length=$200.     format=$200.     label="Reason for Discont of Cabo, Core";
attrib DCTREASS     length=$200.     format=$200.     label="Reason for Discont of Sora, Core";
attrib DCTRESXA     length=$200.     format=$200.     label="Reason for Discont of Atez, XVR";
attrib DCTRESXC     length=$200.     format=$200.     label="Reason for Discont of Cabo, XVR";
attrib DTHCAUS     length=$200.     format=$200.     label="Cause of Death";
attrib DTHCGR1     length=$200.     format=$200.     label="Cause of Death Group 1 (Preferred Term)";
attrib DTHASOSI     length=$75.     format=$75.     label="Death Associated With Study Indication";
attrib GR5RELFL     length=$1.     format=$1.     label="GR5 AE Related to Study Treatment Flag";
attrib EOSRDTC     length=$19.     format=$19.     label="Non-imputed End of Survival FUP Date";
attrib EOSDT     format=date9.     label="End of Survival FUP Date";
attrib EOSDTF     length=$1.     format=$1.     label="End of Study Date Imputation Flag";
attrib DCSREAS     length=$200.     format=$200.     label="Reason for Discont. From Survival FUP";
attrib EORDT     format=date9.     label="Date of End of Radiographic FUP";
attrib DCRREAS     length=$200.     format=$200.     label="Reason for Discont. From Radio FUP";
attrib LSTVISDT          format=date9.         label="Last Visit Date";
attrib WTBL              format=best20.        label="Baseline Weight (kg)";
attrib WTBLGR1       length=$15.      format=$15.          label="Pooled Baseline Weight (kg) Group 1";
attrib HTBL              format=best20.        label="Baseline Height (cm)";
attrib BMIBL             format=best20.        label="Baseline BMI  (kg/m2)";
attrib ECOGBL            format=best20.        label="Baseline ECOG per eCRF";
attrib SMOKSTAT      length=$10.      format=$10.          label="Smoking Status";
attrib ALCHSTAT      length=$10.      format=$10.          label="Alcohol Status";
attrib DIAGDT            format=date9.         label="Cancer Diagnosis Date";
attrib DIAGDTF       length=$1.       format=$1.           label="Cancer Diagnosis Date Imputation Flag";
attrib DIAGDURD          format=best20.        label="Duration of Diagnosis to Rand (Days)";
attrib DIAGDURY          format=best20.        label="Duration of Diagnosis to Rand (Years)";
attrib NRADSDT           format=date9.         label="Last Non-Rad Tx Start Date";
attrib NRADSDTF      length=$1.       format=$1.           label="Last Non-Rad Tx Start Imp Flag";
attrib NRADEDT           format=date9.         label="Last Non-Rad Therapy End Date";
attrib NRADEDF       length=$1.       format=$1.           label="Last Non-Rad Tx End Imp Flag";
attrib NRADDURM          format=best20.        label="Dur of Last Non-Rad Tx to Rand (Months)";
attrib PNRDTXFL      length=$1.       format=$1.           label="Prior Local Non-Rad Therapy Flag ";
attrib NPNRADTX          format=best20.        label="Number of Prior Local Non-Rad Thx";
attrib NPNRDGR1      length=$10.      format=$10.          label="Num of Prior Local Non-Rad Tx Group 1";
attrib PRADTXFL      length=$1.       format=$1.           label="Prior Radiation Therapy Flag";
attrib NPRADTX           format=best20.        label="Number of Prior Radiation Therapies";
attrib NPRADGR1      length=$10.      format=$10.          label="Num of Prior Radiation Therapy Group 1";
attrib PSAFFL   length=$1.  format=$1.    label="Primary Safety Population Flag";
attrib DTHPER2   length=$50.  format=$50.    label="Death Period (100 Days)";
attrib SYSACTD   format=BEST20.  label="Time to First Systemic Non-Rad (Days)";
attrib TRTADURM  format=BEST20.  label="Total Atezolizumab Duration (Months)";
attrib FNHTGR1       length=$20.      format=$10.          label="First NHT Collapsed Group";
attrib DOCGR1        length=$10.      format=$10.          label="Prior Docetaxel Collapsed Group";
attrib DSGR1         length=$10.      format=$10.          label="Pres of Liver meta Collapsed Group";
%mend attrib_adsl;

%let var_list = %str(STUDYID, USUBJID, SUBJID, SITEID, COUNTRY, CNTRYGR1, STRATAR, STRATARN, REGIONI, REGNIGR1,
ETIOIXRS, DSIXRS,
STRATAV, STRATAVN, REGIONC, ETIOCRF, DSCRF, PROTVS, AGE, AGEU, AGEGR1, AGEGR2, AGEGR3, SEX, RACE, RACEOTH,
RACEMULT, RACEGR1,
ETHNIC, SAFFL, ITTFL, PITTFL, ELIGIBFL, ARM, ACTARM, TRTSEQP, TRT01P, TRT01PN, TRT01A, TRT01AN, TRT02P, TRT02PN,
TRT02A, TRT02AN,
XVRDT, PGICYN, RFICDT, RFIC2DT, RANDDT, TRTSDT, TRTSDTM, TRTASDT, TRTCSDT, TRTSSDT, TRTAXSDT, TRTCXSDT,
RFXENDTC,
TRTEDT, TRTEDTF, TR01SDT, TR01STM,TR01SDTM, TR01EDT, TR01ETM, TR01EDTM, TR02SDT, TR02STM, TR02SDTM, TR02EDT,
TR02ETM, TR02EDTM,
TRTAEDT, TRTAEDTF, TRTCEDT, TRTCEDTF, TRTSEDT, TRTSEDTF, TRTAXEDT, TRTCXEDT, LTRTOGDT, EOADCDT, EOCDCDT,
EOSDCDT,
EOXADCDT, EOXCDCDT, EOTDCDT, EOTXDCDT, EOTDCODT, EOXDCODT, EOAODCDT, EOAODT, EOCODCDT, EOSODCDT,
WDCFDT, DSEVTADT, DSEVTCDT, DSEVTSDT, DSEVXADT, DSEVXCDT, LSTALVDT, LSALVDTF, PROTVS,
DTHDT, DTHDTF, DTHDTC, DTHUNFDT, DTHUNDTF, DTHDY, DTHSL0DY, DTHPER, DTHAESFL, DTHAEAFL, DTHAECFL,
CMSTYN,
EOSOBSDT, EOSSXSDT, EOECASDT, EOECCSDT, EOECMSDT, EOEXASDT, EOEXCSDT, EOSOBEDT, EOSSXEDT, EOECAEDT,
EOECCEDT, EOECMEDT, EOEXAEDT, EOEXCEDT,
CUTDT, SYSACTDT, SYSACTWK, FSBACTDT, TRCRDURD, TRCRDURW, TRCRDURM, TRTXDURD, TRTXDURW, TRTXDURM,
TRTDURD, TRTDURW, TRTDURM,
TRTADURD, TRTCDURD, TRTCDURM, TRTSDURD, TRTSDURM, TROCDURD, TROCDURM, TROSDURD, TROSDURM,
DCTREASA, DCTREASC, DCTREASS, DCTRESXA, DCTRESXC,
DTHCAUS, DTHCGR1, DTHASOSI, GR5RELFL, EOSRDTC, EOSDT, EOSDTF, DCSREAS, EORDT, DCRREAS, LSTVISDT, WTBL,
WTBLGR1, HTBL, BMIBL, ECOGBL,
SMOKSTAT, ALCHSTAT, DIAGDT, DIAGDTF, DIAGDURD, DIAGDURY, NRADSDT, NRADSDTF, NRADEDT, NRADEDF,
NRADDURM, PNRDTXFL, NPNRADTX, NPNRDGR1, PRADTXFL, NPRADTX, NPRADGR1,
PSAFFL, DTHPER2, SYSACTD, TRTADURM, DURNRADM, DNRADGR1 FNHTGR1 DOCGR1 DSGR1 opittfl msaffl mittfl LTFUDT
CUTRNKDT RANDCDTM
CUTRDDTM DURNRADM DNRADGR1 dcsfl dcrfl


);
```

```
%let subject=  ;
%let block = ;

*PN --------------------------------------------------*;
*PN Sorted by variables for the final permanent dataset *;
*PN --------------------------------------------------*;
%let sortby = %str(studyid usubjid) ;
%let dsname = adsl;

*PN ---------------------*;
*PN local macro          *;
*PN ---------------------*;
%macro psort(ds=, where=, key=, dup=, out=);
  %if &out ^= %then %do;
    proc sort data=&ds &dup out=&out;
  %end;
  %else %do;
    proc sort data=&ds &dup out=&ds;
  %end;
    by &sortby &key;
    %if &where ^= %then %do;
      where &where.;
    %end;
  run;
%mend psort;
%macro VarExist(ds, var);
    %local rc dsid varexist;
    %global varexist_&var.;
    %let dsid = %sysfunc(open(&ds));

    %if %sysfunc(varnum(&dsid, &var)) > 0 %then %do;
        %let varexist_&var. = 1;

    %end;
    %else %do;
        %let varexist_&var. = 0;
        %put WARNING: Variable &var does not exist in &ds;
    %end;

    %let rc = %sysfunc(close(&dsid));

%mend VarExist;
options mlogic mprint;


*PN ---------------------*;
*PN Read in sdtm dm data *;
*PN ---------------------*;

%merge_supp(domain=dm, libname=sdtmdata);
%psort(ds=dm);

*PN----------------------------------------------------------PN*;
*PN Derive: TRTSEQP TRT01P TRT01A TRT02P TRT02A TRTSDT TRTSDTM PN*;
*PN----------------------------------------------------------PN*;


data dm1;
    attrib
        CNTRYGR1       length=$40      label='Pooled Country Group 1'
        REGIONC   length=$10      label='Geographic Region, CRF'
        PROTVS    length=$20      label='Protocol Version Subject Enrolled'
        AGEGR1                         length=$10      label='Pooled Age Group 1'
        AGEGR2                         length=$10      label='Pooled Age Group 2'
        AGEGR3                         length=$10      label='Pooled Age Group 3'
        TRTSEQP   length=$200     label='Planned Sequence of Treatments'
        TRT01P    length=$25      label='Planned Treatment for Period 01'
        TRT01PN   length=8        label='Planned Treatment for Period 01 (N)'
        TRT01A    length=$25      label='Actual Treatment for Period 01'
        TRT01AN   length=8        label='Actual Treatment for Period 01 (N)'
        TRTSDT    length=8        label='Date of First Exposure to Treatment'     format=date9.
        TRTSDTM   length=8        label='Datetime of First Exposure to Treatment' format=datetime19.
        RACEOTH   length=$200     label='Race Other Specify'
        RACEMULT  length=$200     label='Race Multiple'
        RACEGR1   length=$40      label='Pooled Race Group 1'
        PGICYN    length=$122     label='Consent to Pharmacogenetic Blood Sample?'
        TR01SDT   length=8        label='Date of First Exposure in Period 01'     format=date9.
        TR01STM   length=8        label='Time of First Exposure in Period 01'     format=time5.
        TR01SDTM  length=8        label='Datetime of First Exposure in Period 01' format=datetime19.
        CUTDT     length=8        label='Date of Data Cutoff Date'                format=date9.
        SAFFL     length=$1       label='Safety Population Flag'
                protvs    length=$20      label='Protocol Version Subject Enrolled';
      ;

*format trtsdt xvrdt TR01SDT CUTDT date9. TR01SDTM TRTSDTM datetime19. TR01STM time5.;
*length trt01p trt02p trt01a trt02a cntrygr1 regionc region1 agegr1 agegr2 agegr3 racemult  $200;

    set dm;
                randdtc = RFSTDTC;

    trtseqp = arm;

    if armcd = 'CABO+ATEZO' then trt01p ='Cabozantinib+Atezolizumab';
    else if armcd = 'ABI+PRED\ENZA' then trt01p ='Second NHT';

                if actarmcd = 'CABO+ATEZO' then trt01a ='Cabozantinib+Atezolizumab';
    else if actarmcd = 'ABI+PRED\ENZA' then trt01a ='Second NHT';

    if trt01p='Cabozantinib+Atezolizumab' then trt01pn = 0;
    else if trt01p='Second NHT' then trt01pn = 1;

    if trt01a='Cabozantinib+Atezolizumab' then trt01an = 0;
    else if trt01a='Second NHT' then trt01an = 1;

    if length(rfxstdtc) ge 10 then trtsdt = input(substr(rfxstdtc,1,10),yymmdd10.);
    if length(rfxstdtc) ge 16 then trtsdtm = input(rfxstdtc,e8601dt19.);

    tr01sdt = trtsdt;
    tr01sdtm = trtsdtm;
    if not missing(trtsdtm) then tr01stm = timepart(trtsdtm);

    if not missing(country) then do;
        if country in ('USA','CAN') then cntrygr1 = 'NORTH AMERICA (CANADA/USA)';
        else if country in ('HKG' 'PHL' 'THA' 'KOR' 'TWN' 'SGP' 'CHN' 'JPN') then cntrygr1 = 'ASIA';
        else if country in ('BEL' 'ROU' 'FRA' 'IRL' 'ITA' 'NLD' 'POL'
                         'ESP' 'TUR' 'GBR' 'DEU' 'AUS' 'NZL' 'AUT' 'CHE' 'CZE' 'HUN' 'GRC' 'UKR' 'PRT') then cntrygr1 =
'EUROPE/AUSTRALIA/NEW ZEALAND';
        else cntrygr1 = 'OTHER';

                if country in ('USA', 'CAN') then region1 = 'NORTH AMERICA';
                else if COUNTRY in ('BEL', 'ROU', 'FRA', 'IRL', 'ITA', 'NLD', 'POL', 'ESP', 'TUR', 'GBR', 'DEU', 'AUT','CHE', 'CZE', 'HUN','GRC',
'UKR', 'RUS', 'PRT','GEO', 'ISR') then region1 = 'EUROPE';
    else if COUNTRY in ('HKG', 'PHL', 'THA', 'KOR', 'TWN', 'SGP', 'CHN', 'JPN','AUS', 'NZL' ) then region1 = 'ASIA PACIFIC';
    else if COUNTRY in ('ARG', 'BRA', 'CHL', 'MEX')  then region1 = 'LATIN AMERICA';
    Else if country ne ' '  then region1= 'OTHER' ;
    end;

     if race7 = 'OTHER' then racemult= catx('; ', of race1-race7);
    /*if randpv1 = 'ORIGINAL' then protvs = '0';
    else if index(randpv1,'AMEND') then protvs = scan(randpv1,2);*/
    * protvs = strip(randpv1);

    if . < age < 65 then agegr1 = '<65';
    else if age >= 65 then agegr1 = '>=65';

    if . <  age  < 75 then agegr2 = '<75';
    else if age >= 75 then agegr2 = '>=75';

    if . < age < 65 then agegr3 = '<65';
    else if 65 <= age < 75 then agegr3 = '65 to <75';
    else if 75 <= age < 85 then agegr3 = '75 to <85';
    else if age >=85 then agegr3 = '>=85';

    *pgicyn = _pgicyn;
    cutdt = input("&cutdt.",yymmdd10.);

    if length(rfxendtc) ge 10 then exendtn = input(substr(rfxendtc,1,10),yymmdd10.);
    if exendtc ne . and exendtn > cutdt then rfxendtc = "&cutdt.";
  *   if rfxendtc > &cutdt. then rfxendtc = "&cutdt.";

    if not missing(trtsdt) then saffl = 'Y';
    else saffl = 'N';

    STRATAR = STRATC;
    STRATARN = STRAT;
    STRATAV = STRATC;
                if stratar ne ' ' then DSIXRS=substr(scan(scan(STRATAR,1,','),2,'='),1,1);
                if stratar ne ' ' then DOCIXRS=substr(scan(scan(STRATAR,2,','),2,'='),1,1);
    if stratar ne ' ' then FNHTIXRS=substr(scan(scan(STRATAR,3,','),2,'='),1,6);
                if STRATAV ="Liver metastasis disease=Yes,Prior docetaxel for mCSPC=Yes,Cancer type for first NHT given=mCRPC" then
STRATAVn =1;
if STRATAV ="Liver metastasis disease=Yes,Prior docetaxel for mCSPC=Yes,Cancer type for first NHT given=m0CRPC" then STRATAVn =2;
if STRATAV ="Liver metastasis disease=Yes,Prior docetaxel for mCSPC=Yes,Cancer type for first NHT given=mCSPC" then STRATAVn =3;
if STRATAV ="Liver metastasis disease=Yes,Prior docetaxel for mCSPC=No,Cancer type for first NHT given=mCRPC" then STRATAVn =4;
if STRATAV ="Liver metastasis disease=Yes,Prior docetaxel for mCSPC=No,Cancer type for first NHT given=m0CRPC" then STRATAVn =5;
if STRATAV ="Liver metastasis disease=Yes,Prior docetaxel for mCSPC=No,Cancer type for first NHT given=mCSPC" then STRATAVn =6;
if STRATAV ="Liver metastasis disease=No,Prior docetaxel for mCSPC=Yes,Cancer type for first NHT given=mCRPC" then STRATAVn =7;
if STRATAV ="Liver metastasis disease=No,Prior docetaxel for mCSPC=Yes,Cancer type for first NHT given=m0CRPC" then STRATAVn =8;
if STRATAV ="Liver metastasis disease=No,Prior docetaxel for mCSPC=Yes,Cancer type for first NHT given=mCSPC" then STRATAVn =9;
if STRATAV ="Liver metastasis disease=No,Prior docetaxel for mCSPC=No,Cancer type for first NHT given=mCRPC" then STRATAVn =10;
if STRATAV ="Liver metastasis disease=No,Prior docetaxel for mCSPC=No,Cancer type for first NHT given=m0CRPC" then STRATAVn =11;
if STRATAV ="Liver metastasis disease=No,Prior docetaxel for mCSPC=No,Cancer type for first NHT given=mCSPC" then STRATAVn =12;

