```
/*Evaluation of FLuorometholone as Adjunctive MEdical Therapy for Trachomatous Trichiasis Surgery: The FLAME Randomized Controlled Clinical Trial*/
/*purpose: generate Kaplan-Meier curves for cumulative incidence of postoperative trachomatous trichiasis during follow-up by treatment groups*/
/*Note: raw datasets in SAS library 'in' corresponds to the shared data file in xlsx format, their relationship are as below:
in.eligibility - EL
in.PRIMARY_END_M12, in.PRIMARY_END_M6, in.PRIMARY_END_W4 - month 12, month 6 and week 4 record in PE*/

/*set up SAS library, TLF path, TLF title, format catalog which will be used in later analysis*/
%include "setup.sas";
libname in "&_root_in";
libname fm "&_root_fm";
libname data "&_root_data";

%let fig_path =&_root_output\Fig2.pdf;
%let TLF_title = Supplemental Figure: Kaplan-Meier curves for the cumulative incidence of postoperative trachomatous trichiasis during follow-up by treatment groups;
OPTIONS FMTSEARCH=(fm.fmelig fm.fmprimary);

/*set up variable format for visualization*/
proc format;
        value trtf  0 = "Placebo" 1 = "Fluorometholone";
        value fake  0 = "Group 1" 1 = "Group 2" 2 = "All";
        value gf  0 = "A" 1 = "B" 2 = "C";
        value ynf 1='Yes' 0='No';
        value typef 2 = "One or more lashes touching the globe"
                                3 = 'Surgery for post-operative TT'
```

```
                                    4 = 'Evidence of epilation';
        value $ qf 'peeye_0' = "Do not have primary endpoints"
                                    'peeye_2' = "   One or more lashes touching the globe in upper lids"
                                    'peeye_3' = '   Surgery for post-operative TT in upper lids'
                                    'peeye_4' = '   Evidence of epilation in upper lids'
                                    'have' = "Have primary endpoint in upper lids *";
run;

* Rename Macro - rename old variable name as new variable name;
%macro proc_rename1(oldvarlist, newvarlist);
 %let k=1;
 %let old = %scan(&oldvarlist, &k);
 %let new = %scan(&newvarlist, &k);
   %do %while(("&old" NE "") & ("&new" NE ""));
    rename &old = &new;
          %let k = %eval(&k + 1);
    %let old = %scan(&oldvarlist, &k);
    %let new = %scan(&newvarlist, &k);
 %end;
%mend;

*remove r_, l_ term from variables in two dataset and turn eye-level variables into patient-level variables
turn records in dataset from patient-level into eye-level;
%macro manipulate_data(indata,outdata);
data temp_primary;
        set &indata;
run;
```

```
* extract variable names from dataset;
proc sql noprint;
	create table varnames as
	select name
	from dictionary.columns
	 where libname='WORK' and memname= 'TEMP_PRIMARY';
quit;

* extract left/right eye related variable name;
data right_var left_var ;
	set varnames;
	if substr(name,length(name)-4,2) = "r_" then do;
	right = 1;
	main = substr(name,1,length(name)-5)||substr(name,length(name),2);
	end;
	else if substr(name,length(name)-4,2) = "l_" then do;
	right = 0 ;
	main = substr(name,1,length(name)-5)||substr(name,length(name),2);
	end;
	if right = 1 then output  right_var;
	else if right = 0 then output left_var;
run;

proc sql noprint;
	select name into : OD_names separated by ', '
	from right_var;
```

```
        select name into : OS_names separated by ', '
        from left_var;

        select name into : OD_names2 separated by ' '
        from right_var;

        select name into : OS_names2 separated by ' '
        from left_var;

        select main into : new_names separated by ' '
        from left_var;
quit;
 %put &OD_names;
 %put &OS_names;
 %put &OD_names2;
 %put &OS_names2;
 %put &new_names;

* extract OS/OD variables into different dataset,
 reports Restricted to Eligible eyes;
proc sql;
        create table Primary_OD as
        select e.subjid,t.itt_treat as treat format=trtf.,
        &OD_names,"OD" as eye
