2023.07.10 개인 암 이력 수정 & Leukemia Outcome 대상 counting process format 형성

담당자	은경 이
상태	진행 중
마감일	@2023년 7월 11일
태그	My work

What TO DO)

: 폐암 발생 Outcome 대상, 개인 암 이력 변수 coding 방안 수정 + Cox-ph model 재 적합

: Leukemia(백혈병) 발생 Outcome 대상, 이전과 동일한 방식으로 counting-process format 형성,

개인 암 이력 변수 시간 가변 공변량 형태로 정의 + Cox-ph model 적합

Share & Result)

1) 폐암 발생 Outcome 대상, 개인 암 이력 변수 coding 수정

: 이전에는 "*OUT_DT = fdx1 or ECNY_DT ≥ fdx1*"이면 "*lung_cancer_history*" = 1 부여하였지만, 조건을 "*ECNY_DT ≥ fdx1*"이면 "*lung_cancer_history*" = 1을 가지는 것으로 변화 (시간 가변 공변량 정의 자체)

```
/* MERGE data who have cancer history for lung cancer outcome */
DATA mine.final_history_yes;
SET yes_all check_ID(DROP = between_yes);
BY INDI_ID ECNY_DT OUT_DT;
IF ECNY_DT >= fdx1 then lung_cancer_history=1; /********/
ELSE lung_cancer_history=0;
run;
```

2) 개인 암 이력 변수 수정 뒤, Cox-ph model 재적합

: "*Phreg*" procedure 때 "*COVSANDWICH*" option 추가한 뒤 재 적합

```
%macro split_data_fitted_Cox(input_dataset, output_prefix);
  %do year = 1940  %to 1984 %by 5;
         DATA mine.&output_prefix._&year._M;
         SET &input_dataset(DROP = Outcome_Category Outcome_date);
         WHERE BYEAR between &year and (&year + 4) and SEX = "남";
         run;

      proc phreg data=mine.&output_prefix._&year._M COVSANDWICH FAST;
      class Event(ref='0') Entry_2004(ref='0') Entry_2009(ref='0') Entry_2014(ref='0')
            Entry_2018(ref='0') Lung_Cancer_History(ref='0');
      model (start, stop) * Event(0) = Entry_2004 Entry_2009 Entry_2014 Entry_2018
                                       Lung_Cancer_History /rl;
      run;

      DATA mine.&output_prefix._&year._F;
          SET &input_dataset(DROP = Outcome_Category Outcome_date);
          WHERE BYEAR between &year and (&year + 4) and SEX = "여";
          run;

      proc phreg data=mine.&output_prefix._&year._F COVSANDWICH FAST;
      class Event(ref='0') Entry_2004(ref='0') Entry_2009(ref='0') Entry_2014(ref='0')
            Entry_2018(ref='0') Lung_Cancer_History(ref='0');
      model (start, stop) * Event(0) = Entry_2004 Entry_2009 Entry_2014 Entry_2018
                                       Lung_Cancer_History /rl;
      run;
    %end;
%mend;
```

```
%split_data_fitted_Cox(input_dataset=mine.final_lung_time_varying_NO3, output_prefix=dt);
```

3) 폐암 Outcome일 때와 동일한 방식으로 백혈병 Outcome 대상, “*Outcome_Category*”, “*Outcome_date*”, “*between_yes*” 변수 정의

— ***mine.leukemia_raw*** tbl

```
proc sort data = mine.company_info;
BY INDI_ID ECNY_DT OUT_DT;
run;

proc sort data = mine.N1_raw;
BY INDI_ID;
run;

/* Outcome이 백혈병인 Data generate */
DATA mine.leukemia_raw;
MERGE mine.company_info mine.N1_raw;
BY INDI_ID;
IF leukemia=1 then do;
  Outcome_Category="leukemia_cancer";
  Outcome_date = leukemia_date;
end;
ELSE IF Death=1 then do;
  Outcome_Category="Death";
  Outcome_date = DTH_DATE;
end;
ELSE DO;
  Outcome_Category="last_follow";
  Outcome_date = '20181231';
end;
IF (Outcome_date> ECNY_DT) and (Outcome_date < OUT_DT) then between_yes=1;
ELSE between_yes=0;
DROP DTH_DATE Death lung_cancer lung_cancer_date lung_cancer_history leukemia leukemia_date;
run;
```