    If ARMNRS='SCREEN FAILURE' then SCRNFFL = "Y";
    Else SCRNFFL = "";

    protvs = strip(dspv1)||' '||strip(dspv);

                if COUNTRY='UKR' then ukrsufl = 'Y';
     if COUNTRY ^ ='UKR' then ukrsufl = 'N';

    keep arm actarm studyid usubjid subjid siteid trtsdt country randdtc age: ageu sex stratav STRATAVn race: ethnic arm actarm trtsdt trtseqp trt:
tr01: trt01: trt02:
      trtsdt: pgicyn xvrdt rfxstdtc rfxendtc cutdt dthfl dthdtc saffl regionc protvs dspv: ukrsufl
      cntrygr1 STRATAR STRATARN REGIONI REGNIGR1 ETIOIXRS DSIXRS randcdtc ARMNRS region1 SCRNFFL DSIXRS DOCIXRS
FNHTIXRS  exendtn cutdt;
run;


data dm1;
     set dm1;
     if exendtn > cutdt then do;
      rfxendtc = "&cutdt.";
      end;
     run;
```

```
proc sort data=dm1 out=__dm nodupkey;
 by studyid usubjid;
run;
*PN ---------------------*;
*PN Read in sdtm cd data *;
*PN ---------------------*;
%merge_supp(domain=cd, libname=sdtmdata);
%psort(ds=cd);

data cdethbv cdethcv cdall(keep=studyid usubjid);
   set cd;
    if cdtestcd = 'CDETHBV' then output cdethbv;
    else if cdtestcd = 'CDETHCV' then output cdethcv;
                 else output cdall;
run;

proc sort data=cdall nodupkey;
 by studyid usubjid;
run;


data etiocrf;
    merge cdall(in=cdall)
          cdethbv(in=hbv keep=studyid usubjid cdorres rename=(cdorres=cdethbv))
          cdethcv(in=hcv keep=studyid usubjid cdorres rename=(cdorres=cdethcv))
    ;
    by studyid usubjid;
    attrib
        ETIOCRF           length=$200      label='Disease Etiology, CRF'
        ;
    if cdethbv = 'Y' then etiocrf = 'HBV [with or without HCV]';
    else if cdethcv = 'Y' and cdethbv eq 'N' then etiocrf = 'HCV [without HBV]';
    else etiocrf = 'Other';
    *if cdall and ^(hbv or hcv) then etiocrf='';
    keep studyid usubjid etiocrf;
run;

data CDEXMVI CDEXEHS;
    set sdtmdata.cd;
    if cdtestcd = 'CDEXMVI' then output CDEXMVI;
    else if cdtestcd = 'CDEXEHS' then output CDEXEHS;
run;

proc sort data=CDEXMVI;
 by studyid usubjid;
 run;

proc sort data=CDEXEHS;
 by studyid usubjid;
 run;

proc sort data = sdtmdata.tu out=tu (keep = studyid usubjid) nodupkey;
    where VISIT= "SCREENING" and TULOC="LIVER" and TUSTRESC ^= "NEW";
                    by studyid usubjid;
run;

/**

data dscrf;
    merge CDEXMVI(keep=studyid usubjid cdstresc rename=(cdstresc=CDEXMVI))
          CDEXEHS(keep=studyid usubjid cdstresc rename=(cdstresc=CDEXEHS))
    ;
    by studyid usubjid;
   attrib
      DSCRF              length=$200      label='Pres of Extrahep Dis or Macro Inv, CRF'
      ;
   if CDEXMVI = 'Y' or CDEXEHS = 'Y' then DSCRF = 'Yes';
   else if  CDEXMVI = 'N' and CDEXEHS = 'N' then DSCRF = 'No';
   else DSCRF = '';

   keep studyid usubjid DSCRF;
run;
***/

%merge_supp(domain=pr, libname=sdtmdata);
****PSURSDT***;

data _PSURSDT;
    length PRSTDTC_ PSURSDTF $10;
  set pr;
  where prCAT='SURGERY AND OTHER PROCEDURES HISTORY' and PRLOC='PROSTATE GLAND'  and PRSTDTC ne ' ' ; * and
       PRSTDTC =     APSETT="METASTATIC CASTRATION-SENSITIVE PROSTATE CANCER (MCSPC)";

            if length(PRSTDTC)=4   then do; PRSTDTC_=strip(PRSTDTC) || '-01-01'; PSURSDTF='M'; end;
        else if length(PRSTDTC)=7   then do; PRSTDTC_=strip(PRSTDTC) || '-01';    PSURSDTF='D'; end;
        else if length(PRSTDTC)>=10 then PRSTDTC_=strip(substr(PRSTDTC,1,10));

  if length (PRSTDTC_) = 10 then PSURSDT = input (PRSTDTC_, is8601da.);
  format PSURSDT date9.;
run;

proc sort data = _PSURSDT;
  by usubjid PRSTDTC_;
run;

proc freq data = _PSURSDT;
  table  PRSTDTC*PRSTDTC_*PSURSDT*PSURSDTF/list missing nocum nopct;
run;


data __PSURSDT (keep = usubjid PSURSDT PSURSDTF);
  set _PSURSDT;
     by usubjid PRSTDTC_;
            if last.usubjid;
run;


****RHTXSDT***;

data _RHTXSDT;
    length PRENDTC_ RHTXSDTF $10;
  set pr;
  where prCAT="HISTORY OF RADIATION THERAPY"  and PRENDTC ne ' ' ; * and
       PRENDTC =     APSETT="METASTATIC CASTRATION-SENSITIVE PROSTATE CANCER (MCSPC)";

            if length(PRENDTC)=4   then do; PRENDTC_=strip(PRENDTC) || '-01-01'; RHTXSDTF='M'; end;
        else if length(PRENDTC)=7   then do; PRENDTC_=strip(PRENDTC) || '-01';    RHTXSDTF='D'; end;
        else if length(PRENDTC)>=10 then PRENDTC_=strip(substr(PRENDTC,1,10));

  if length (PRENDTC_) = 10 then RHTXSDT = input (PRENDTC_, is8601da.);
  format RHTXSDT date9.;
run;

proc sort data = _RHTXSDT;
  by usubjid PRENDTC_;
run;

proc freq data = _RHTXSDT;
```

```
  table  PRENDTC*PRENDTC_*RHTXSDT*RHTXSDTF/list missing nocum nopct;
run;


data __RHTXSDT (keep = usubjid RHTXSDT RHTXSDTF);
  set _RHTXSDT;
    by usubjid PRENDTC_;
            if last.usubjid;
run;



%merge_supp(domain=cm, libname=sdtmdata);

******PNRTMCFL***;
data _PNRTMC;
  set cm;
  where CMCAT="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY"  and APSETT eq  'MCRPC' and cmtrt ne ' ' ;
   PNRTMCFL = 'Y';
run;

proc sort data = _PNRTMC out = __PNRTMC   (keep = usubjid PNRTMCFL) nodupkey;
  by usubjid;
run;

******NPNRTXMC   NPRADGR1***;

data _NPNR;
  set cm;
  where CMCAT='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY'  and APSETT eq  'MCRPC' and cmtrt ne ' ' ;
run;


proc sort data = _NPNR out = _NPNRn nodupkey;
  by usubjid cmtrt cmstdtc;
run;

proc freq data = _NPNRn noprint;
  table usubjid/out= onpnrn;
run;



data __NPNRn;
   length NPNRMGR1 $20;
   set oNPNRn;
              by usubjid;
            NPNRTXMC = count;
   NPNRMGR1 = put (count,1.);
            if count > = 3 then NPNRMGR1 = '>=3';
run;


****RDPRSDT***;

data _RDPRSDT;
    length APRRDTC_ RDPRSDTF $10;
  set cm;
  where CMCAT="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY" and cmdecod in ("ENZALUTAMIDE" "ABIRATERONE"
"APALUTAMIDE" "DAROLUTAMIDE")  and APRRDTC ne ' '; * and
      APRRDTC =     APSETT="METASTATIC CASTRATION-SENSITIVE PROSTATE CANCER (MCSPC)";

           if length(APRRDTC)=4   then do; APRRDTC_=strip(APRRDTC) || '-01-01'; RDPRSDTF='M'; end;
       else if length(APRRDTC)=7   then do; APRRDTC_=strip(APRRDTC) || '-01';    RDPRSDTF='D'; end;
       else if length(APRRDTC)>=10 then APRRDTC_=strip(substr(APRRDTC,1,10));
  if length (APRRDTC_) = 10 then RDPRSDT = input (APRRDTC_, is8601da.);
  format RDPRSDT date9.;
run;

proc sort data = _RDPRSDT;
  by usubjid APRRDTC_;
run;

proc freq data = _RDPRSDT;
  table  APRRDTC*APRRDTC_*RDPRSDT*RDPRSDTF/list missing nocum nopct;
run;


data __RDPRSDT (keep = usubjid RDPRSDT RDPRSDTF);
  set _RDPRSDT;
    by usubjid APRRDTC_;
            if last.usubjid;
run;


**** psadsdt ***;
data _psadsdt;
   length APRBDTC_ PSADSDTF $10;
  set cm;
  where CMCAT="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY"  and  cmdecod in ("ENZALUTAMIDE" "ABIRATERONE"
"APALUTAMIDE" "DAROLUTAMIDE") and APRBDTC ne ' '; * and
      APRBDTC =     APSETT="METASTATIC CASTRATION-SENSITIVE PROSTATE CANCER (MCSPC)";

           if length(APRBDTC)=4   then do; APRBDTC_=strip(APRBDTC) || '-01-01'; PSADSDTF='M'; end;
       else if length(APRBDTC)=7   then do; APRBDTC_=strip(APRBDTC) || '-01';    PSADSDTF='D'; end;
       else if length(APRBDTC)>=10 then APRBDTC_=strip(substr(APRBDTC,1,10));

  if length (APRBDTC_) = 10 then psadsdt = input (APRBDTC_, is8601da.);
  format psadsdt date9.;
run;

proc sort data = _psadsdt;
  by usubjid APRBDTC_;
run;

proc freq data = _psadsdt;
  table  APRBDTC*APRBDTC_*psadsdt/list missing nocum nopct;
run;


data __psadsdt (keep = usubjid psadsdt PSADSDTF);
  set _psadsdt;
    by usubjid APRBDTC_;
            if last.usubjid;
run;


data _cm;
  set cm;
  where CMCAT="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY" and CMTRT="DOCETAXEL"  and
APSETT="METASTATIC CASTRATION-SENSITIVE PROSTATE CANCER (MCSPC)";
run;


data _nht ;
   attrib fnhtcrf length=$20 label='First NHT given for, CRF';
  set cm;
  if CMCAT="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY"  and CMtrt in ("ENZALUTAMIDE", "ABIRATERONE",
"APALUTAMIDE", "DAROLUTAMIDE")
```

```
     and upcase(APSETT)="MCRPC" then  fnhtcrf="mCRPC";

   if CMCAT="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY"  and CMtrt in ("ENZALUTAMIDE", "ABIRATERONE",
"APALUTAMIDE", "DAROLUTAMIDE")
     and upcase(APSETT)="M0 CRPC"  then  fnhtcrf="m0CRPC";

   if CMCAT="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY"  and CMtrt in ("ENZALUTAMIDE", "ABIRATERONE",
"APALUTAMIDE", "DAROLUTAMIDE")
     and upcase(APSETT)="METASTATIC CASTRATION-SENSITIVE PROSTATE CANCER (MCSPC)" then  fnhtcrf="mCSPC";
            if fnhtcrf ne ' ';
   keep usubjid studyid CMSTDTC CMCAT CMtrt APSETT fnhtcrf;
run;

proc sort data = _nht;
  by studyid usubjid  CMSTDTC;
run;

data _nht;
  set _nht;
    by studyid usubjid  CMSTDTC;
            if first.usubjid;
run;


proc sort data =_cm out=__cm (keep=studyid usubjid);
   by studyid usubjid;
run;


proc sort data=__dm out=regionc(keep=studyid usubjid regionc randdtc);
by studyid usubjid;
run;

data __crf;
   attrib DSCRF length=$20 label='Pres of Liver metastasis, CRF'
        DOCCRF length=$20 label='Prior Docetaxel for mCSPC, CRF';
            merge __dm (in=a) etiocrf tu (in=c) __cm (in=d) _nht;
    by studyid usubjid;
                if c then DSCRF = 'Yes';
                        if d then DOCCRF = 'Yes';
                        if ^ d then DOCCRF = ' ';
run;

data _trtnht;
   length trtnht $20;
   set sdtmdata.ex;
      if EXTRT in ("ABIRATERONE", "PREDNISONE" ) then trtnht= 'Abi+Pred';
      if EXTRT in ("ENZALUTAMIDE" ) then trtnht= 'Enza';
  if trtnht ne ' ';
run;



proc sort data = _trtnht out= __trtnht  (keep = studyid usubjid trtnht) ; * nodupkey;
  by usubjid trtnht;
run;


data __trtnht;
  set __trtnht;
    by usubjid trtnht;
            if first.usubjid;
run;
```

```
*PN --------------------------------*;
*PN Read in SDTMDATA.DS              *;
*PN Derive: RFICDT, RANDDT, RFIC2DT *;
*PN --------------------------------*;

data __rficdt(keep=studyid usubjid rficdt) __randdt(keep=studyid usubjid randdt ittfl);
   set sdtmdata.ds;
   attrib
      RFICDT        length=8     label='Date of Informed Consent'    format=date9.
      RANDDT    length=8     label='Date of Randomization'      format=date9.
      RFIC2DT       length=8     label='Date of Informed Consent 2'  format=date9.
      ITTFL         length=$1    label='Intent-To-Treat Population Flag'
      ;

   if dsdecod = 'INFORMED CONSENT OBTAINED' then do;
      if length(dsstdtc) ge 10 then rficdt = input(substr(dsstdtc,1,10),yymmdd10.);
      output __rficdt;
    end;
   else if dsdecod = 'RANDOMIZED' then do;
      if length(dsstdtc) ge 10 then randdt = input(substr(dsstdtc,1,10),yymmdd10.);
      if not missing(randdt) then ittfl = 'Y';
      else ittfl = 'N';
      output __randdt;
    end;
run;


*PN --------------------------------*;
*PN Read in SDTMDATA.IE              *;
*PN Derive: ELIGIBFL               *;
*PN --------------------------------*;

proc sort data = sdtmdata.ie out=ie;
  by studyid usubjid;
run;

data __eligibfl;
   set ie;
   by studyid usubjid;
   attrib
      ELIGIBFL      length=$1     label='Met All Eligibility Criteria Flag'
      ;
   if first.usubjid;
   eligibfl = 'N';
   keep studyid usubjid eligibfl;
run;

proc sort data=sdtmdata.ds out=rficdt(keep=studyid usubjid dsstdtc);
   by studyid usubjid;
   where dsdecod = 'INFORMED CONSENT OBTAINED';
run;


*PN ------------------------------------------*;
*PN Read in SDTMDATA.DS                        *;
*PN Derive: Disposition Dates                 *;
*PN EOADCDT EOCDCDT EOSDCDT EOXADCDT EOXCDCDT *;
*PN ------------------------------------------*;

*PN ---------------------*;
*PN Read in sdtm ds data *;
*PN ---------------------*;
%merge_supp(domain=ds, libname=sdtmdata);
```

```
%psort(ds=ds);


proc sort data = ds out= lfds;
   by usubjid dsstdtc ;
    where DSTERM='LOST TO FOLLOW-UP' and DSSCAT in ('END OF SURVIVAL FOLLOW-UP','END OF RADIOGRAPHIC FOLLOW-
UP');
run;

data __lfds (keep =usubjid LTFUDT);
  set lfds;
      by usubjid dsstdtc ;
                if first.usubjid;
                if length (dsstdtc) = 10 then LTFUDT = input (dsstdtc, is8601da.);
run;


data __eoadcdt(keep=studyid usubjid eoadcdt  ) __eocdcdt(keep=studyid usubjid eocdcdt ) __eoedcdt(keep=studyid usubjid eoedcdt  )
    __eobdcdt(keep=studyid usubjid eobdcdt  ) __eopdcdt(keep=studyid usubjid eopdcdt )
    __eosdt  (keep = studyid usubjid eosrdtc eosdt eosdtf )
    ;
   set ds(keep = studyid usubjid dsscat dsstdtc );

   attrib
         EOADCDT     length=8        label='Date of Decision to Discont. Atez'      format=date9.
         EOCDCDT   length=8        label='Date of Decision to Discont. Cabo'      format=date9.
         EOEDCDT   length=8        label='Date of Decision to Discont. Enza'      format=date9.
         EOBDCDT  length=8        label='Date of Dec to Discont. Abir'       format=date9.
         EOPDCDT  length=8        label='Date of Dec to Discont. Pred'       format=date9.
         EOSDT     length=8        label='End of Survival FUP Date'               format=date9.
         EOSDTF    length=$1       label='End of Study Date Imputation Flag'
         ;

  if length(dsstdtc) ge 10 then dsstdtn = input(substr(dsstdtc,1,10),yymmdd10.);
  if length(dsstdtc) =7  then  do;
     /* if length(dsstdtc) =10 then  */ *  dsstdtn = input(substr((strip(dsstdtc)||'-1'),1,10),yymmdd10.);
      dsstdtn = input((strip(dsstdtc)||'-1'),yymmdd10.);
     * if length(dsstdtc) =7  then *;
               eosdtf = 'D';
  end;
  if dsscat = 'END OF STUDY TREATMENT, ATEZOLIZUMAB' then do;
    eoadcdt = dsstdtn;
    output __eoadcdt;
   end;
  if dsscat = 'END OF STUDY TREATMENT, CABOZANTINIB' then do;
    eocdcdt = dsstdtn;
    output __eocdcdt;
   end;
  if dsscat = 'END OF STUDY TREATMENT, ENZALUTAMIDE' then do;
    eoedcdt = dsstdtn;
    output __eoedcdt;
   end;
  if dsscat = 'END OF STUDY TREATMENT, ABIRATERONE' then do;
    eobdcdt = dsstdtn;
    output __eobdcdt;
   end;
  if dsscat = 'END OF STUDY TREATMENT, PREDNISONE' then do;
    eopdcdt = dsstdtn;
    output __eopdcdt;
   end;
  if dsscat = 'END OF SURVIVAL FOLLOW-UP' then do;
    eosdt = dsstdtn;
    eosrdtc=dsstdtc;
    if eosdt>input("&cutdt.",yymmdd10.) then eosrdtc="&cutdt.";
    if length(dsstdtc) =7 then  eosdtf = 'D';
   * end;
    output __eosdt;
   end;
 run;