        from  Temp_PRIMARY as e left join  IN.ID_GROUP as t
        on e.subjid = t.subjid
```

```
        where e.subjid in (select subjectid from IN.ELIGIBILITY2 where  elelig_r= 1);

        create table Primary_OS as
        select e.subjid,t.itt_treat as treat format=trtf.,
        &OS_names,"OS" as eye
        from  Temp_PRIMARY as e left join  IN.ID_GROUP as t
        on e.subjid = t.subjid
        where e.subjid in (select subjectid from IN.ELIGIBILITY2 where elelig_l = 1);
quit;

*remove r_, l_ term from variables in two dataset;
data Primary_OD2;
 set Primary_OD;
 %proc_rename1(&OD_names2,&new_names);
run;

data Primary_OS2;
 set Primary_OS;
 %proc_rename1(&OS_names2,&new_names);
run;

* combine datasets, turn patient-level dataset into eye-level dataset;
data PRIMARY;
        set Primary_OD2 Primary_OS2;
run;

data  &outdata;
```

```
        set PRIMARY;
        if peeye_2=1 or peeye_3=1 or peeye_4=1 then have = 1;
        else if peeye_0=1 then have = 0;
run;

proc datasets library=WORK nolist;
        delete  RIGHT_VAR LEFT_VAR PRIMARY TEMP_PRIMARY VARNAMES  PRIMARY_OD PRIMARY_OD2 PRIMARY_OS
PRIMARY_OS2;
run;

%mend manipulate_data;

%manipulate_data(indata= IN.PRIMARY_END_W4,outdata = Primary_W4);
%manipulate_data(indata= IN.PRIMARY_END_M6,outdata = Primary_M6);
%manipulate_data(indata= IN.PRIMARY_END_M12,outdata = Primary_M12);

/*extract Eligible eye*/
proc sql;
        create table eli_OS as
        select  subjectid as subjid, "OS" as eye
        from  IN.ELIGIBILITY2
        where subjectid ne " " and  elelig_l = 1;

        create table eli_OD as
        select  subjectid as subjid, "OD" as eye
        from  IN.ELIGIBILITY2
        where subjectid ne " " and  elelig_r = 1;
```

```
        create table eligible as
        select * from eli_OS
        union
        select * from eli_OD;
quit;
proc datasets library=WORK nolist;
        delete eli_: ;
run;

proc sql;
        create table  eligible as
        select e.subjid,e.eye,t.itt_treat as treat format=trtf.
        from eligible as e left join  IN.ID_GROUP as t on e.subjid = t.subjid;
quit;

/*generate datasets for km plot*/
proc sql;
        create table endpoint_time as
        select a.subjid, a.eye, a.treat, b.have as endpoint_w4, c.have as endpoint_M6,d.have as endpoint_M12,
        case when endpoint_w4 = 1 then 28-1
                when endpoint_m6 = 1 then 168-1
                when endpoint_m12 = 1 then 365-1
                when endpoint_M12 = 0 then 380
                when endpoint_M6 = 0 then 168
                when endpoint_w4 = 0 then 28
                when endpoint_w4 = . then 0 end as time,
```

```
        case when endpoint_w4 = 1 or endpoint_M6 = 1 or endpoint_M12 = 1 then 1
                        else 0 end as endpoint
        from eligible as a
        left join  WORK.PRIMARY_W4 as b on a.subjid = b.subjid and a.eye = b.eye
        left join  WORK.PRIMARY_M6 as c on a.subjid = c.subjid and a.eye = c.eye
        left join  WORK.PRIMARY_M12 as d on a.subjid = d.subjid and a.eye = d.eye;
quit;

/*customize template for KM plot*/
proc template;
define statgraph Stat.Lifetest.Graphics.ProductLimitFailure;
  dynamic NStrata xName maxTime plotAtRisk plotCensored plotCL plotHW plotEP labelCL labelHW
    labelEP yMin xtickVals xtickValFitPol method StratumID classAtRisk plotTest GroupName
    Transparency rowWeights SecondTitle TestName pValue _byline_ _bytitle_ _byfootnote_;
  BeginGraph;
                discreteattrmap name="symbols" / ignorecase=true ;
                  value "Placebo" / lineattrs=(color=black pattern=dash THICKNESS=1MM) ;
                        value "Fluorometholone" / lineattrs=(color=red pattern=Solid THICKNESS=1.2MM) ;
                enddiscreteattrmap ;
                discreteattrvar attrvar=groupmarkers var=STRATUM attrmap="symbols" ;
                legenditem  type=line name="Placebo" / label='Placebo' lineattrs=(color=black pattern=dash thickness=1MM)