4) 이전과 동일한 방식으로 case 나누어 Counting-process format 형성

(가능한 경우)

- “between_yes” = 1 + 반복 측정치가 1개
- 반복 측정치가 여러 개 + “between_yes” = 1인 경우가 존재
- “between_yes” = 0 + 반복 측정치가 1개
- 반복 측정치가 여러 개 + “between_yes” 변수 값이 모두 0인 경우

→ 위 경우로 나누어 counting process format 생성한 후 모두 병합

: ***mine.final_between_yes*** + ***mine.final_between_no*** = ***mine.final_format tbl***

→ 데이터 병합한 후, “*NO*” 변수가 결측인, 즉, 무직 기간에 해당하는 관측치에 대해 사업장 변수 모두 결측으로 처리

→ *(Start, Stop]* 변수 추가(단위는 year) + “*Event*” 변수 추가 정의 + “*NO2*” 변수 정의(obs number)

— ***mine.final_leukemia_NO2*** tbl

```
/* "leukemia" tbl check */
proc sql;
create table check1 as select distinct Outcome_Category from mine.leukemia_raw
where between_yes=1;
quit;

proc sql;
create table check2 as select INDI_ID, ECNY_DT, OUT_DT, Outcome_Category, Outcome_date from mine.leukemia_raw
where INDI_ID not in (select distinct INDI_ID from mine.leukemia_raw where between_yes=1) and Outcome_Category='leukemia_cancer';
quit;
```

```
/** Counting-process format data generate **/
/* Get all INDI_ID's record who have "between_yes=1" */
proc sql;
create table between_yes_all as select * from mine.leukemia_raw
where INDI_ID in (select distinct INDI_ID from mine.leukemia_raw where between_yes=1);
quit;

proc sort data = between_yes_all;
BY INDI_ID ECNY_DT OUT_DT;
run;

/* first.INDI_ID + between_yes=1 */
DATA first_yes;
SET between_yes_all;
BY INDI_ID ECNY_DT OUT_DT;
IF first.INDI_ID and between_yes=1;
run;

DATA temp_yes last_yes;
SET first_yes;
start = ECNY_DT;
stop = Outcome_date;
Output last_yes;
DROP ECNY_DT OUT_DT between_yes;
RENAME start = ECNY_DT stop = OUT_DT;
run;

/* between_yes=1 & have multiple records for each INDI_ID */
proc sql;
create table between_yes_much as select * from between_yes_all
where INDI_ID not in (select distinct INDI_ID from first_yes);
quit;

proc sort data = between_yes_much;
by INDI_ID ECNY_DT OUT_DT;
run;

DATA temp2 last;
SET between_yes_much;
BY INDI_ID ECNY_DT OUT_DT;
start = lag(OUT_DT);
stop=ECNY_DT;
NO2=put('',8.);

IF first.INDI_ID and between_yes=0 then do;
start = ECNY_DT;
stop = OUT_DT;
NO2=NO;
end;
Output temp2;

IF between_yes=1 then do;
  start = ECNY_DT;
  stop = Outcome_date;
  NO2=NO;
  Output last;
end;
DROP ECNY_DT OUT_DT NO between_yes;
RENAME NO2 = NO start = ECNY_DT stop = OUT_DT;
run;

DATA mine.final_between_yes;
SET between_yes_all last_yes temp2 last;
IF ECNY_DT = OUT_DT and NO = '' then delete;
IF OUT_DT > Outcome_date then delete;
DROP NEW_NY between_yes;
run;

proc sort data=mine.final_between_yes noduprecs;
BY INDI_ID ECNY_DT OUT_DT;
run;

/* Get all INDI_ID's record who don't have "between_yes=1" */
proc sql;