 *PN -----------------------------------------*;
 *PN Read in __eoadcdt __eocdcdt __eosdcdt     *;
 *PN Derive: Disposition Dates                *;
 *PN EOTDCDT                                  *;
 *PN -----------------------------------------*;


 data __eotdcdt;
   merge __eoadcdt __eocdcdt  __eoedcdt __eopdcdt __eobdcdt __dm(keep=studyid usubjid trt01an);
   by studyid usubjid;

   attrib
         EOTDCDT     length=8      label='Date of Decision to Discont. Core Tx'     format=date9.
         ;

   if nmiss(eoadcdt, eocdcdt, eoedcdt, eobdcdt, eopdcdt) lt 5  then eotdcdt = max(eoadcdt, eocdcdt, eopdcdt, eobdcdt,  eoedcdt);
   if trt01an = 0 and nmiss(eoadcdt,eocdcdt) gt 0 then eotdcdt = .;
   /**
   if trt01an = 0 and nomiss(eoadcdt,eocdcdt) gt 0 then eotdcdt = .;
   else if trt01an = 1 and  eoedcdt ne   .  then eotdcdt = eoedcdt;
   else if trt01an = 1 and  nmiss( eopdcdt, eobdcdt) gt 0  then eotdcdt = .;
 ***/
   keep studyid usubjid  trt01an  eotdcdt eoadcdt eocdcdt eopdcdt eobdcdt eoedcdt;
 run;



 *PN -----------------------------------------*;
 *PN Read in sdtmdata.ds                      *;
 *PN Derive: Date of Withdrawal of Full Consent *;
 *PN WDCFDT DSEVTADT DSEVTCDT DSEVTSDT DSEVXADT DSEVXCDT *;
 *PN -----------------------------------------*;

 data __wdcfdt(keep=studyid usubjid wdcfdt)
    __dsevtadt(keep=studyid usubjid dsevtadt) __dsevtcdt(keep=studyid usubjid dsevtcdt)
    __dsevtbdt(keep=studyid usubjid dsevtbdt) __dsevtpdt(keep=studyid usubjid dsevtpdt)
    __dsevtedt(keep=studyid usubjid dsevtedt)
    ;
   set ds(keep = studyid usubjid dsterm dsscat dsstdtc etevdtc);

   attrib
         WDCFDT       length=8      label='Date of Withdrawal of Full Consent'     format=date9.
         DSEVTADT  length=8      label='Date of Disposition Event (Atez)' format=date9.
         DSEVTCDT  length=8      label='Date of Disposition Event (Cabo)' format=date9.
         DSEVTBDT  length=8      label='Date of Disposition Event (Arbi)' format=date9.
         DSEVXPDT  length=8      label='Date of Disposition Event (Pred)'  format=date9.
         DSEVXEDT  length=8      label='Date of Disposition Event (Enza)'  format=date9.
         ;
   if length(dsstdtc) ge 10 then dsstdtn = input(substr(dsstdtc,1,10),yymmdd10.);
   if length(etevdtc) ge 10 then etevdtn = input(substr(etevdtc,1,10),yymmdd10.);

   if index(strip(upcase(dsterm)), 'WITHDREW FULL CONSENT') and dsscat='RE-CONSENT OR CONSENT WITHDRAWAL' then do;
      wdcfdt = dsstdtn;
      output __wdcfdt;
    end;
   if dsscat='END OF STUDY TREATMENT, ATEZOLIZUMAB' and dsterm in ('CLINICAL DETERIORATION' 'PROTOCOL DEVIATION')
    then do;
      dsevtadt = etevdtn;
      output __dsevtadt;
```

```
       end;
    if dsscat='END OF STUDY TREATMENT, CABOZANTINIB' and dsterm in ('CLINICAL DETERIORATION' 'PROTOCOL DEVIATION')
      then do;
        dsevtcdt = etevdtn;
        output __dsevtcdt;
       end;
    if dsscat='END OF STUDY TREATMENT, ABIRATERONE' and dsterm in ('CLINICAL DETERIORATION' 'PROTOCOL DEVIATION')
      then do;
        dsevtbdt = etevdtn;
        output __dsevtbdt;
       end;
    if dsscat='END OF STUDY TREATMENT, PREDNISONE' and dsterm in ('CLINICAL DETERIORATION' 'PROTOCOL DEVIATION')
      then do;
        dsevtpdt = etevdtn;
        output __dsevtpdt;
       end;
    if dsscat='END OF STUDY TREATMENT, ENZALUTAMIDE' and dsterm in ('CLINICAL DETERIORATION' 'PROTOCOL DEVIATION')
      then do;
        dsevtedt = etevdtn;
        output __dsevtedt;
       end;
run;


proc sort data=__wdcfdt ;
                  by  studyid usubjid wdcfdt;
run;

data __wdcfdt;
    set __wdcfdt;
    by studyid usubjid wdcfdt;
    if first.usubjid;
run;

*PN ------------------------------------------*;
*PN Read in SDTMDATA.EX                       *;
*PN Derive: First Exposure variables          *;
*PN TRTASDT TRTCSDT TRTSSDT TRTAXSDT TRTCXSDT *;
*PN ------------------------------------------*;

data ex_a ex_c ex_e ex_b ex_p;
    merge sdtmdata.ex __dm(keep=studyid usubjid );
    by studyid usubjid;
    if length(exstdtc) ge 10 then exstdtn = input(substr(exstdtc,1,10),yymmdd10.);

    if extrt='ATEZOLIZUMAB' & exdose > 0 then output ex_a;
    if extrt='CABOZANTINIB' & exdose > 0 then output ex_c;
    if extrt='ENZALUTAMIDE' & exdose > 0  then output ex_e;
    if extrt='PREDNISONE'  & exdose > 0  then output ex_p;
    if extrt='ABIRATERONE'  & exdose > 0  then output ex_b;
                  keep studyid usubjid exstdtn exdose;
run;

proc sort data=ex_a;
    by studyid usubjid exstdtn;
    where not missing(exstdtn) ;
run;

proc sort data=ex_c;
    by studyid usubjid exstdtn;
    where not missing(exstdtn);
run;

proc sort data=ex_e;
    by studyid usubjid exstdtn;
```

```
   where not missing(exstdtn)  ;
run;

proc sort data=ex_p;
   by studyid usubjid exstdtn;
   where not missing(exstdtn) ;
run;

proc sort data=ex_b;
   by studyid usubjid exstdtn;
   where not missing(exstdtn) ;
run;

data __trtasdt;
   set ex_a;
   by studyid usubjid exstdtn;
   if first.usubjid;
   attrib  TRTASDT    length=8       label='Date of First Exposure to Atezo, Core' format=date9.;
   trtasdt = exstdtn;
   drop exstdtn;
run;

data __trtcsdt;
   set ex_c;
   by studyid usubjid exstdtn;
   if first.usubjid;
   attrib  TRTCSDT    length=8       label='Date of First Exposure to Cabo, Core' format=date9.;
   trtcsdt = exstdtn;
   drop exstdtn;
run;

data __trtesdt;
   set ex_e;
   by studyid usubjid exstdtn;
   if first.usubjid;
   attrib  TRTESDT    length=8       label='Date of First Exposure to Enzo' format=date9.;
   trtesdt = exstdtn;
   drop exstdtn;
run;

data __trtpsdt;
   set ex_p;
   by studyid usubjid exstdtn;
   if first.usubjid;
   attrib  TRPSDT     length=8       label='Date of First Exposure to Pred' format=date9.;
   trtpsdt = exstdtn;
   drop exstdtn;
run;

data __trtbsdt;
   set ex_b;
   by studyid usubjid exstdtn;
   if first.usubjid;
   attrib  TRTBSDT    length=8       label='Date of First Exposure to Arbi' format=date9.;
   trtbsdt = exstdtn;
   drop exstdtn;
run;


*PN ------------------------------------------*;
*PN Read in SDTMDATA.EX                        *;
*PN Derive: Last Exposure variables           *;
*PN ------------------------------------------*;
```

```
/**DTC2DT Macro**/
%macro dtc2dt(dtcvar , prefix=a,  );

    if length(trim(&dtcvar))=10 and index(&dtcvar,'--')=0 then
      &prefix.dt = input(&dtcvar, yymmdd10.);
    else if length(&dtcvar)=16 and index(&dtcvar,'--')=0 and index(&dtcvar,'-:')=0 then
      do;
        &prefix.dtm = input(trim(&dtcvar)||":00", e8601dt19.);
        &prefix.dt  = datepart(&prefix.dtm);
                                  &prefix.tm  = timepart(&prefix.dtm);
        * optionally add formats: ;
         format &prefix.dtm datetime19. &prefix.dt date9. &prefix.tm time5.;
      end;
%mend dtc2dt;


data ex ;
    set sdtmdata.ex ;
    by studyid usubjid;

    if length(exendtc)=4 then exendtc=";/*Conditional Statement-Remove after data issues are fixed*/
if length(exstdtc)>=10 then do;
exstdtcx=strip(substr(exstdtc,1,10));
exstdd=substr(exstdtcx,9,2);
exstmm=substr(exstdtcx,6,2);
exstyy=substr(exstdtcx,1,4);
end;
if length(exstdtc)<10 then do  ;
exstdd=substr(exstdtc,9,2);
exstmm=substr(exstdtc,6,2);
exstyy=substr(exstdtc,1,4);
end;

if length(exendtc)>=10 then do;
exendtcx=strip(substr(exendtc,1,10));
exendd=substr(exendtcx,9,2);
exenmm=substr(exendtcx,6,2);
exenyy=substr(exendtcx,1,4);
end;
if length(exendtc)<10 then do  ;
exendd=substr(exendtc,9,2);
exenmm=substr(exendtc,6,2);
exenyy=substr(exendtc,1,4);
end;


if exendtc^=" then do;
if exendd=" then do;
exendtcx=strip(exenyy)||"-"||strip(put(input(exenmm, best.),z2.));
if input(exenmm,best.)=2 then do;
if intck("day", mdy(2,1,input(exenyy,best.)),mdy(3,1,input(exenyy,best.)))=28
then exendtc_imp=catx('-',exenyy,"02","28");
else exendtc_imp=catx('-',exenyy,"02","29");
end;
else if input(exenmm,best.) in (4,6,9,11) then
exendtc_imp=catx('-',exenyy,put(input(exenmm, best.),z2.),"30");
else exendtc_imp=catx('-',exenyy,put(input(exenmm, best.),z2.),"31");
end;
else do;
exendtcx=catx('-',exenyy,put(input(exenmm, best.),z2.),put(input(exendd, best.),z2.));
exendtc_imp=exendtcx;
end;
end;
if exstdtc^=" then do;
if exstdd='' then do;
exstdtc=strip(exstyy)||"-"||strip(put(input(exstmm, best.),z2.));
if input(exstmm,best.)=2 then do;
if intck("day", mdy(2,1,input(exstyy,best.)),mdy(3,1,input(exstyy,best.)))=28
then exstdtc_imp=catx('-',exstyy,"02","28");
else exstdtc_imp=catx('-',exstyy,"02","29");
end;
else if input(exstmm,best.) in (4,6,9,11) then
exstdtc_imp=catx('-',exstyy,put(input(exstmm, best.),z2.),"30");
else exstdtc_imp=catx('-',exstyy,put(input(exstmm, best.),z2.),"31");
end;
else do;
exstdtcx=catx('-',exstyy,put(input(exstmm, best.),z2.),put(input(exstdd, best.),z2.));
exstdtc_imp=exstdtcx;
end;
end;
    if length(exstdtc) ge 10 then exstdtn = input(substr(exstdtc,1,10),yymmdd10.);
    %x_dtimpute(indtc=exendtc, outdt=exendtn, outdtf=EXENDTF, impalign=e);
    if length(exendtc) ge 10 then exendtn = input(substr(exendtc,1,10),yymmdd10.);
    %dtc2dt(exstdtc , prefix=exst  );
    %dtc2dt(exendtc , prefix=exen  );

    if nmiss(exstdtn, exendtn) lt 2 then lastdtn = max(exstdtn, exendtn);
    if nmiss(exstdtm, exendtm) lt 2 then lastdtm = max(exstdtm, exendtm);
if nmiss(exsttm, exentm) lt 2 then lasttm = max(exsttm, exentm);
format lastdtm /* trtedtm */ datetime19. lasttm trtetm time5. lastdtn date9.;

run;

proc print data =ex;
     title 'ex 184315-1116-3061';
    where usubjid = '184315-1116-3061';
run;



proc sort data=ex;
                   by studyid usubjid;
run;

data exl_a exl_c exl_e exl_p  exl_b;
    merge ex __dm(keep=studyid usubjid);
    by studyid usubjid;

    keep studyid usubjid exstdtn exendtn EXENDTF lastdtn lastdtm lasttm  exstdtc;
    if extrt='ATEZOLIZUMAB' and (exdose gt .) and (exdose ^ = 0)  then output exl_a;
    if extrt='CABOZANTINIB' and (exdose gt .)  and (exdose ^ = 0) then output exl_c;
    if  extrt='ENZALUTAMIDE' and (exdose gt .)  and (exdose ^ = 0) then output exl_e;
    if extrt='PREDNISONE'  and (exdose gt .)  and (exdose ^ = 0) then output exl_p;
    if extrt='ABIRATERONE' and (exdose gt .)  and (exdose ^ = 0) then output exl_b;
run;

proc sort data=exl_a;
    by studyid usubjid lastdtn;
run;
data __trtaedt;
    set exl_a;
    by studyid usubjid lastdtn;
    if last.usubjid;
    attrib  TRTAEDT     length=8        label='Date of Last Exposure to Atezo, Core' format=date9.;
    trtaedt = lastdtn;
    trtaedtx = lastdtn;
    TRTAEDTF=strip(EXENDTF);
    * drop lastdtn EXENDTF;
run;
```

```
proc print data = exl_a;
     title '184315-1636-3650';
   where usubjid = '184315-1636-3650';
run;


data __trtaedt;
merge __dm(keep=studyid usubjid trt01p cutdt rfxstdtc) __trtaedt __eoadcdt ;
by studyid usubjid ;

if ^missing(rfxstdtc) then do;
if missing(exendtn) & ^missing(eoadcdt) then do;
TRTAEDT = min((eoadcdt-1),cutdt);
TRTAEDTF='Y';
end;
end;


if missing(eoadcdt) then do;
                  TRTAEDT=.;
  TRTAEDTF='';
end;

run;

proc sort data=exl_p;
    by studyid usubjid lastdtn;
run;

data __trtpedt;
    set exl_p;
    by studyid usubjid lastdtn;
    if last.usubjid;
    attrib  TRTpEDT     length=8        label='Date of Last Exposure to Pred, Core' format=date9.;
    trtpedt = lastdtn;
    trtpedtx = lastdtn;
    TRTpEDTF=strip(EXENDTF);
    * drop lastdtn EXENDTF;
run;


data __trtpedt;
merge __dm(keep=studyid usubjid trt01p cutdt rfxstdtc) __trtpedt __eopdcdt ;
by studyid usubjid ;

if ^missing(rfxstdtc) then do;
if missing(exendtn) & ^missing(eopdcdt) then do;
TRTpEDT = min((eopdcdt-1),cutdt);
TRTpEDTF='Y';
end;
end;

if  4 < = length(exendtc) < 10 and  eopdcdt ne .  then trtpedt = min((eopdcdt-1),cutdt);;

if missing(eopdcdt) then do;
                  TRTpEDT=.;
  TRTpeDTF='';
end;

run;



proc sort data=exl_c;
```

```
   by studyid usubjid lastdtn;
run;
data __trtcedt;
   set exl_c;
   by studyid usubjid lastdtn;
   if last.usubjid;
   attrib  TRTCEDT    length=8        label='Date of Last Exposure to Cabo, Core' format=date9.;
   trtcedt = lastdtn;
   TRTCEDTF=strip(EXENDTF);
              * if usubjid ='184315-4702-3525' then trtcedt = '05jun2022'd;   ***ERROR this code need to be removed ***;
   * drop lastdtn EXENDTF lastdtm lasttm;
run;

proc print data=exl_c;
   title 'exl_c';
  where usubjid = '184315-4702-3525';
run;

proc print data=__trtcedt;
   title '__trtcedt';
  where usubjid = '184315-4702-3525';
run;

data __trtcedt;
merge __dm(keep=studyid usubjid trt01p cutdt rfxstdtc dthdtc) __trtcedt __eocdcdt ;
by studyid usubjid ;
if ^missing(rfxstdtc) then do;
if ^ missing(exendtn) and exendtn < cutdt then trtcedt = exendtn;
if missing(exendtn) & ^missing(eocdcdt) then do;
trtcedt = min((eocdcdt-1),cutdt);
* TRTCEDTF='Y';
end;
if ^ missing(exendtn) & ^missing(eocdcdt)  and exendtn >  eocdcdt then do;
trtcedt = min((eocdcdt-1),cutdt);
* TRTCEDTF='Y';
end;
end;
if   4 < = length(exendtc) < 10 and  eocdcdt ne .  then trtcedt = min((eocdcdt-1),cutdt);
dthdtn = input(substr(dthdtc,1,10),yymmdd10.);
* if 4 < = length(exendtc) < 10  and trtcedt > dthdtn then trtcedt =dthdtn -1;
if missing(eocdcdt) then do;
              TRTCEDT = .;
   TRTCEDTF = '';
end;
run;


proc sort data=exl_e;
   by studyid usubjid lastdtn;
run;
data __trteedt;
   set exl_e;
   by studyid usubjid lastdtn;
   if last.usubjid;
   attrib  TRTeEDT     length=8        label='Date of Last Exposure to Enza, Core' format=date9.;
   trteedt = lastdtn;