labelattrs=(family="Times New Roman" size=10pt);
                legenditem  type=line name="FML" / label='Fluorometholone' lineattrs=(color=red thickness=1.2MM)
labelattrs=(family="Times New Roman" size=10pt);
    if (NSTRATA=1)
      /*if (EXISTS(STRATUMID))
        entrytitle METHOD " Failure Curve" " for " STRATUMID;
```

```
      else
        entrytitle METHOD " Failure Curve";
      endif;*/
      if (PLOTATRISK=1)
        entrytitle " " / textattrs=GRAPHVALUETEXT;
      endif;
      layout overlay / xaxisopts=(label= "Follow-up time (days)" labelattrs=(family="Times New Roman" size=10pt)
tickvalueattrs=(family="Times New Roman" size=9pt) shortlabel=XNAME offsetmin=.05 linearopts=(viewmax=380
        tickvaluelist=(0 28 168 365 ) /*tickdisplaylist=("Baseline" "Week 4" "Month 6" "Month 12")*/)) yaxisopts=(label=
         "Cumulative rate of postoperative TT" labelattrs=(family="Times New Roman" size=10pt) tickvalueattrs=(family="Times
New Roman" size=9pt) shortlabel="Failure" linearopts=(viewmin=0 viewmax=0.16
        tickvaluelist=(0  .02 .04 .06 .08 .10 .12 .14 .16) tickdisplaylist=('0·00' '0·02' '0·04' '0·06' '0·08' '0·10' '0·12' '0·14' '0·16')));
        if (PLOTHW=1 AND PLOTEP=0)
          bandplot LimitUpper=eval (1-HW_LCL) LimitLower=eval (1-HW_UCL) x=TIME /
            displayTail=false modelname="Failure" fillattrs=GRAPHCONFIDENCE name="HW"
            legendlabel=LABELHW;
        endif;
        if (PLOTHW=0 AND PLOTEP=1)
          bandplot LimitUpper=eval (1-EP_LCL) LimitLower=eval (1-EP_UCL) x=TIME /
            displayTail=false modelname="Failure" fillattrs=GRAPHCONFIDENCE name="EP"
            legendlabel=LABELEP;
        endif;
        if (PLOTHW=1 AND PLOTEP=1)
          bandplot LimitUpper=eval (1-HW_LCL) LimitLower=eval (1-HW_UCL) x=TIME /
            displayTail=false modelname="Failure" fillattrs=GRAPHDATA1 datatransparency=.55
            name="HW" legendlabel=LABELHW;
          bandplot LimitUpper=eval (1-EP_LCL) LimitLower=eval (1-EP_UCL) x=TIME /
```

```
    displayTail=false modelname="Failure" fillattrs=GRAPHDATA2 datatransparency=.55
    name="EP" legendlabel=LABELEP;
endif;
if (PLOTCL=1)
  if (PLOTHW=1 OR PLOTEP=1)
    bandplot LimitUpper=eval (1-SDF_LCL) LimitLower=eval (1-SDF_UCL) x=TIME /
      displayTail=false modelname="Failure" display=(outline) outlineattrs=
      GRAPHPREDICTIONLIMITS name="CL" legendlabel=LABELCL;
  else
    bandplot LimitUpper=eval (1-SDF_LCL) LimitLower=eval (1-SDF_UCL) x=TIME /
      displayTail=false modelname="Failure" fillattrs=GRAPHCONFIDENCE name="CL"
      legendlabel=LABELCL;
  endif;
endif;
stepplot y=eval (1-SURVIVAL) x=TIME / name="Failure" rolename=(_tip1=ATRISK _tip2=
  EVENT) tiplabel=(y="Failure Probability" _tip1="Number at Risk" _tip2=
  "Observed Events") tip=(x y _tip1 _tip2) legendlabel="Failure";
if (PLOTCENSORED)
  scatterplot y=eval (1-CENSORED) x=TIME / tiplabel=(y="Failure Probability")
    markerattrs=(symbol=plus) name="Censored" legendlabel="Censored";
endif;
if (PLOTCL=1 OR PLOTHW=1 OR PLOTEP=1)
  discretelegend "Censored" "CL" "HW" "EP" / location=outside halign=center;
else
  if (PLOTCENSORED=1)
    discretelegend "Censored" / location=inside autoalign=(topleft bottomright);
  endif;
```

```
      endif;
      if (PLOTATRISK=1)
        innermargin / align=bottom;
          axistable x=TATRISK value=ATRISK / display=(label) valueattrs=(size=7pt);
        endinnermargin;
      endif;
    endlayout;
  else
    /*entrytitle METHOD " Failure Curves";
    if (EXISTS(SECONDTITLE))
      entrytitle SECONDTITLE / textattrs=GRAPHVALUETEXT;
    endif;*/
    layout overlay / xaxisopts=(label= "Follow-up time (days)" labelattrs=(family="Times New Roman" size=10pt)
tickvalueattrs=(family="Times New Roman" size=9pt) shortlabel=XNAME offsetmin=.05 linearopts=(viewmax=380