   TRTeEDTF=strip(EXENDTF);
   drop lastdtn EXENDTF lastdtm lasttm;
run;

data __trteedt;
merge __dm(keep=studyid usubjid trt01p rfxstdtc cutdt) __trteedt __eoedcdt ;
by studyid usubjid ;
```

```
if ^missing(rfxstdtc) then do;
if missing(exendtn) & ^missing(eoedcdt) then do;
trteedt = min((eoedcdt-1),cutdt);
TRTSEDTF='Y';
end;
end;

if  4 < = length(exendtc) < 10 and  eoedcdt ne .  then trteedt = min((eoedcdt-1),cutdt);;

if missing(eoedcdt) then do;
              TRTeEDT=.;
              TRTeEDTF='';
end;
run;

/**
proc sort data=exl_p;
   by studyid usubjid lastdtn;
run;
data __trtpedt;
   set exl_p;
   by studyid usubjid lastdtn;
   if last.usubjid;
   attrib  TRTPEDT     length=8        label='Date of Last Exposure to Pred, XVR' format=date9.;
   trtpedt = lastdtn;
   drop lastdtn;
run;
**/

proc sort data=exl_b;
   by studyid usubjid lastdtn;
run;
data __trtbedt;
   set exl_b;
   by studyid usubjid lastdtn;
   if last.usubjid;
   attrib  TRTbEDT     length=8        label='Date of Last Exposure to Arbi, XVR' format=date9.;
   trtbedt = lastdtn;
   * drop lastdtn lastdtm lasttm;
run;


data __trtbedt;
merge __dm(keep=studyid usubjid trt01p rfxstdtc cutdt) __trtbedt __eobdcdt ;
by studyid usubjid ;

if ^missing(rfxstdtc) then do;
if missing(exendtn) & ^missing(eobdcdt) then do;
trtbedt = min((eobdcdt-1),cutdt);
TRTbEDTF='Y';
end;
end;

if  4 < = length(exendtc) < 10 and  eobdcdt ne .  then trtbedt = min((eobdcdt-1),cutdt);;

if missing(eobdcdt) then do;
              TRTbEDT=.;
              TRTbEDTF='';
end;
run;

*PN ---------------------------------------------*;
*PN Derive TRTEDT: Max of TRTEEDT,TRTAEDT,TRTCEDT*;
*PN ---------------------------------------------*;
```

```
proc sort data = exl_a out=exla nodupkey;
   by studyid usubjid;
run;

proc sort data = exl_p out=exlp nodupkey;
   by studyid usubjid;
run;

proc sort data = exl_e out=exle nodupkey;
   by studyid usubjid;
run;


data bpe;
    merge exla (in=a keep=studyid usubjid ) exlp (in=p keep=studyid usubjid ) exle (in=e keep=studyid usubjid );
                 by studyid usubjid;
               if a then exla = 1;
               if p then exlp = 1;
               if e then exle = 1;
run;

/**

data __dm;
  length trtnht $20;
  merge __dm (in=a) bpe;
       by studyid usubjid;
  if exla = 1 or exlp = 1 then TRTNHT = 'Abi+Pred';
  if exle = 1 then TRTNHT = 'Enza';
run;
**/

data __TRTEDT; * (keep=studyid usubjid trtedt TRTEDTF lastdtm lasttm);
               attrib
               TRTEDT   length=8      label='Date of Last Exposure to Treatment'     format=date9.;
merge __dm(keep=studyid usubjid trt01p ) __trtcedt  __trtaedt  __trteedt __trtbedt __trtpedt  __trteedt __eoadcdt  __eocdcdt __eobdcdt
__eopdcdt  __eoedcdt __trtnht;
by studyid usubjid ;
if nmiss(TRTeEDT,TRTAEDT,TRTCEDT,trtpedt,trtbedt) lt 5 then trtedt=max(of TRTeEDT,TRTAEDT,TRTCEDT,trtpedt,trtbedt);

if trt01p ="Cabozantinib+Atezolizumab" then do;
if nmiss(eoadcdt,eocdcdt) >=1     then do;
trtedt=.;
* lastdtm=.;
* lasttm=.;
end;
end;

if trt01p ="Second NHT"  and TRTNHT="Abi+Pred" then do;
if nmiss(EOBDCDT, EOPDCDT) = 2          then do;
trtedt=.;
* lastdtm=.;
* lasttm=.;
end;
end;

if trt01p ="Second NHT"  and TRTNHT="Enza" then do;
if nmiss( EOEDCDT) >=1          then do;
trtedt=.;
lastdtm=.;
lasttm=.;
end;
end;
```

```
if nmiss(lastdtm,trtedt) =0  & datepart(lastdtm)^=trtedt then do;
lastdtm=.;
lasttm=.;
end;


if trtedt ne . then trtetm = lasttm;
if trtedt ne . and  lasttm  ne . then trtedtm = lastdtm;
* else if trtedt ne . and  lasttm  eq . then TRTEDTM = trtedt;
TRTEDTM = trtedt;
if ^missing(TRTAEDTF) then TRTEDTF=TRTAEDTF;
else if ^missing(TRTCEDTF) then TRTEDTF=TRTCEDTF;
else if ^missing(TRTeEDTF) then TRTEDTF=TRTeEDTF;
else if ^missing(TRTbEDTF) then TRTEDTF=TRTbEDTF;
format trtedtm date9.;
run;

*PN ------------------------------------------*;
*PN Read in SDTMDATA.EX                        *;
*PN Derive: Last Exposure variables            *;
*PN TR01EDT TR01ETM TR01EDTM                   *;
*PN ------------------------------------------*;

data tr01e_core ;
   merge sdtmdata.ex __dm(keep=studyid usubjid  ) __TRTEDT (drop = exendtc);
   by studyid usubjid;
   if length(exendtc) ge 10 then exendtn = input(substr(exendtc,1,10),yymmdd10.);
   if length(exstdtc) ge 10 then exstdtn = input(substr(exstdtc,1,10),yymmdd10.);

   if length(rfxendtc) gt 10 then rfxentm = input(substr(rfxendtc,12),time5.);
   if length(exstdtc)  gt 10 then exsttm  = input(substr(exstdtc,12),time5.);

run;

proc sort data=tr01e_core;
   by studyid usubjid exendtn;
   where exdose > 0;
run;

data tr01e_core;
   set tr01e_core;
   by studyid usubjid;
   attrib
         TR01EDT       length=8       label='Date of Last Exposure in Period 01'      format=date9.
         TR01ETM   length=8       label='Time of Last Exposure in Period 01'      format=time5.
         TR01EDTM  length=8       label='Datetime of Last Exposure in Period 01'  format=datetime19.
         ;

   if missing(xvrdt) then do;
      tr01edt = trtedt;
      tr01etm = lasttm;
      tr01edtm= lastdtm;
      *if nmiss(tr01edt,tr01etm)=0 and length(rfxendtc) ge 16 then tr01edtm = input(rfxendtc,e8601dt19.);
     end;

   if last.usubjid;

   keep studyid usubjid tr01edt tr01etm tr01edtm;

run;

*PN ---------------------*;
*PN Read in sdtm cm data *;
*PN ---------------------*;
%merge_supp(domain=cm, libname=sdtmdata);
```

```
%psort(ds=cm);

*PN ------------------------------------------*;
*PN Read in cm                                *;
*PN Derive: First Local Non-Rad Therapy Date  *;
*PN LOCACTDT                                  *;
*PN ------------------------------------------*;
*PN SM:20201102 Added Imputation to CMSTDTC *;

proc freq data =sdtmdata.suppcm;
   table qnam*qlabel/list missing;
run;

proc contents data =cm;
   title  'cmxxx';
run;


data cm1;
    merge cm __dm __randdt __TRTEDT;
    by studyid usubjid;

    if length(cmstdtc) ge 10 then cmstdtn = input(substr(cmstdtc,1,10),yymmdd10.);
                  * if cmstdtn ge randdt;

    * keep studyid usubjid trtsdt randdt cmstdtc cmstdtn cmcat cmstyn  cmtype  cmdecod cmtrt trtedt  ;

run;



proc sort data=cm1 out=locactdt;
    by studyid usubjid cmstdtn;
    where cmcat = 'CONCOMITANT AND SUBSEQUENT NON-RADIATION ANTI-CANCER THERAPY' and  CMdecOD not in
('LEUPRORELIN', 'GOSERELIN', 'DEGARELIX', 'TRPTORELIN')
    and (not missing(cmtrt) and not missing(cmstdtc));
run;


data  __locactdt(keep=studyid usubjid LOCACTDTC LOCACTDT  LOCACTWK) ;
                   merge locactdt  __randdt __eotdcdt;
                   by studyid usubjid;

   if first.usubjid ;

   LOCACTDTC=cmstdtc;

   if length(cmstdtc)<10 then do;
    dd=substr(cmstdtc,9,2);
                    mm=substr(cmstdtc,6,2);
                    yy=substr(cmstdtc,1,4);
                   end;
    trtedtc=put(trtedt, yymmdd10.);
    trtedyy=substr(trtedtc,1,4);
    trtedmm=substr(trtedtc,6,2);

                     if length(yy)=4 and length(mm)<2  then do;
                                    if input(yy,best.) > input(trtedyy,best.) then LOCACTDTC=strip(yy||'-01-01');
                      if input(yy,best.) = input(trtedyy,best.) then LOCACTDT=trtedt+1;
     end;

     else if length(yy)=4 and length(mm)=2 and length(dd)<2 then do;
                        LOCACTDTC=strip(substr(cmstdtc, 1, 7)) || '-01';
                        if input(LOCACTDTC,yymmdd10.)<=input(trtedtc,yymmdd10.) or (input(yy,best.) = input(trtedyy,best.) and input(mm,best.) >
input(trtedmm,best.) )  then LOCACTDT=trtedt+1;
```

```
    end;

    if ^missing(LOCACTDTC) & ^missing(cmstdtn) then LOCACTDT=cmstdtn;
    else if ^missing(LOCACTDTC) & missing(cmstdtn) then LOCACTDT=LOCACTDT;
    if ^missing(LOCACTDTC) & missing(LOCACTDT) then LOCACTDT=input(LOCACTDTC,yymmdd10.);

format LOCACTDT date9.;
  if nmiss(LOCACTDT,RANDDT)=0 then do;
    locactwk = (locactdt - randdt + 1) / 7;
  end;

run;

/*
data anaxl.__locactdt;
                set __locactdt;
run;
*/
* Derive: First Systemic Non-Rad Therapy Date  *;


proc sort data=cm1 out=sysactdt;
                  by studyid usubjid cmstdtc;
                  where cmcat = "CONCOMITANT AND SUBSEQUENT SYSTEMIC ANTI-CANCER THERAPY"  and cmstyn = 'Y'  and cmdecod
^ in ('LEUPRORELIN', 'GOSERELIN', 'DEGARELIX', 'TRPTORELIN'
)
   and (not missing(CMTRT) or not missing(CMSTDTC));
run;

data  __sysactdt(keep=studyid usubjid sysactdtc sysactdt SYSACTD SYSACTWK) ;
                 merge sysactdt  __randdt __eotdcdt;
                 by studyid usubjid;

   if first.usubjid ;
   sysactdt=cmstdtn;
   sysactdtc=cmstdtc;

   if length(cmstdtc)<10 then do;
    dd=substr(cmstdtc,9,2);
                  mm=substr(cmstdtc,6,2);
                  yy=substr(cmstdtc,1,4);
                 end;
    *** suppressed before ***;
    trtedtc=put(trtedt, yymmdd10.);
    trtedyy=substr(trtedtc,1,4);
    trtedmm=substr(trtedtc,6,2);

                   if length(yy)=4 and length(mm)<2  then do;
                                 if input(yy,best.) > input(trtedyy,best.) then sysactdtc=strip(yy||'-01-01');
                    if input(yy,best.) = input(trtedyy,best.) then sysactdt=trtedt+1;
     end;

     else if length(yy)=4 and length(mm)=2 and length(dd)<2 then do;
                      sysactdtc=strip(substr(cmstdtc, 1, 7)) || '-01';
                      if input(sysactdtc,yymmdd10.)<=input(trtedtc,yymmdd10.) or (input(yy,best.) = input(trtedyy,best.) and input(mm,best.) >
input(trtedmm,best.) )  then sysactdt=trtedt+1;
    end;

    *** suppressed before ***;

    eotdcdtc=put(EOTDCDT, yymmdd10.);
    EOTDCyy=substr(eotdcdtc,1,4);
    EOTDCmm=substr(eotdcdtc,6,2);

                   if length(yy)=4 and length(mm)<2  then do;
```

```
                    if input(yy,best.) > input(EOTDCyy,best.) then sysactdtc=strip(yy||'-01-01');
              if input(yy,best.) = input(EOTDCyy,best.) then sysactdt=EOTDCDT+1;
  end;

  else if length(yy)=4 and length(mm)=2 and length(dd)<2 then do;
              sysactdtc=strip(substr(cmstdtc, 1, 7)) || '-01';
              if input(sysactdtc,yymmdd10.)<=input(eotdcdtc,yymmdd10.) or (input(yy,best.) <= input(EOTDCyy,best.) and input(mm,best.) <
input(EOTDCmm,best.) )  then sysactdt=EOTDCDT+1;

  end;

  if ^missing(sysactdtc) & ^missing(cmstdtn) then sysactdt=cmstdtn;
  else if ^missing(sysactdtc) & missing(cmstdtn) then sysactdt=sysactdt;
   if ^missing(sysactdtc) & missing(SYSACTDT) then sysactdt=input(sysactdtc,yymmdd10.);
        *  if usubjid ='184315-5308-3020' then sysactdt = .;  *** Need to put cmstyn in the suppcm data in order to use cmstyn = N records
***;

format sysactdt date9.;
 if nmiss(SYSACTDT,RANDDT)=0 then do;
  SYSACTD = SYSACTDT-RANDDT+1;
  SYSACTWK =(SYSACTDT-RANDDT+1)/7;
 end;
  if sysactdt >= randdt;
  if CMTRT = ' ' and CMSTDTC = ' ' then delete;
run;

proc print data = cm1;
  title ' xxx cm1';
  where usubjid in ('184315-8606-3568');
run;


proc print data = sysactdt;
  title ' xxx';
  where usubjid in ('184315-8606-3568');
run;


*PN ------------------------------------------*;
*PN Read in ds                               *;
*PN Derive: Reason for Discon                 *;
*PN                                           *;
*PN ------------------------------------------*;

data __dctreasa(keep=studyid usubjid DCTREASA) __dctreasc(keep=studyid usubjid DCTREASC)
   __dctreasp(keep=studyid usubjid DCTREASp) __dctrease (keep=studyid usubjid DCTREASe)
   __dctreasb (keep=studyid usubjid DCTREasb) __dcsreas (keep=studyid usubjid DCsREAS )
   __dcrreas (keep =studyid usubjid dcrreas )
   ;
  set ds(keep = studyid usubjid dsscat dsdecod);

  attrib
      DCTREASA  length=$200       label='Reason for Discont of Atez'
      DCTREASC  length=$200       label='Reason for Discont of Cabo'
      DCTREASp  length=$200       label='Reason for Discont of Pred'
      DCTREASe  length=$200       label='Reason for Discont of Enza'
      DCTREASb  length=$200       label='Reason for Discont of Rbi'
      DCSREAS   length=$200       label='Reason for Discont. From Survival FUP'
      ;

  if dsscat = 'END OF STUDY TREATMENT, ATEZOLIZUMAB' then do;
    DCTREASA = dsdecod;
    output __dctreasa;
  end;
  if dsscat = 'END OF STUDY TREATMENT, CABOZANTINIB' then do;
```

```
      DCTREASC = dsdecod;
      output __dctreasc;
    end;
   if dsscat = 'END OF STUDY TREATMENT, ABIRATERONE' then do;
      DCTREASB = dsdecod;
      output __dctreasb;
    end;
   if dsscat = 'END OF STUDY TREATMENT, PREDNISONE' then do;
      DCTREASP = dsdecod;
      output __dctreasp;
    end;
   if dsscat = 'END OF STUDY TREATMENT, ENZALUTAMIDE' then do;
      DCTREASE = dsdecod;
      output __dctrease;
    end;
   if index(dsscat , 'END OF SURVIVAL FOLLOW-UP') gt 0  then do;
      DCSREAS = dsdecod;
                  /**
                    if dsdecod ne ' ' then DCSFL = 'Y';
                    if dsdecod eq ' ' then DCSFL = 'N';
                  **/
      output __dcsreas;
    end;
   if index(dsscat  , 'END OF RADIOGRAPHIC FOLLOW-UP') gt 0  then do;
      DCrREAS = dsdecod;
                  /*  if dsdecod ne ' ' then DCrFL = 'Y';
                    if dsdecod eq ' ' then DCrFL = 'N';  */
      output __dcrreas;
    end;
run;


proc print data =__dcsreas;
    title 'dcsreas';
    where usubjid ='184315-7109-3072';
run;


proc print data =__dcrreas;
    title 'dcrreas';
    where usubjid ='184315-7109-3072';
run;


*PN ---------------------*;
*PN Read in sdtm dd data *;
*PN ---------------------*;
%merge_supp(domain=dd, libname=sdtmdata);
%psort(ds=dd);

*PN -----------------------------------------*;
*PN Read in dd                               *;
*PN Derive: Reason for Death                 *;
*PN                                          *;
*PN -----------------------------------------*;

data __dthdate(keep=studyid usubjid dthdate) __DTHASOSI(keep=studyid usubjid DTHASOSI)
     ;
    set dd(keep = studyid usubjid ddtestcd ddorres dddtc );
  by studyid usubjid;
    attrib

          DTHASOSI  length=$200        label='Death Associated With Study Indication'
          ;
   if ddtestcd = 'PRCDTH' then do;
      dthdate = dddtc;
      output __dthdate;
   end;
   if ddtestcd = 'DTHRELPR' then do;
      dthasosi = ddorres;
      output __DTHASOSI;
   end;

run;

proc sort data=__DTHASOSI nodupkey;
              by studyid usubjid;
run;

proc sort data=__dthdate nodupkey;
              by studyid usubjid;
run;