      tickvaluelist=(0 28 168 365 ) /*tickdisplaylist=("Baseline" "Week 4" "Month 6" "Month 12")*/)) yaxisopts=(label=
     "Cumulative rate of postoperative TT" labelattrs=(family="Times New Roman" size=10pt) tickvalueattrs=(family="Times
New Roman" size=9pt) shortlabel="Failure" linearopts=(viewmin=0 viewmax=.16
      tickvaluelist=(0  .02 .04 .06 .08 .10 .12 .14 .16) tickdisplaylist=("0·00" "0·02" "0·04" "0·06" "0·08" "0·10" "0·12" "0·14"
"0·16")));
      if (PLOTHW=1)
        bandplot LimitUpper=eval (1-HW_LCL) LimitLower=eval (1-HW_UCL) x=TIME /
          displayTail=false group=STRATUM index=STRATUMNUM modelname="Failure"
          datatransparency=Transparency;
      endif;
      if (PLOTEP=1)
        bandplot LimitUpper=eval (1-EP_LCL) LimitLower=eval (1-EP_UCL) x=TIME /
          displayTail=false group=STRATUM index=STRATUMNUM modelname="Failure"
```

```
    datatransparency=Transparency;
endif;
if (PLOTCL=1)
  if (PLOTHW=1 OR PLOTEP=1)
    bandplot LimitUpper=eval (1-SDF_LCL) LimitLower=eval (1-SDF_UCL) x=TIME /
      displayTail=false group=STRATUM index=STRATUMNUM modelname="Failure" display
      =(outline) outlineattrs=(pattern=ShortDash);
  else
    bandplot LimitUpper=eval (1-SDF_UCL) LimitLower=eval (1-SDF_LCL) x=TIME /
      displayTail=false group=STRATUM index=STRATUMNUM modelname="Failure"
      datatransparency=Transparency;
  endif;
endif;

stepplot y=eval (1-SURVIVAL) x=TIME / group=groupmarkers index=STRATUMNUM name="Failure"
  rolename=(_tip1=ATRISK _tip2=EVENT) tiplabel=(y="Failure Probability" _tip1=
  "Number at Risk" _tip2="Observed Events") tip=(x y _tip1 _tip2);
if (PLOTCENSORED=1)
  scatterplot y=eval (1-CENSORED) x=TIME / tiplabel=(y="Failure Probability") group=
    STRATUM index=STRATUMNUM markerattrs=(symbol=plus);
endif;
if (PLOTATRISK=1)
  innermargin / align=bottom;
    axistable x=TATRISK value=ATRISK / display=(label) valueattrs=(size=7pt) class=
      CLASSATRISK colorgroup=CLASSATRISK;
  endinnermargin;
endif;
```

```
*DiscreteLegend "Failure" / title=GROUPNAME location=outside;
                    DiscreteLegend "Placebo" "FML" / title='' location=outside;
if (PLOTCENSORED=1)
  if (PLOTTEST)
    layout gridded / rows=2 autoalign=(TOPLEFT BOTTOMRIGHT BOTTOM TOP) border=true
      BackgroundColor=GraphWalls:Color Opaque=true;
      entry "+ Censored";
      if (PVALUE < .0001)
        entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
      else
        entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
      endif;
    endlayout;
  else
    layout gridded / rows=1 autoalign=(TOPLEFT BOTTOMRIGHT BOTTOM TOP) border=true
      BackgroundColor=GraphWalls:Color Opaque=true;
      entry "+ Censored";
    endlayout;
  endif;
else
  if (PLOTTEST=1)
    layout gridded / rows=1 autoalign=(TOPLEFT BOTTOMRIGHT BOTTOM TOP) border=true
      BackgroundColor=GraphWalls:Color Opaque=true;
      if (PVALUE < .0001)
        entry TESTNAME " p " eval (PUT(PVALUE, PVALUE6.4));
      else
        entry TESTNAME " p=" eval (PUT(PVALUE, PVALUE6.4));
```

```
            endif;
          endlayout;
        endif;
      endif;
    endlayout;
  endif;
  if (_BYTITLE_)
    entrytitle _BYLINE_ / textattrs=GRAPHVALUETEXT;
  else
    if (_BYFOOTNOTE_)
      entryfootnote halign=left _BYLINE_;
    endif;
  endif;
 EndGraph;
end;
run;

/*generate KM plot*/
ods _all_ close;
options nodate nonumber;
ods escapechar='~';
ODS NOPROCTITLE;
ods graphics on / reset=all height=106mm width=141mm noborder;
ods select failureplot(persist);
ods pdf file = "&fig_path" DPI=600;
title;
ods pdf text="~S={just=l font_face='Times New Roman' font_size=10pt font_weight= bold} &TLF_title";
```

```
proc lifetest data=endpoint_time plots=survival(NOCENSOR failure);
        time Time * endpoint(0);
        strata treat ;
        format treat trtf.;
run;
ods graphics off;
ods pdf close;
ods exclude none;

proc template;
 delete Stat.Lifetest.Graphics.ProductLimitFailure;
 run;
```