*PN ---------------------*;
*PN Read in sdtm ae data *;
*PN ---------------------*;
%merge_supp(domain=ae, libname=sdtmdata);
%psort(ds=ae);

proc sort data=ae out=aedth(keep=studyid usubjid aeterm aedecod aerela aerelb aerelc aerele aeterm aedecod  aerelp aestdtc ) nodupkey;
  by studyid usubjid;
     where aetoxgr="5";
run;

proc sort data=ae out=aedthdt (keep=studyid usubjid aetoxgr aestdtc aeterm aedecod aerela aerelb aerelc aerele aerelp ) nodupkey;
  by usubjid;
     where aetoxgr="5";
run;

data __aedthdt (keep =usubjid aedthdtn aestdtc );
  set aedthdt;
     by usubjid;
  if length(aestdtc) ge 10 then aedthdtn = input(substr(aestdtc,1,10),yymmdd10.);
run;

proc print data = aedthdt;
   title '3632';
  where usubjid ='184315-1563-3632';
run;

proc print data = __aedthdt;
   title '3632';
  where usubjid ='184315-1563-3632';
run;


data __dthcaus /* (keep=studyid usubjid DTHCAUS ) */
                __DTHCGR1  /* (keep=studyid usubjid DTHCGR1 ) */
                __GR5RELFL  /*  (keep=studyid usubjid GR5RELFL ) */ ;
merge aedth(in=ae) __dthdate(in=dth) __dm (in = c  keep = studyid usubjid dthdtc where =(dthdtc ne ' ')) ;
by studyid usubjid;
   if ae or c;
   attrib
         DTHCAUS   length=$200       label='Cause of Death' ;

      if aeterm ne ' ' then dthcaus = aeterm;
      if  aeterm = ' '  then dthcaus='MISSING';
      output __dthcaus;
     if aedecod ne ' ' then DTHCGR1 = aedecod;
     if aedecod = ' '  then DTHCGR1='UNCODED/MISSING';
     output __DTHCGR1;


    if ae and (aerela='RELATED' or aerelc='RELATED' or aerele='RELATED' or aerelp ='RELATED' or aerelb='RELATED') then GR5RELFL
= 'Y';
    if ae and ((aerela='NOT RELATED' and aerelc='NOT RELATED') or  aerele='NOT RELATED'  or (aerelp ='NOT RELATED' and aerelb =
'NOT RELATED') )  then GR5RELFL = 'N';
    if   dthdtc ne ' '  and ^ ae  then GR5RELFL = 'N';
     output __GR5RELFL;
run;

proc sort data=__dthcaus nodupkey;
        by studyid usubjid;
run;

proc sort data=__DTHCGR1 nodupkey;
        by studyid usubjid;
run;

proc sort data=__GR5RELFL nodupkey;
        by studyid usubjid GR5RELFL ;
run;


proc freq data= __dthcaus;
  *  where dthcaus in ('MISSING',' ');
  table dthcaus*aeterm/list missing;;
  title 'dthcaus';
run;


proc freq data= __dthcgr1;
  * where dthcgr1 in ('UNCODED/MISSING',' ');
  table dthcgr1*aedecod/list missing;;
  title 'dthcgr1';
run;


proc freq data= __gr5relfl;
  * where gr5relfl in ('N', ' ');
  table usubjid*gr5relfl*aeterm*aedecod*aerela*aerelb*aerelc*aerele*aerelp/list missing;;
  title 'gr5relfl';
run;

proc freq data = __GR5RELFL;
  table gr5relfl;
run;

proc print data =dm;
  where usubjid ='184315-3004-3140';
run;


*PN ---------------------*;
*PN Read in sdtm sv data *;
*PN ---------------------*;
data sv;
   set sdtmdata.sv;

   if length(svstdtc) ge 10 then svstdtn = input(substr(svstdtc,1,10),yymmdd10.);
   if length(svendtc) ge 10 then svendtn = input(substr(svendtc,1,10),yymmdd10.);
   keep studyid usubjid svstdtn svendtn trtsdt;run;
```

```
proc sort data=sv out=sv1;
    by studyid usubjid svendtn;
run;

data __lstvisdt;
    set sv1(in=sv);
    by studyid usubjid svendtn;
    attrib  LSTVISDT   length=8           label='Last Visit Date'     format=date9.;
    lstvisdt=svendtn;
    if last.usubjid;
   keep studyid usubjid lstvisdt;
run;

proc sort data=sv out=_lstvisdt;
    by studyid usubjid ;
run;

proc sort data=__dm;
    by  usubjid ;
run;


data __lstvisdt;
                  merge __dm(in=dm keep=usubjid) __lstvisdt(in=sv);
                  by usubjid;
                  if dm & sv;
run;


*PN--------------------------------------------------------------PN*;
*PN--Macro to remove duplicate observation                   --*PN*;
*PN--------------------------------------------------------------PN*;

%macro dup(dsn=,var=);
proc sort data=&dsn nodupkey dupout=&dsn.dup out=&dsn.nodup;
by usubjid &var;
run;
%mend;

*PN ---------------------------------------------------------*;
*PN Merge supplemental qualifiers back onto parent domain ds *;
*PN ---------------------------------------------------------*;
%merge_supp(domain=vs, libname=sdtmdata) ;
%merge_supp(domain=qs, libname=sdtmdata) ;

*PN-------------------------------------PN*;
*PN-------------BASELINE----------------PN*;
*PN-------------------------------------PN*;
*PN---------------------------------------------PN*;
*PN--Macro to  subset dataset with condition-----PN*;
*PN---------------------------------------------PN*;

%macro condn(dsn=,var=,condn1=,condn2=);
%local;
data &var;
set &dsn;
if &condn1 then &var=&condn2;
else delete;
run;
%mend;
**Macro to find Bweight,bheight,becogc**;
%macro m_base(dsn=,testcd=,result=);
data &testcd._base_m;
set &dsn;
```

```
        if &dsn.testcd="&testcd"  and &dsn.blfl="Y";
        _base="Y";
        _base_n=&dsn.&result;
run;

proc sort data=&testcd._base_m;
 by usubjid;
run;
%condn(dsn=&dsn,condn1=&dsn.testcd="&testcd" and &dsn.&result ne .,var=&testcd,condn2=&dsn.&result);
data &testcd._no_basefl;
 merge &testcd._base_m(in=b drop=&dsn.blfl) &testcd(in=a);
 by usubjid;
 if a and not b;
 &dsn.testcd="&testcd";
 put "WAR" "NING: Baseline flag is missing for " "&testcd. : "  Usubjid=;
run;
data &testcd._base_m2;
merge sdtmdata.dm (in=a) &testcd._base_m(in=b);
by usubjid;
if a;
if _base="Y" and _base_n=. and &dsn.testcd="&testcd" then put "WAR" "NING: baseline value missing for" "&testcd."  Usubjid=;
run;
%mend;

****WEIGHT***************;
options mprint symbolgen;
%condn(dsn=vs,condn1=vstestcd="WEIGHT" and vsblfl="Y",var=wtbl,condn2=vsstresn);
%m_base(dsn=vs,testcd=WEIGHT,result=stresn);
%dup(dsn=WEIGHT_no_basefl,var=);

data weight_sub_no_fl;
merge WEIGHT_no_baseflnodup(in=a) vs(where=(vstestcd="WEIGHT"));
by usubjid;
if a;
if visit="SCREENING" or (visit='WEEK 1 DAY 1' and epoch='BASELINE' and vsdy le 1);
wtbl=vsstresn;
run;

data wtbl0;
merge wtbl weight_sub_no_fl;
by usubjid;
run;

proc sort data=wtbl0;
by usubjid vsdy;
run;

data weightbl(keep= studyid usubjid wtbl wtblgr1);
set wtbl0;
by usubjid vsdy;
   attrib
          WTBL      length=8           label='Baseline Weight (kg)'
          WTBLGR1   length=$15         label='Pooled Baseline Weight (kg) Group 1'
          ;
  if . lt wtbl lt 60 then wtblgr1 = '<60 kg';
   else if 60 le wtbl le 80 then wtblgr1 = '>=60 to <=80 kg';
   else if wtbl gt 80 then wtblgr1 = '>80 kg';
if last.usubjid;
run;

****HEIGHT**************;
%condn(dsn=vs,condn1=vstestcd="HEIGHT" and vsblfl="Y",var=htbl,condn2=vsstresn);
%m_base(dsn=vs,testcd=HEIGHT,result=stresn);
%dup(dsn=HEIGHT_no_basefl,var=);
```

```
data height_sub_no_fl;
merge HEIGHT_no_baseflnodup(in=a) vs(where=(vstestcd="HEIGHT"));
by usubjid;
if a;
if visit="SCREENING" or (visit='WEEK 1 DAY 1' and epoch='BASELINE' and vsdy le 1);
htbl=vsstresn;
run;

data htbl0;
merge htbl height_sub_no_fl;
by usubjid;
run;

proc sort data=htbl0;
by usubjid vsdy;
run;

data heightbl(keep= studyid usubjid htbl);
set htbl0;
by usubjid vsdy;
   attrib
          htbl      length=8           label='Baseline Height (cm)' ;
if last.usubjid;
run;

data __bmibl;
label  bmibl ='Baseline BMI (kg/m2)';
merge weightbl heightbl;
by usubjid;
if wtbl ne . and htbl ne . then bmibl=wtbl/((htbl*0.01)*(htbl*0.01));
run;

******BECOGC******;


data becogc_;
merge __dm(in=a keep=usubjid trtsdt) qs;
by usubjid;
if a;
if qstestcd="ECOG101";
if length(qsdtc)>=10 then becogcdt=input(substr(qsdtc,1,10),yymmdd10.);
format becogcdt date9.;
if .<becogcdt<=trtsdt;
run;
proc sort data=becogc_;
by usubjid becogcdt;
run;

data becogcdt;
set becogc_;
by usubjid becogcdt;
if last.usubjid;
ecogbl=qsstresn;
run;

proc sort data =__dm;
  by usubjid;
run;

data becogc_2;
merge __dm(in=a keep=usubjid trtsdt) qs;
by usubjid;
if a;
if trtsdt eq .;
if qstestcd="ECOG101";
if visit="SCREENING" or (visit='WEEK 1 DAY 1' and epoch='BASELINE' and qsdy le 1);
ecogbl=qsstresn;
run;

proc sort data=becogc_2;
by usubjid qsdy;
run;

data becogc_2;
set becogc_2;
by usubjid qsdy;
if last.usubjid;
run;

data __becogcdt;
merge becogcdt becogc_2;
by usubjid;
run;

*PN ------------------------------------------*;
*PN Read in  __dm                    *;
*PN Derive: PITTFL                           *;
*PN                                    *;
*PN ------------------------------------------*;

proc sort data=__dm;
   by studyid usubjid randdtc;
run;

data pittfl ;
   set __dm;
      by studyid usubjid randdtc;
   if not missing(trt01p) and not missing(randdtc) ;
   keep studyid randdtc usubjid trt01p randdtc ukrsufl;
run;


proc sort data = pittfl;
  by randdtc usubjid;
run;


data __pittfl (drop = ukrsufl);
  attrib OPITTFL        length=$1.        format=$1.             label="Original Prim PFS Population Flag";
   attrib PITTFL      length=$1          label='Primary PFS Population Flag';
   set pittfl;
      by randdtc usubjid;
   if _n_ = 1 then count=0;
              count + 1;
   if count le 400 and ukrsufl ne  'Y' then pittfl = 'Y';
                if count le 300 and ukrsufl ne  'Y' then opittfl = 'Y';
run;


proc sort data=__pittfl;
   by studyid usubjid;
run;


*PN ------------------------------------------*;
*PN Derive: LSTALVDT                         *;
*PN                                    *;
*PN ------------------------------------------*;

data aedth;
```

```
   set sdtmdata.ae;
       where AETOXGR = '5';
       dthdtc =aestdtc;
                keep studyid usubjid dthdtc  AETOXGR;
run;


data lstalvdt_ (keep=studyid usubjid lstalvdt_ lsalvdtf);
        length lsalvdtf $1;
                set dm (keep=studyid usubjid dthdtc ) aedth;
                    where dthdtc > '';
*               lstalvdt_= input(dthdtc,yymmdd10.);
                if length (dthdtc) = 7   then           dthdtc_= strip(dthdtc)||'-01';
                else if length (dthdtc) = 10   then     dthdtc_= dthdtc;
        if length (dthdtc_) = 10   then                 lstalvdt_= input(dthdtc_,yymmdd10.);
                if length (dthdtc) = 7   then           lsalvdtf ='D';;
run;


proc sort data = lstalvdt_ nodupkey ;
     by studyid usubjid  lstalvdt_;
run;

data dsa(keep=studyid usubjid dsadt) dsb(keep=studyid usubjid dsbdt) ;
    set sdtmdata.ds;
    if dsscat = 'RE-CONSENT OR CONSENT WITHDRAWAL' and index(dsterm,'WITHDREW FULL CONSENT') = 1  then do;
        if length(dsstdtc) ge 10 then dsadt = input(substr(dsstdtc,1,10),yymmdd10.);
        output dsa;
      end;
    else if dsscat = 'END OF SURVIVAL FOLLOW-UP' and dsdecod ^= "DEATH" then do;
        if length(dsstdtc) ge 10 then dsbdt = input(substr(dsstdtc,1,10),yymmdd10.);
        output dsb;
      end;
run;

proc sort data=dsa;
                 by studyid usubjid dsadt;
run;

data dsa;
set dsa;
                 by studyid usubjid dsadt;
if last.usubjid;
run;


proc print data = dsa;
    where usubjid = '184315-3005-3492';
run;


proc sort data=dsb;
                 by studyid usubjid dsbdt;
run;

data dsb;
set dsb;
                 by studyid usubjid dsbdt;
if last.usubjid;
run;

%merge_supp(domain=ss, libname=sdtmdata);
%psort(ds=ss);

data ssc(keep=studyid usubjid ALIVEDT);
```

```
          set ss;
 if SSCAT = "SURVIVAL FOLLOW-UP" and SSORRES="ALIVE";
 alivedt=input(alivedtc,yymmdd10.);
run;

proc sort data=ssc;
          by studyid usubjid ALIVEDT;
run;

data ssc;
set ssc;
          by studyid usubjid ALIVEDT;
if last.usubjid;
run;

/**
data eot (keep=studyid usubjid eotdt);
  set sdtmdata.ds;
  if index(dsscat,'END OF STUDY TREATMENT') = 1  then do;
    if length(dsstdtc) ge 10 then eotdt = input(substr(dsstdtc,1,10),yymmdd10.);
    output eot;
          end;
run;
**/

data eot (keep=studyid usubjid dsstdtc);
  set sdtmdata.ds;
  if index(dsscat,'END OF STUDY TREATMENT') = 1 ;
run;

/**
proc sort data =eot;
  by studyid usubjid eotdt;
run;

data eot;
  set eot;
    by studyid usubjid eotdt;
  if last.usubjid;
run;
**/

data dsds (keep=studyid USUBJID dsstdtc dscat dsterm dsscat);
  set sdtmdata.ds ;
   where dsscat in ('END OF RADIOGRAPHIC FOLLOW-UP','END OF STUDY TREATMENT' ,'RE-CONSENT OR CONSENT
WITHDRAWAL');
run;

data sv (drop = svstdtc  svendtc cmstdtc cmendtc   aestdtc aeendtc  exstdtc exendtc  lbdtc);
  set sdtmdata.sv (keep=studyid USUBJID svstdtc)
    sdtmdata.sv (keep=studyid USUBJID svendtc)
                    sdtmdata.cm (keep=studyid USUBJID cmstdtc)
    sdtmdata.cm (keep=studyid USUBJID cmendtc)
            sdtmdata.ex (keep=studyid USUBJID exstdtc)
    sdtmdata.ex (keep=studyid USUBJID exendtc)
                     sdtmdata.lb (keep=studyid USUBJID lbdtc)
                     sdtmdata.ss (keep=studyid USUBJID ssdtc)
                      sdtmdata.ae (keep=studyid USUBJID aestdtc)
                     eot
                     dsds (keep=studyid usubjid dsstdtc)
          /**        sdtmdata.ds (keep=studyid USUBJID dsstdtc dscat dsterm dsscat where = (dsscat ='END OF RADIOGRAPHIC
FOLLOW-UP' or dscat in ('END OF STUDY TREATMENT')
     or (dscat ='RE-CONSENT OR CONSENT WITHDRAWAL' and DSTERM in ( 'WITHDREW CONSENT FROM STUDY
INTERVENTIONS', 'NON-INTERVENTIONAL STUDY ASSESSMENTS MAY CONTINUE') )))
                    */
```

```
                  sdtmdata.ae (keep=studyid USUBJID aeendtc);
    if not missing(svstdtc) then svall =    substr(svstdtc,1,10);
                  if not missing(svendtc) then svall =    substr(svendtc,1,10);
                  if not missing(cmstdtc) then svall =    substr(cmstdtc,1,10);
                  if not missing(cmendtc) then svall =    substr(cmendtc,1,10);
                  if not missing(exstdtc) then svall =    substr(exstdtc,1,10);
                  if not missing(exendtc) then svall =    substr(exendtc,1,10);
            if not missing(aestdtc) then svall =    substr(aestdtc,1,10);
                  if not missing(aeendtc) then svall =    substr(aeendtc,1,10);
                  if not missing(lbdtc) then svall =    substr(lbdtc,1,10);
                  if not missing(ssdtc) then svall =    substr(ssdtc,1,10);
                  if not missing(dsstdtc) then svall =    substr(dsstdtc,1,10);
                  if not missing(aestdtc) then svall =    substr(aestdtc,1,10);
                  if not missing(aeendtc) then svall =    substr(aeendtc,1,10);
    if length(svall) = 10 then svstdt = input(svall,yymmdd10.);
                  format svstdt date9.;



proc sort data=sv ;
  by studyid usubjid svstdt;
  *  where length (strip(svall)) = 10 ;
run;

data sv;
  set sv;
     by studyid usubjid svstdt;
  if last.usubjid;
run;


data __lstalvdt ; * (keep=studyid usubjid lstalvdt lsalvdtf) ;
         merge dsa(in=a) dsb(in=b) ssc(in=c) lstalvdt_ (in=d) eot (in=e) sv (in=max) ;
           by studyid usubjid;
         cutdt = &cutdt.;
         if dsbdt ne . or ALIVEDT ne . then maxdate = max(dsbdt, ALIVEDT);
         if d and lstalvdt_ ne . then lstalvdt=lstalvdt_;
         else if dsadt ne . then lstalvdt=dsadt;
         else if dsadt eq . and dsbdt ne . or ALIVEDT ne . then lstalvdt=max(dsbdt, ALIVEDT);
         else if dsadt eq . and dsbdt eq . and ALIVEDT eq . and  svstdt ne . then lstalvdt=svstdt;
         else if dsadt eq . and dsbdt eq . and ALIVEDT eq . and  svstdt eq . then lstalvdt=cutdt;

  *  else if (not a) and (not b) and ^ c and  ^ max  and e and eotdt ne . then lstalvdt=eotdt;
         /*
  else lstalvdt=  max( dsadt,dsbdt ,ALIVEDT,eotdt);
         else if a then lstalvdt=dsadt;
         else if (not a) and b then lstalvdt=dsbdt;
         else if (not a) and (not b) and c then lstalvdt=ALIVEDT;
         else if (not a) and (not b) and (not c) then lstalvdt=eotdt;
         **/
         /* else lstalvdt= .;*/
         * lsalvdtf=";
         *if lstalvdt > &cutoffdt. then lstalvdt=&cutoffdt. ;
         * lstalvdt=  max( dsadt,dsbdt ,ALIVEDT,eotdt);
  format lstalvdt date9.;
run;



proc sort data =__lstalvdt;
  by usubjid lstalvdt;
run;

data __lstalvdt;
  set __lstalvdt;
```

```
      by usubjid lstalvdt;
   if last.usubjid;
run;


proc freq data= __lstalvdt;
   table  usubjid* lstalvdt* lstalvdt_* dsadt* dsbdt* ALIVEDT* svstdt* eotdt/list missing nopct nocum;
   format  lstalvdt lstalvdt_ dsadt dsbdt ALIVEDT svstdt eotdt date9.;
run;



*PN ----------------------------------------------------*;
*PN read in sdtm su data                               *;
*PN ----------------------------------------------------*;
data smok(keep=&sortby smokstat) ;
  set sdtmdata.su;
  length smokstat $10;
  if sutrt = 'TOBACCO' then do;
    if suoccur = 'N' then smokstat = 'NEVER';
    else if suoccur = 'Y' then do;
      if suenrtpt = 'BEFORE' then smokstat = 'FORMER';
      if suenrtpt = 'ONGOING' then smokstat = 'CURRENT';
    end;
    output smok;
  end;

run;

%psort(ds=smok);

proc sort data=smok out=__smok nodupkey;
    by studyid usubjid;
run;

data alch(keep=&sortby alchstat);
  set sdtmdata.su;
  length  ALCHSTAT $10;

  if sutrt = 'ALCOHOL' then do;
    if suoccur = 'N' then ALCHSTAT = 'NEVER';
    else if suoccur = 'Y' then do;
      if suenrtpt = 'BEFORE' then ALCHSTAT = 'FORMER';
      if suenrtpt = 'ONGOING' then ALCHSTAT = 'CURRENT';
    end;
    output alch;
  end;
run;
%psort(ds=alch);

proc sort data=alch out=__alch nodupkey;
    by studyid usubjid;
run;

*PN --------------------*;
*PN Read in sdtm ds data *;
*PN --------------------*;
%merge_supp(domain=ds, libname=sdtmdata);
%psort(ds=ds);

data __dsstdtc_withdrew;
   set ds;
   by usubjid;
   if index(upcase(dsterm),'WITHDREW FULL CONSENT') and upcase(dsscat) in("RE-CONSENT OR CONSENT WITHDRAWAL") then do;
      __dsstdtc_withdrew = input(substr(DSSTDTC,1,10),yymmdd10.);
    end;
   if last.usubjid;
   keep studyid usubjid __dsstdtc_withdrew;
run;

*PN ---------------------*;
*PN Read in sdtm cm data *;
*PN ---------------------*;
%merge_supp(domain=cm, libname=sdtmdata);
%psort(ds=cm, key=usubjid cmstdtc cmendtc);


proc sql undo_policy=none;
  create table cm as
    select a.*, b.randdt
    from cm a left join __randdt b
    on a.usubjid=b.usubjid
    order by a.studyid, a.usubjid, a.cmstdtc,a.cmendtc;
quit;

data cm2(keep=studyid usubjid cmstdtc cmendtc NRADSDT NRADSDTF NRADEDT NRADEDTF nraddurd nraddurw nraddurm);
  set cm;
  by usubjid cmstdtc cmendtc;
    if cmcat="HISTORY OF NON-RADIATION ANTI-CANCER THERAPY"  and cmdecod in ("ENZALUTAMIDE" "ABIRATERONE"
"APALUTAMIDE" "DAROLUTAMIDE") and cmstdtc>" and (not (cmtrt=" and cmstdtc=")) ; * and cmtype ="SYSTEMIC" ;
  *  if last.usubjid;
  format NRADSDT NRADEDT date9.;
  length NRADSDTF NRADEDTF $3;

  if cmstdtc = '21' then cmstdtc = '2021';  *** ERROR 2229-3633 partial yr 20 and 21 **;
   if cmstdtc = '20' then cmstdtc = '2020';  *** ERROR 2229-3633 partial yr 20 and 21 **;
  if length(cmstdtc)>=10 then NRADSDT=input(cmstdtc, yymmdd10.);
  else do;
    * to be confimed if imputation is needed???;
    dd=substr(cmstdtc,9,2);
    mm=substr(cmstdtc,6,2);

    if 4<=length(cmstdtc)<10 then do;
      if dd=" and mm=" then do; cmstdtc=substr(cmstdtc,1,4)||'-01-01'; NRADSDTF='M'; end;  ***ERROR 2229-3633 partial  yr 21 and 20 **;
      else if dd=" then do; cmstdtc=substr(cmstdtc,1,7)||'-01'; NRADSDTF='D'; end;
    end;
    NRADSDT=input(substr(cmstdtc,1,10),yymmdd10.);
  end;


  if length(cmendtc)>=10 then NRADEDT=input(substr(cmendtc,1,10),yymmdd10.);
  else do;
    * impute cmendtc;
    if length(cmendtc)>=" then do;
      dd=substr(cmendtc,9,2);
      mm=substr(cmendtc,6,2);
      yy=substr(cmendtc,1,4);
    end;

    if length(yy)=4 and mm=" then do;
      * missing month/day or month;
      cmendtc=compbl(strip(yy)||'-12-31');
      NRADEDT=input(cmendtc, yymmdd10.);
      NRADEDTF='M';
    end;
    else if dd=" then do;
      * missing day;
      %x_dtimpute(indtc=cmendtc, outdt=NRADEDT, outdtf=NRADEDTF, impalign=e);
      NRADEDTF='D';
    end;
```

```
 end;

   * if NRADEDT>randdt then NRADEDT=randdt-1;
           if nmiss(randdt, NRADEDT)=0 then nraddurd  = randdt - NRADEDT ;
  if ^missing(nraddurd) then nraddurw  = nraddurd / 7;
  if ^missing(nraddurd) then nraddurm  = nraddurd / 30.4375;
run;

proc print data =cm2;
  title 'cm2';
  where usubjid = '184315-3420-3774';
run;

proc freq data = cm2;
  table cmendtc*  NRADEDT* NRADEDTF/list missing;
run;

data __NRAD;
 set cm2;
 by usubjid cmstdtc cmendtc;
 if last.usubjid;
 NRADEDF=NRADEDTF;
 keep studyid usubjid NRADSDT NRADSDTF NRADEDT NRADEDF NRADEDTF nraddurd nraddurw nraddurm;
run;

*PN ---------------------*;
*PN Read in sdtm cm HISTORY OF NON-RADIATION ANTI-CANCER THERAPY data *;
*PN ---------------------*;
data ypnr npnr;
   set cm;
 by usubjid cmstdtc cmendtc;
   if cmcat='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY'  then do;
   PNRDTXFL='Y';
          output ypnr;
          end;
          else if cmcat  ^ ='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY' then do;
   PNRDTXFL='N';
          output npnr;
          end;

run;
/**
data PRNHTGR;
  length PRNHTGR1 $20;
  set sdtmdata.cm;
  if CMCAT='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY' and  CMDECOD= 'ABIRATERONE' then PRNHTGR1 =
'ABIRATERONE' ;
  if CMCAT='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY' and  CMDECOD= 'ENZALUTAMIDE' then PRNHTGR1 =
'ENZALUTAMIDE' ;
  if CMCAT='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY' and  CMDECOD in ('APALUTAMIDE','DAROLUTAMIDE')
then PRNHTGR1 = 'OTHER' ;
run;
**/

proc sort data=ypnr  nodupkey;
by usubjid cmtrt cmstdtc cmendtc;
* by usubjid regnum;
run;

data ypnr0(keep=studyid usubjid PNRDTXFL NPNRADTX);
 set ypnr;
 NPNRADTX + 1;
 by usubjid cmtrt cmstdtc cmendtc;
 * by usubjid regnum;
  if first.usubjid then NPNRADTX = 1;
  if last.usubjid;
run;

proc sort data=npnr;
 by usubjid cmtrt cmstdtc cmendtc;
* by usubjid regnum;
run;


data npnr0(keep=studyid usubjid PNRDTXFL NPNRADTX);
  set npnr;
  by usubjid cmtrt cmstdtc cmendtc;
* by usubjid regnum;
  NPNRADTX = 0;
  if last.usubjid;
run;

data npryn;
length NPNRDGR1 $8.;
set ypnr0 npnr0;
by usubjid ;
if NPNRADTX=0 then NPNRDGR1='0';
else if NPNRADTX=1 then NPNRDGR1='1';
else if NPNRADTX=2 then NPNRDGR1='2';
else if NPNRADTX>=3 then NPNRDGR1='>=3';
run;


proc sort data=npryn out=__npryn nodupkey dupout=test;
  by usubjid ;
run;



*PN ----------------------------------------------------------------------------------------------------------*;
*PN read in PR (PRCAT=HISTORY OF RADIATION THERAPY) and (PR.PRINDC=DISEASE UNDER STUDY) & PROCCUR in ('','Y') *;
*PN ----------------------------------------------------------------------------------------------------------*;

proc sort data=sdtmdata.pr out=pr1;
     by studyid  usubjid prtrt prstdtc prendtc ;
run;


data yprad nprad;
     set pr1;
  by usubjid prtrt prstdtc prendtc;
    if (PRCAT='HISTORY OF RADIATION THERAPY') and (PRINDC='DISEASE UNDER STUDY') and (PROCCUR in ('','Y'))  then do;
    PRADTXFL='Y';
                output yprad;
                end;
                else do;
    PRADTXFL='N';
                output nprad;
                end;

run;

proc sort data=yprad nodupkey;
by usubjid prtrt prstdtc prendtc;
run;

data yprad0(keep=studyid usubjid PRADTXFL NPRADTX);
  set yprad;
  NPRADTX + 1;
  by usubjid prtrt prstdtc prendtc;
  if first.usubjid then NPRADTX = 1;
  if last.usubjid;
run;

proc sort data=nprad;
  by usubjid prtrt prstdtc prendtc;
run;


data nprad0(keep=studyid usubjid PRADTXFL NPRADTX);
  set nprad;
  by usubjid prtrt prstdtc prendtc;
  NPRADTX = 0;
  if last.usubjid;
run;

data pradyn;
length NPRADGR1 $8.;
set yprad0 nprad0;
by usubjid ;
if NPRADTX=0 then NPRADGR1='0';
else if NPRADTX=1 then NPRADGR1='1';
else if NPRADTX=2 then NPRADGR1='2';
else if NPRADTX>=3 then NPRADGR1='>=3';
run;


proc sort data=pradyn out=__pradyn nodupkey dupout=test;
  by usubjid ;
run;



*PN ---------------------*;
*PN Read in sdtm pr data *;
*PN ---------------------*;
%merge_supp(domain=pr, libname=sdtmdata);
%psort(ds=pr);

* Derive: First Subsequent Radiation Therapy Date  *;

data pr1;
    merge pr __dm __randdt __TRTEDT;
    by studyid usubjid;

    if length(prstdtc) ge 10 then prstdtn = input(substr(prstdtc,1,10),yymmdd10.);

    keep studyid usubjid trtsdt randdt prstdtn prstdtc prcat prindc prtrt rasityp csitl /*  csprel */  visit TRTEDT;

run;

proc sort data=pr1 out=prsrdt;
    by studyid usubjid prstdtc;
    where prcat = 'CONCOMITANT AND SUBSEQUENT RADIATION THERAPY' and PRINDC='DISEASE UNDER STUDY' and
RASITYP^='BONE' and VISIT^='SCREENING'
    and (not missing(prtrt) and not missing(prstdtc)) ;
run;

data  __prsrdt(keep=studyid usubjid prsrdtc prsrdt ) ;
                   merge prsrdt  __randdt;
                   by studyid usubjid;

   if first.usubjid ;
   *sysactdt=cmstdtn;
   prsrdtc=prstdtc;

   if length(prstdtc)<10 then do;
    dd=substr(prstdtc,9,2);
                   mm=substr(prstdtc,6,2);
                   yy=substr(prstdtc,1,4);
                  end;
    trtedtc=put(trtedt, yymmdd10.);
    trtedyy=substr(trtedtc,1,4);
    trtedmm=substr(trtedtc,6,2);

                    if length(yy)=4 and length(mm)<2  then do;
                                   if input(yy,best.) > input(trtedyy,best.) then prsrdtc=strip(yy||'-01-01');
                     if input(yy,best.) = input(trtedyy,best.) then prsrdt=trtedt+1;
     end;

     else if length(yy)=4 and length(mm)=2 and length(dd)<2 then do;
                        prsrdtc=strip(substr(prstdtc, 1, 7)) || '-01';
                        if input(prsrdtc,yymmdd10.)<=input(trtedtc,yymmdd10.) or (input(yy,best.) = input(trtedyy,best.) and input(mm,best.) >
input(trtedmm,best.) )  then prsrdt=trtedt+1;
    end;

    if ^missing(prsrdtc) & ^missing(prstdtn) then prsrdt=prstdtn;
    else if ^missing(prsrdtc) & missing(prstdtn) then prsrdt=prsrdt;
    if ^missing(prsrdtc) & missing(prsrdt) then prsrdt=input(prsrdtc,yymmdd10.);

format prsrdt date9.;

   if prsrdt >= randdt;
run;

*Derive: First Date of Subsequent Surgery / Procedure Impacting Tumor Lesion  *;



proc sort data=pr1 out=prssptldt;
    by studyid usubjid prstdtn;
    where prcat = 'CONCOMITANT/SUBSEQUENT SURGERY AND PROCS' and csitl='Y' /* and csprel='Y' */ and VISIT^='SCREENING'
    /* and (not missing(prtrt) and not missing(prstdtn)) and prstdtn ge trtsdt */;
run;

data __prssptldt;
    set prssptldt;
    by studyid usubjid;

    attrib
            prssptldt     length=8        label='First Date of Subsequent Surgery or Procedure Impacting Tumor Lesion'     format=date9.
            ;
    if first.usubjid;
    prssptldt = prstdtn;
    keep studyid usubjid prssptldt;

run;


*PN ---------------------*;
*PN Read in sdtm cd data *;
*PN ---------------------*;
%merge_supp(domain=cd, libname=sdtmdata);
%psort(ds=cd);

data __diag; *(keep=studyid usubjid DIAGDT DIAGDTF diagdurd  diagdury);
  merge cd __dm __randdt __rficdt;
    by studyid usubjid;
   if CDTESTCD="CDPCDTC"   and  cdstresc >' ' ;   ***ERROR suppress > ' ' just for matching purpose **;
  format DIAGDT  date9.;
  length DIAGDTF  $3;
  if length(cdstresc)>=10 then DIAGDT=input(cdstresc, yymmdd10.);
  else do;
    dd=substr(cdstresc,9,2);
    mm=substr(cdstresc,6,2);
    if 4<=length(cdstresc)<10 then do;
      if dd='' and mm='' then do; cdstresc=substr(cdstresc,1,4)||'-07-01'; DIAGDTF='M'; end;
      else if dd='' then do; cdstresc=substr(cdstresc,1,7)||'-15'; DIAGDTF='D'; end;
    end;
    DIAGDT=input(substr(cdstresc,1,10),yymmdd10.);
 end;
  *  if DIAGDT>rficdt then DIAGDT=rficdt-1;
    diagdurd  = randdt - diagdt ;
    diagdury  = diagdurd / 365.25;

run;

proc print data = __diag;
   title 'diag';
   where usubjid = '184315-2229-3633';   *** ALLUCENT ERROR need to reset to inform consent date  ***;
run;

*PN ---------------------*;
*PN Read in sdtm ds data *;
*PN ---------------------*;
%merge_supp(domain=ds, libname=sdtmdata);
%psort(ds=ds);

data __eordt(keep=studyid usubjid DCRREAS eordt eslctdt eslbsdt eslmrdt );
  set ds;
  format eordt eslctdt eslbsdt eslmrdt date9.;
if upcase(dsscat) in("END OF RADIOGRAPHIC FOLLOW-UP");
eslctdt =input(eslctdtc,yymmdd10.);
eslbsdt =input(eslbsdtc,yymmdd10.);
eslmrdt =input(eslmrdtc,yymmdd10.);
DCRREAS=dsdecod;
if nmiss(eslctdt, eslbsdt, eslmrdt) lt 3 then eordt=max( of eslctdt, eslbsdt, eslmrdt );
run;

proc sort data = __eordt;
   by usubjid eordt;
run;

data __eordt;
   set __eordt;
      where eordt ne .;
         by usubjid eordt;
      if last.usubjid;
run;



proc datasets lib=work nolist;
                  change __dm=___dm;
run;
quit;


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

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

proc sort data =__EOTDCDT;
  by usubjid;
run;

proc sort data =__EOTDCDT;
  by usubjid;
run;

proc sort data =__LOCACTDT;
  by usubjid;
run;

proc sort data =__SYSACTDT;
  by usubjid;
run;

proc sort data =__TRTAEDT;
  by usubjid;
run;

proc sort data =__TRTCEDT;
  by usubjid;
run;


proc sort data =__TRTEDT;
  by usubjid;
run;


proc sort data =__TRTEEDT;
  by usubjid;
run;

proc sort data =__EORDT nodupkey;
  by usubjid;
run;

data __dsx (keep=studyid usubjid dsxdt)  ;
   set sdtmdata.ds;
   if dsscat = 'RE-CONSENT OR CONSENT WITHDRAWAL' and index(dsterm,'WITHDREW FULL CONSENT FROM ALL STUDY INTERVENTIONS AND NON-INTERVENTIONAL STUDY ASSESSMENTS.') > 0  then do;
       if length(dsstdtc) ge 10 then dsxdt = input(substr(dsstdtc,1,10),yymmdd10.);
       output __dsx;
     end;
run;

proc sort data=__dsx;
  by usubjid;
run;

/***
%merge_supp(domain=cm, libname=sdtmdata);
%merge_supp(domain=pr, libname=sdtmdata);

data npact;
      merge __trtend (keep= studyid usubjid trtedt)
            cm       (in=cm keep= studyid usubjid cmstdtc cmcat cmtype
           where= (cmcat='CONCOMITANT AND SUBSEQUENT NON-RADIATION ANTI-CANCER THERAPY' and cmtype='SYSTEMIC'));
        by studyid usubjid;
         if cm;
```

```
run;

data sprod;
      merge __trtend (keep= studyid usubjid trtedt)
            pr       (in=pr);
         by studyid usubjid;
        if pr;
run;

%xl313imp (indsn=sprod, indate=prstdtc, refdt=trtedt, conmed_npact=y);

data raddt;
  set sprod;
   where prcat='CONCOMITANT AND SUBSEQUENT RADIATION THERAPY' and rasityp ne 'BONE' and visit ne 'SCREENING' and
PRINDC='DISEASE UNDER STUDY';
 run;
 %psort (ds=raddt, out=raddt2, where=%str(prstdtc ne ' '), key= prstdtc, first_last=first);

 data surgdt;
   set sprod;
      where prcat='CONCOMITANT AND SUBSEQUENT SURGERIES AND PROCEDURES' and csitl='Y' and visit ne 'SCREENING' ;
  run;
%psort (ds=surgdt, out=surgdt2, where=%str(prstdtc ne ' '), key= prstdtc, first_last=first);
***/

*PN --------------------------------------------------------------------------------------------------------*;
*PN END OF ADSL DERIVATIONS, MERGE ALL DATASETS BELOW AND COMPARE *;
*PN ---------------------------------------------------------------------------------------------------------*;


proc sort data = sdtmdata.pr out = prpg nodupkey;
    by usubjid PRTRT PRSTDTC;
      where  PRCAT='SURGERY AND OTHER PROCEDURES HISTORY' and PRLOC='PROSTATE GLAND';
run;

proc freq data =prpg noprint;
   table usubjid/ out= prpgout ;
run;

data __prpgout;
    length NPSURGR1 $20;
    set prpgout;
                  by usubjid;
               npsurg = count;
    NPSURGR1 = put (count,1.);
               if count > = 3 then NPSURGR1 = '>=3';
run;

proc sort data=___dm;
   by usubjid;
run;

proc sort data = odsdata.randomdo out = __randomdo (keep=usubjid armcd);
   by usubjid;
run;

proc sort data = sdtmdata.ds out = dsfu (keep= usubjid dsstdtc);
     where  DSTERM="LOST TO FOLLOW-UP" and DSSCAT in ('END OF SURVIVAL FOLLOW-UP','END OF RADIOGRAPHIC FOLLOW-
UP');
   by usubjid;
run;

data __dsfu (keep = usubjid LTFUDT);
   attrib LTFUDT format = date9. label ="Date of Lost to Follow Up";
   set dsfu;
```

```
     by usubjid;
   if dsstdtc ne ' ' then LTFUDT = input (dsstdtc , is8601da.);
run;


data all;
   merge ___dm(in=dm) __aedthdt __dsx __dcrreas __diag __dthcaus  __dthcgr1  __sysactdt  __dcsreas __gr5relfl   __eosdt __dsfu __:
  ;
   by usubjid;
if dm;
   attrib
                  TRT01PD  length=$25          label='Dummy TRT01P'
                                      TRT01AD  length=$25         label='Dummy TRT01A'
         EOTDCODT  length=8          label='Date of Last Dose Decision Ongoing, Core'       format=date9.
         EOAODCDT  length=8          label='Date of Last Dose Dec Ongo (Atez, Core)'        format=date9.
         EOAODT    length=8          label='Date of Last Dose Ongo (Atez, Core)'           format=date9.
         EOCODCDT  length=8          label='Date of Last Dose Dec Ongo (Cabo, Core)'        format=date9.
         EOBODCDT  length=8          label='Date of Last Dose Dec Ongo (ARBI)'        format=date9.
         EOEODCDT  length=8          label='Date of Last Dose Dec Ongo (ENZO)'        format=date9.
                                      EOPODCDT  length=8          label='Date of Last Dose Dec Ongo (PRED)'        format=date9.
         EOECASDT  length=8          label='Date of Start of SOP (Ext, Core, Atezo)'        format=date9.
         EOECCSDT  length=8          label='Date of Start of SOP (Ext, Core, Combo)'        format=date9.
         EOECMSDT  length=8          label='Date of Start of SOP (Ext, Core, Mono)'         format=date9.
         EOSOBEDT  length=8          label='Date of End of SOP (Standard, Core)'           format=date9.
         EOECAEDT  length=8          label='Date of End of SOP (Ext, Core, Atezo)'          format=date9.
         EOECCEDT  length=8          label='Date of End of SOP (Ext, Core, Combo)'          format=date9.
         EOECMEDT  length=8          label='Date of End of SOP (Ext, Core, Mono)'           format=date9.


         TRCRDURD  length=8          label='Total Treatment Duration, Core (Days)'
         TRCRDURW  length=8          label='Total Treatment Duration, Core (Weeks)'
         TRCRDURM  length=8          label='Total Treatment Duration, Core (Months)'
         TRTXDURD  length=8          label='Total Treatment Duration, XVR (Days)'
         TRTXDURW  length=8          label='Total Treatment Duration, XVR (Weeks)'
         TRTXDURM  length=8          label='Total Treatment Duration, XVR (Months)'
         TRTDURD   length=8          label='Total Treatment Duration (Days)'
         TRTDURW   length=8          label='Total Treatment Duration (Weeks)'
         TRTDURM   length=8          label='Total Treatment Duration (Months)'
         TRTADURD  length=8          label='Total Atezolizumab Duration (Days)'
         TRTCDURD  length=8          label='Total Cabozantinib Duration (Days)'
         TRTCDURM  length=8          label='Total Cabozantinib Duration (Months)'
         TRTSDURD  length=8          label='Total Sorafenib Duration (Days)'
         TRTSDURM  length=8          label='Total Sorafenib Duration (Months)'
         LSTALVDT    length=8          label='Last Known Alive' format=date9.
         STRATAV       length=$200       label='Strata from Verification Source'

         DTHDT     length=8        label='Date of Death'     format=date9.
         DTHDTF    length=$1       label='Date of Death Imputation Flag'
         DTHUNFDT  length=8        label='Date of Death Unfiltered'     format=date9.
         DTHUNDTF  length=$1       label='Date of Death Unf Imp Flag'
         DTHDY     length=8        label='Relative Day of Death'
         DTHSL0DY  length=8        label='Day of Death Rel to Last Non-Zero Dose'
         DTHPER    length=$50      label='Death Period (30 Days)'
         DTHPER2    length=$50      label='Death Period (100 Days)'
                                      DURNRADM   length=8        label ='Dur Lst Non-Rad Tx (Mnths)'
         DNRADGR1   length=$15      label='Dur Lst Non-Rad Tx Group 1'
                                       FNHTGR1                 length=$20.        format=$10.              label="First NHT Collapsed Group"
          DOCGR1        length=$10.        format=$10.              label="Prior Docetaxel Collapsed Group"
          DSGR1          length=$10.        format=$10.              label="Pres of Liver meta Collapsed Group"
                                       MSAFFL   length=$1.        format=$1.              label="Modified Safety Population Flag"
          MITTFL         length=$1.        format=$1.              label="Modified ITT Population Flag"
                                       CUTRNKDT                length=8                  label="Date of 202nd Event Date"  format=date9.
                                       RANDCDTM                length=8                 label="Randomization Datetime"  format=Datetime19.
                                       CUTRDDTM                length=8                   label="Rand Datetime of 202nd Event Date"   format=Date9.
                                       DURNRADM                length=8      label="Dur Lst Non-Rad Tx (Mnths)"  format=best20.
     DNRADGR1 length=$15.     format=$15.        label="Dur Lst Non-Rad Tx Group 1"
     DCSFL    length=$1.     format=$1.        label="Reason for Discont. From Surv FUP Flag"
     DCRFL    length=$1.     format=$1.        label="Reason for Discont. From Radio FUP Flag"
    ;
          if saffl = 'Y' and dcsreas ne ' ' then dcsfl = 'Y';
          else if saffl = 'Y' and dcsreas eq ' ' then  dcsfl = 'N';
          if saffl = 'Y' and dcrreas ne ' ' then dcrfl = 'Y';
         * else dcrfl = 'N';
CUTRNKDT = "12dec2022"d;
RANDCDTM = input (RANDCDTC,is8601dt.);
        CUTRDDTM =  1975755005;
DURNRADM  = (NRADEDT-NRADSDT)/30.4375;
if . < DURNRADM < 1 then  DNRADGR1 = '< 1';
if 1  le DURNRADM < 3 then  DNRADGR1 = '>= 1 to 3';
        if 3  le DURNRADM < 6 then  DNRADGR1 = '>= 3 to 6';
        if 6  le DURNRADM then  DNRADGR1 = '>= 6';
if SAFFL='Y' /** and PITTFL='Y' **/  and UKRSUFL ^= 'Y' then MSAFFL = 'Y';
    else MSAFFL = 'N';
        if RANDDT ^ = .  and UKRSUFL ^= 'Y'  then MITTFL ='Y';
    Else MITTFL ='N';
        if armcd = 'ABI+PRED\ENZA' then do;
          trt01pd =  'Second NHT';
   trt01pdn = 1;
                  trt01ad =  'Second NHT';
   trt01adn = 1;
        end;
if armcd = 'CABO+ATEZO' then do;
          trt01pd =  'Cabozantinib+Atezolizumab';
   trt01pdn = 0;
                  trt01ad =  'Cabozantinib+Atezolizumab';
   trt01adn = 0;
        end;

if lstalvdt > cutdt then lstalvdt=cutdt ;
if  . < lstalvdt <= randdt then lstalvdt = randdt;
        if lstalvdt = . and trtsdt = . and randdt ne . then lstalvdt = randdt;
if length(dthdtc) ge 10 then dthdtn = input(substr(dthdtc,1,10),yymmdd10.);
dddd=substr(dthdtc,9,2);
ddmm=substr(dthdtc,6,2);
ddyy=substr(dthdtc,1,4);
dddd=substr(dthdtc,9,2);
ddmm=substr(dthdtc,6,2);
ddyy=substr(dthdtc,1,4);
if dthdtn ne . then do;
  %dtimpute(outvar=dthdtn ,day=dddd, month=ddmm, year=ddyy,flagvar=dthundtf,align=b);
end;

 if dthdt ne .  and LTRTOGDT ne .  then dthsl0dy = DTHDT - LTRTOGDT + 1;
if dthfl = 'Y' then do;
 if ^missing(dthdtn) then dthunfdt = dthdtn;
   if missing(dthdtn) then do;
          if LSTALVDT ne . then dthunfdt = LSTALVDT; ***  +1 ;

                              if length(deathc) <7 then do;
              dthundtf ='Y';
              dthdtf='Y';
                                      end;
                                      * if length(deathc) =7 then do;
                                      if dddd = ' ' and ddmm ne ' '  then do;
              dthundtf ='D';
              dthdtf='D';
                                      end;

                            if LSTALVDT eq . and aedthdtn ne . then dthunfdt = aedthdtn;
  end;
 end;

if not missing(dthunfdt) then dthdt = dthunfdt;
if dthunfdt gt cutdt then dthdt = .;
* if not missing(dthdt) then dthdtf=dthundtf;
if nmiss(dthdt, trtsdt)=0 then dthdy = dthdt - trtsdt + 1;
* if nmiss(dthdt, trtedt)=0 then dthsl0dy = dthdt - trtedt + 1;
* if dthdt ne .  and LTRTOGDT ne .  then dthsl0dy = DTHDT - LTRTOGDT + 1;

format LTRTOGDT date9.;
if not missing(trtsdt) then do;
if  not missing(trtedt) then LTRTOGDT=trtedt;
else if not missing(dthdt) then LTRTOGDT=  min(dthdt, cutdt);
else LTRTOGDT= cutdt;
end;

if nmiss(dthdt,ltrtogdt)=0 and (((dthdt - ltrtogdt + 1)) <= 30) then dthper = 'Death <= 30 days within last dose';
if nmiss(dthdt,ltrtogdt)=0 and (((dthdt - ltrtogdt + 1))  > 30) then dthper = 'Death > 30 days after last dose';

if nmiss(dthdt,ltrtogdt)=0 and (((dthdt - ltrtogdt + 1)) <= 100) then dthper2 = 'Death <= 100 days within last dose';
if nmiss(dthdt,ltrtogdt)=0 and (((dthdt - ltrtogdt + 1))  > 100) then dthper2 = 'Death > 100 days after last dose';

 if not missing(TRTBEDT) and missing(EOBDCDT)  then  TRTBEDT=.;
            if not missing(TRTPEDT) and missing(EOPDCDT)  then  TRTPEDT=.;
            if not missing(TRTEEDT) and missing(EOEDCDT)  then  TRTEEDT=.;

 if not missing(eotdcdt) and saffl = 'Y' then eotdcodt = eotdcdt;
 else if not missing(dthdt) and saffl = 'Y'  then eotdcodt = min(dthdt,cutdt);
 else IF DTHDT = .  and saffl = 'Y'  then eotdcodt = cutdt;


 format FSBACTDT date9.;
            * if n(locactdt,sysactdt,prsrdt,prssptldt)<=4 then FSBACTDT = min(locactdt,sysactdt,prsrdt,prssptldt);
            if n(locactdt,sysactdt,prsrdt,prssptldt)<=4 then FSBACTDT = min(sysactdt,prsrdt,prssptldt);
 * FSBACTDT = min(of locactdt,sysactdt,prsrdt,prssptldt);

 if trt01an = 0 then do;
   if not missing(eoadcdt) and saffl = 'Y' then eoaodcdt = eoadcdt;
   else if not missing(dthdt)  and saffl = 'Y'  then eoaodcdt = min(dthdt,cutdt);
   else if  saffl = 'Y' and dthdt = . then  eoaodcdt = cutdt;
 end;
            if trt01an =0 then do;;
   if not missing(eocdcdt) and saffl = 'Y' then eocodcdt = TRTCEDT;
   else if not missing(dthdt) and saffl = 'Y' then eocodcdt = min(dthdt,cutdt);
   else if saffl = 'Y' and dthdt = .  then eocodcdt = cutdt;
   if saffl ne 'Y' then eocodcdt = .;
 end;
            if trt01an =1 then do;;
   if not missing(eobdcdt) and saffl = 'Y' and  TRTNHT='Abi+Pred' then eobodcdt = TRTBEDT;
   else if not missing(dthdt) and saffl = 'Y' and  TRTNHT='Abi+Pred' then eobodcdt = min(dthdt,cutdt);
   else if saffl = 'Y' and dthdt = .  and  TRTNHT='Abi+Pred' then eobodcdt = cutdt;
   * if saffl ne 'Y' then eobodcdt = .;
 end;
            if trt01an =1 then do;;
   if not missing(eoedcdt) and saffl = 'Y' and  TRTNHT='Enza' then eoeodcdt = TRTEEDT;
   else if not missing(dthdt) and saffl = 'Y' and  TRTNHT='Enza' then eoeodcdt = min(dthdt,cutdt);
   else  if saffl = 'Y' and dthdt = .   and  TRTNHT='Enza' then eoeodcdt = cutdt;
   * if saffl ne 'Y' then eoeodcdt = .;
 end;
            if trt01an =1 then do;;
   if not missing(eopdcdt) and saffl = 'Y' and  TRTNHT='Abi+Pred' then eopodcdt = TRTPEDT;
   else if not missing(dthdt) and saffl = 'Y' and  TRTNHT='Abi+Pred' then eopodcdt = min(dthdt,cutdt);
   else  if saffl = 'Y' and dthdt = .  and  TRTNHT='Abi+Pred'  then eopodcdt = cutdt;
   * if saffl ne 'Y' then eopodcdt = .;
 end;
 * if nmiss(trtaedtx,dthdt,cutdt)=0 then eoaodt = min(trtaedtx+21,dthdt,cutdt);
  if nmiss(trtaedtx,dthdt,cutdt) < = 2   and trtaedtx ne .  then eoaodt = min(trtaedtx,dthdt,cutdt);

 if not missing(eosdcdt) then eosodcdt = eosdcdt;
 else if not missing(dthdt) then eosodcdt = min(dthdt,cutdt);
 else eosodcdt = cutdt;
 if saffl ne 'Y' then eosodcdt = .;

 if not missing(tr01edt) then do;
  if not missing(trtaedt) then eoecasdt = trtaedt + 30;
  if eoecasdt gt cutdt gt .z then eoecasdt = .;
  if eoecasdt gt dthdt gt .z then eoecasdt = .;
  if eoecasdt gt wdcfdt gt .z then eoecasdt = .;
 end;

 if not missing(tr01edt) then do;
  if trt01p ='Cabozantinib+Atezolizumab' then EOECCSDT = tr01edt + 30;
  if EOECCSDT gt cutdt gt .z then EOECCSDT = .;
  if EOECCSDT gt dthdt gt .z then EOECCSDT = .;
  if EOECCSDT gt wdcfdt gt .z then EOECCSDT = .;
 end;

 if  not missing(tr01edt) then do;
  if trt01p ^='Cabozantinib+Atezolizumab'  then eoecmsdt = tr01edt + 30;
  if eoecmsdt gt cutdt gt .z then eoecmsdt = .;
  if eoecmsdt gt dthdt gt .z then eoecmsdt = .;
  if eoecmsdt gt wdcfdt gt .z then eoecmsdt = .;
 end;

 EOSOBSDT=TRTSDT;

 if trtedt ne . then eoeccsdt = trtsdt;

if not missing(eosobsdt) then eosobedt = min((trtedt+30),dsxdt,DTHUNFDT,cutdt);
if trtedt ne . then trtedtt= trtedt + 100;
if eoeccsdt ne . then eoeccedt = min(trtedtt,dsxdt,dthunfdt,cutdt);
if not missing(eoecmsdt) then eoecmedt = min((tr01edt+100),(xvrdt-1),__dsstdtc_withdrew,dthdt,cutdt);
        if not missing(eoecasdt) then eoecaedt = min((trtaedt+100),__dsstdtc_withdrew,dthdt,cutdt);


if nmiss(eotdcodt,trtsdt)=0 then do;
trcrdurd  = eotdcodt - trtsdt + 1;
trcrdurw  = trcrdurd / 7;
trcrdurm  = trcrdurd / 30.4375;
        end;
if nmiss(eotxdcdt,tr02sdt)=0 then do;
trtxdurd  = eotxdcdt-tr02sdt +1;
trtxdurw  = trtxdurd / 7;
trtxdurm  = trtxdurd / 30.4375;
        end;

        if trt01pn = 0 then do;
            TRTDURD = max(EOTDCODT,TRTAEDT)-TRTSDT +1;
                    TRTDURw = (max(EOTDCODT,TRTAEDT)-TRTSDT +1)/7;
                    TRTDURm = (max(EOTDCODT,TRTAEDT)-TRTSDT +1)/30.4375;
        end;
if trt01pn = 1 then do;
            TRTDURD = EOTDCODT-TRTSDT  +1;
                    TRTDURw = (EOTDCODT-TRTSDT +1 )/7;
                    TRTDURm = (EOTDCODT-TRTSDT +1 )/30.4375;
        end;

if nmiss(trtaedt,trtasdt)=0 then trtadurd  = trtaedt - trtasdt +1;
if nmiss(trtaedt,trtasdt)=0 then trtadurm  = (trtaedt - trtasdt +1)/30.4375;
```

```
   if nmiss(eocdcdt,trtcsdt)=0 then trtcdurd  = eocdcdt - trtcsdt +1;
   if nmiss(eocdcdt,trtcsdt)=0 then trtcdurm  = (eocdcdt - trtcsdt +1)/30.4375;
   if nmiss(eobdcdt,trtbsdt)=0 then trtbdurd  = eobdcdt - trtbsdt +1;
   if nmiss(eobdcdt,trtbsdt)=0 then trtbdurm  = (eobdcdt - trtbsdt +1)/30.4375;
           if nmiss(eopdcdt,trtpsdt)=0 then trtpdurd  = eopdcdt - trtpsdt +1;
   if nmiss(eopdcdt,trtpsdt)=0 then trtpdurm  = (eopdcdt - trtpsdt +1)/30.4375;
           if nmiss(eoedcdt,trtesdt)=0 then trtedurd  = eoedcdt - trtesdt +1;
   if nmiss(eoedcdt,trtesdt)=0 then trtedurm  = (eoedcdt - trtesdt +1)/30.4375;

   if nmiss(eocodcdt,trtcsdt)=0 then trocdurd  = eocodcdt - trtcsdt + 1;
   if not missing(trocdurd) then trocdurm  = (eocodcdt - trtcsdt + 1)/ 30.4375;
           if nmiss(eoaodcdt,trtasdt)=0 then troadurd  = eoaodcdt - trtasdt + 1;
   if not missing(troadurd) then troadurm  = (eoaodcdt - trtasdt + 1)/ 30.4375;
           if nmiss(eobodcdt,trtbsdt)=0 then trobdurd  = eobodcdt - trtbsdt + 1;
   if not missing(trobdurd) then trobdurm  = (eobodcdt - trtbsdt + 1)/ 30.4375;
           if nmiss(eoeodcdt,trtesdt)=0 then troedurd  = eoeodcdt - trtesdt + 1;
   if not missing(troedurd) then troedurm  = (eoeodcdt - trtesdt + 1)/ 30.4375;
           if nmiss(eopodcdt,trtpsdt)=0 then tropdurd  = eopodcdt - trtpsdt + 1;
   if not missing(tropdurd) then tropdurm  = (eopodcdt - trtpsdt + 1)/ 30.4375;

           if PSADSDT ne . then PSADDURM=          (RANDDT- PSADSDT )/30.4375;
   if PSADSDT ne . then PSADDURW=       (RANDDT- PSADSDT )/7;

           if RHTXSDT ne . then RHTXDURM=          (RANDDT- RHTXSDT )/30.4375;
   if RHTXSDT ne . then RHTXDURW=       (RANDDT- RHTXSDT )/7;

           if PSURSDT ne . then PSURDURM=          (RANDDT- PSURSDT )/30.4375;
   if PSURSDT ne . then PSURDURW=       (RANDDT- PSURSDT )/7;

           if RDPRSDT ne . then RDPRDURM=          (RANDDT- RDPRSDT )/30.4375;
   if RDPRSDT ne . then RDPRDURW=       (RANDDT- RDPRSDT )/7;


   if missing(eligibfl) then eligibfl = 'Y';
   if missing(ittfl) then ittfl = 'N';
    if missing(pittfl) then pittfl = 'N';
   if missing(opittfl) then opittfl = 'N';
   if SAFFL='Y' and PITTFL='Y' then PSAFFL='Y';
   else PSAFFL= 'N';
   *else PSAFFL=";

   if missing(PNRDTXFL) then do;
           PNRDTXFL='N';NPNRADTX=0;NPNRDGR1='0';
   end;

   if missing(PRADTXFL) then do;
           PRADTXFL='N';NPRADTX=0;NPRADGR1='0';
   end;

  if DTHDT>EOECAEDT>. then DTHAEAFL='Y'; else DTHAEAFL= 'N';
  if DTHDT>EOECCEDT>. then DTHAECFL='Y'; else DTHAECFL= 'N';
  ***Still Pending ***;
  if DTHDT>max (of EOECCEDT,EOECAEDT, EOECCEDT, EOECMEDT/*, EOEXAEDT, EOEXCEDT*/) >. then DTHAESFL='Y'; else
DTHAESFL= 'N';

TRTAXSDT = .;
TRTCXSDT = .;
EOXDCODT = .;
EOSSXSDT = .;
EOEXASDT = .;
EOEXCSDT = .;
EOSSXEDT = .;
EOEXAEDT = .;
EOEXCEDT = .;
```

```
  if RACE = "WHITE" then racegr1 = "White";
  else  if RACE in ( "Black or African American","BLACK OR AFRICAN AMERICAN") then racegr1 =  "Black/African American" ;
  else  if RACE = "ASIAN" then racegr1= "Asian";
  else  racegr1 ="Other"; * "Rest of the races reported/Not Reported";

   if dthdt ne .  and LTRTOGDT ne .  then dthsl0dy = DTHDT - LTRTOGDT + 1;
          if nradedt = . then nradedf = ' ';
          if TRTcEDT = . then TRTcEDTf = ' ';
           if TRTEDT = . then TRTEDTf = ' ';
           if trt01pn = 0 then trtnht = ' ';
          *  if saffl = 'Y' and DCSREAS ne ' ' then DCSFL = 'Y';
          *   else if  saffl = 'Y' and DCSREAS eq ' ' then DCSFL = 'N';
    if saffl = 'Y' and  DCRREAS ne ' ' then DCRFL = 'Y';
           else if  saffl = 'Y' and  DCRREAS eq ' ' then DCRFL = 'N';
           if PNRTMCFL ^ = 'Y' then PNRTMCFL = 'N';
           if NPNRTXMC < 1 then NPNRTXMC = 0;
           if NPNRMGR1  = ' ' then NPNRMGR1  = '0';

           DURNRADM = (NRADEDT-NRADSDT)/30.4375;
    if   . lt DURNRADM lt 1 then DNRADGR1 ='< 1';
            if  1 le DURNRADM lt 3 then DNRADGR1 ='>= 1 to < 3';
                     if  3 le DURNRADM lt 6 then DNRADGR1 ='>=3 to < 6';
                     if   DURNRADM ge 6 then DNRADGR1 ='>=6';
          if region1 = ' ' then region1 =  'MISSING';
          if dscrf = ' '  and randdt ne . then dscrf = 'No';
          if doccrf = ' ' and randdt ne . then doccrf = 'No';
          cut76dt = cutdt;
          dth76dt = dthunfdt;
          lst76adt = lstalvdt;
          dth76dtf = dthunfdtf;
          ls76adtf = lstalvdtf;
          if dth76dt > cutdt then dth76dt = .;
          If (LSTVISDT > CUTDT > . or  LSTVISDT > DTHDT > .) then LSTVISDT= min(DTHDT, CUTDT);
    if rfxendtc >  &cutdt then rfxendtc = &cutdt.;
          if pittfl = 'Y' then do;
           FNHTGR1               = 'mCSPC/m0CRPC/mCRPC';
    DOCGR1            =DOCIXRS;
    DSGR1 =DSIXRS;
          end;
 run;



proc sort data = __WDCFDT out=dup nodupkey;
  by usubjid;
run;


proc sort data =all  dupout=dup nodupkey;
  by usubjid;
run;

proc print data =all ;
   title 'check dthunf';
  where usubjid = '184315-1563-3632';
run;




data adsl;
          %attrib_adsl;
           set all;
  keep STUDYID USUBJID SUBJID SITEID COUNTRY  STRATAR STRATARN DSIXRS DOCIXRS FNHTIXRS STRATAV STRATAVN
DSCRF DOCCRF FNHTCRF PROTVS REGION1
```

```
    AGE AGEU AGEGR1 AGEGR2 AGEGR3 SEX RACE RACEOTH RACEMULT RACEGR1 ETHNIC  SAFFL PSAFFL ITTFL PITTFL
ELIGIBFL UKRSUFL ARM  ACTARM SCRNFFL
    TRT01P TRT01PN TRT01A TRT01AN TRT01PD TRT01PDN TRT01AD TRT01ADN TRTNHT PGICYN RFICDT RANDDT TRTSDT
TRTSDTM TRTASDT TRTCSDT TRTBSDT TRTPSDT
    TRTESDT RFXENDTC TRTEDT TRTEDTF   TRTETM  TRTEDTM   DTH76DTF LS76ADTF  TRTAEDT TRTAEDTF TRTCEDT
TRTCEDTF TRTBEDT TRTBEDTF TRTPEDT TRTPEDTF TRTEEDT TRTEEDTF LTRTOGDT
    EOADCDT EOCDCDT EOBDCDT EOPDCDT EOEDCDT EOTDCDT EOTDCODT EOAODCDT EOAODT EOCODCDT EOBODCDT
EOPODCDT EOEODCDT WDCFDT LTFUDT LSTALVDT LSALVDTF
    LST76ADT LS76ADTF DTHDT DTHDTF DTHDTC DTHUNFDT DTHUNDTF DTHDY DTH76DT DTH76DTF DTHSL0DY DTHPER
DTHPER2 DTHAESFL EOSOBSDT EOECCSDT EOSOBEDT
    EOECCEDT CUTDT CUT76DT SYSACTDT SYSACTD SYSACTWK FSBACTDT TRTDURD TRTDURW TRTDURM TRTADURD
TRTADURM TRTCDURD TRTCDURM TRTBDURD TRTBDURM
    TRTPDURD TRTPDURM TRTEDURD TRTEDURM TROADURD TROADURM TROCDURD TROCDURM TROBDURD
TROBDURM TROPDURD TROPDURM TROEDURD TROEDURM DCTREASA
    DCTREASC DCTREASB DCTREASP DCTREASE DTHCAUS DTHCGR1 DTHASOSI  GR5RELFL EOSRDTC EOSDT EOSDTF
DCSREAS DCSFL EORDT DCRREAS DCRFL LSTVISDT
    WTBL WTBLGR1 HTBL BMIBL ECOGBL SMOKSTAT ALCHSTAT DIAGDT DIAGDTF DIAGDURD DIAGDURY NRADSDT
NRADSDTF NRADEDT NRADEDF NRADDURM PSADSDT PSADSDTF
    PSADDURM PSADDURW RDPRSDT RDPRSDTF RDPRDURM RDPRDURW RHTXSDT RHTXSDTF DURNRADM DNRADGR1
    RHTXDURM RHTXDURW PSURSDT PSURSDTF PSURDURW NPSURG NPSURGR1 PNRDTXFL NPNRADTX NPNRDGR1
PNRTMCFL NPNRTXMC NPNRMGR1 PRADTXFL NPRADTXNPRADGR1  FNHTGR1 DOCGR1 DSGR1
    opittfl msaffl mittfl LTFUDT CUTRNKDT RANDCDTM CUTRDDTM DURNRADM DNRADGR1 dcsfl dcrfl;

run;


proc sort data=adsl nodupkey;
        by _all_;
run;


*PN --------------------------------------*;
*PN Create permanent dataset for analysis *;
*PN --------------------------------------*;

data anaxl.&dsname(
        type     = ANALYSIS
        label    = "Subject-Level Analysis Dataset"
        sortedby = &sortby
        ) ;
  retain &sortby subjid  ;

  set &dsname ;

run ;


%mktitles();

proc sort data = anadata.adsl out=adsl_P;
  by usubjid;
run;

data adsl_p (drop=TRTEDTMx);
  set adsl_p (rename= (TRTEDTM=TRTEDTMx));
    by usubjid;
      TRTEDTM = datepart(TRTEDTMx);
run;

proc compare base=adsl_p  compare=anaxl.adsl  (drop = TRTETM)  maxprint=10000 listall;
  id usubjid;
run;
%mkpages();
```

```
data anaxl.&dsname(
                type      = ANALYSIS
                label     = "Subject-Level Analysis Dataset"
                sortedby = &sortby
                ) ;
                %attrib_adsl;
  set &dsname ;
run ;


/***

libname export  xport "../export/&anadate/&dsname..xpt";

 data export.&dsname(replace=yes);
    set anaxl.&dsname ;
run;
***/


%*=----------------------------------;
%* QC comparison                    ;
%*=----------------------------------;

data adsl;
  set anadata.adsl;
   * keep usubjid EOADCDT EOBDCDT EOEDCDT EOCDCDT EOPDCDT dthdt cutdt;
  drop  M      NPSURG  NPSURGR1  PNRTMCFL NPNRTXMC NPNRMGR1 ;
  *format _all_;
   /*  &block if usubjid="&subject"; */
run;


data v_adsl;
  set anaxl.adsl;
   drop trtetm trtedtm  CNTRYGR1  REGIONI  REGNIGR1 ETIOIXRS  REGIONC  ETIOCRF  TRTSEQP  TRT02P  TRT02PN   TRT02A
TRT02AN  XVRDT   RFIC2DT  TRTSSDT
          TRTAXSDT  TRTCXSDT  TR01SDT   TR01STM  TR01SDTM   TR01EDT  TR01ETM   TR01EDTM  TR02SDT  TR02STM
TR02SDTM    TR02EDT  TR02ETM
                                    TR02EDTM  TRTSEDT TRTAXEDT TRTCXEDT  EOSDCDT EOXADCDT  EOXCDCDT EOTXDCDT
EOXDCODT EOSODCDT DSEVTADT  DSEVTCDT  DSEVTSDT  DSEVXADT
                                                DSEVXCDT DTHAEAFL DTHAECFL  EOSSXSDT  EOECASDT  EOECMSDT EOEXASDT EOEXCSDT
EOSSXEDT  EOECAEDT  EOECMEDT  EOEXAEDT  EOEXCEDT  TRCRDURD
          TRCRDURW  TRCRDURM  TRTXDURD  TRTXDURW  TRTXDURM  TRTSDURD TRTSDURM   TROSDURD  TROSDURM
DCTREASS  DCTRESXA DCTRESXC /* NPNRMGR1 NPNRTXMC PNRTMCFL NPSURGR1 NPSURG */                    ;
run;



%mktitles();
proc compare base=adsl   compare=v_adsl maxprint=10000 listall  ;
  id usubjid;
  * var EOADCDT EOBDCDT EOEDCDT EOCDCDT EOPDCDT dthdt cutdt;
run;
%mkpages();


/**
%mktitles();
proc contents data=anaxl.adsl;
run;
%mkpages();
***/

                /**
```

```
data PRNHTGR1 ;
   length PRNHTGR1  $20;
   set sdtmdata.cm;
   if CMCAT='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY' and  CMDECOD= 'ABIRATERONE' then PRNHTGR1 =
'ABIRATERONE' ;
   if CMCAT='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY' and  CMDECOD= 'ENZALUTAMIDE' then PRNHTGR1 =
'ENZALUTAMIDE' ;
   if CMCAT='HISTORY OF NON-RADIATION ANTI-CANCER THERAPY' and  CMDECOD in ('APALUTAMIDE','DAROLUTAMIDE')
then PRNHTGR1 = 'OTHER' ;
run;



libname oldadsl "\\celerra03\SAS\stat\xl184\184315\dev\anadata\cro\20221219\archive\20230222";
libname newadsl "\\celerra03\SAS\stat\xl184\184315\dev\anadata\cro\20221219";


proc compare data =oldadsl.adsl compare =newadsl.adsl listall;
    id studyid usubjid;
run;

                  **/
```