create table between_no_all as select * from mine.leukemia_raw
where INDI_ID not in (select distinct INDI_ID from between_yes_all);
quit;

proc sort data = between_no_all;
by INDI_ID ECNY_DT OUT_DT;
run;

/* Get all INDI_ID's record who have only just one replication */
```

```
proc sql;
create table counts as select INDI_ID, count(*) as obs_count  from between_no_all
group by INDI_ID;
quit;

/* between_yes=0 & first.INDI_ID */
DATA first_no;
MERGE between_no_all counts;
BY INDI_ID;
IF obs_count = 1;
DROP between_yes obs_count;
run;

DATA temp_no last_no;
SET first_no;
start = OUT_DT;
stop = Outcome_date;
NO2=put('',8.);;
Output last_no;
DROP ECNY_DT OUT_DT NO;
*DROP Outcome_date;
RENAME NO2 = NO start = ECNY_DT stop = OUT_DT;
run;

DATA no_first;
SET first_no last_no;
IF ECNY_DT = OUT_DT and NO = '' then delete;
run;

proc sort data=no_first noduprecs;
BY INDI_ID ECNY_DT OUT_DT;
run;

/* between_yes=0 & have much replication records */
proc sql;
create table between_no_much as select * from between_no_all
where INDI_ID not in (select distinct INDI_ID from first_no);
quit;

proc sort data = between_no_much;
by INDI_ID ECNY_DT OUT_DT;
run;

DATA temp2 last;
SET between_no_much;
BY INDI_ID ECNY_DT OUT_DT;
start = lag(OUT_DT);
stop=ECNY_DT;
NO2=put('',8.);
IF first.INDI_ID then do;
start = ECNY_DT;
stop = OUT_DT;
NO2=NO;
end;
Output temp2;
IF last.INDI_ID then do;
  start = OUT_DT;
  stop = Outcome_date;
  NO2=put('',8.);
  Output last;
end;
DROP NO ECNY_DT OUT_DT between_yes;
RENAME NO2 = NO start = ECNY_DT stop = OUT_DT;
run;

DATA mine.final_between_no;
SET between_no_much(drop = between_yes) no_first temp2 last;
IF ECNY_DT = OUT_DT and NO = '' then delete;
IF OUT_DT > Outcome_date then delete;
DROP NEW_NY;
run;

proc sort data = mine.final_between_no noduprecs;
by INDI_ID ECNY_DT OUT_DT;
run;

/* Data rbind */
DATA mine.final_format;
SET mine.final_between_yes mine.final_between_no;
IF NO='' then do;
  OUT_CZ='';
  BIZ_INDUTY10='';
  JSSFC_CD='';
  JSSFC_NO='';
```

```
  INDDIS_NO='';
  ENROL_NO='';
  BIZ_NM='';
  BIZ_NO='';
  BIZ_ADDRESS='';
  BIZ_ZIP='';
end;
IF OUT_DT = Outcome_date and Outcome_Category="leukemia_cancer" then Event=1;
ELSE Event=0;
RUN;

proc sort data = mine.final_format;
BY INDI_ID ECNY_DT OUT_DT;
run;

DATA mine.final_leukemia_NO2;
SET mine.final_format;
BY INDI_ID ECNY_DT OUT_DT;
No2 = _n_;
RETAIN first;
IF first.INDI_ID then first = ECNY_DT;
start = (input(ECNY_DT, yymmdd8.) - input(first,yymmdd8.))/365.25;
stop = (input(OUT_DT, yymmdd8.) - input(first,yymmdd8.))/365.25;
DROP first;
run;
```

5) 백혈병 발병 Outcome 대상, 개인 암 이력을 시간 가변 공변량으로 변환 + counting process format 재 형성

: 개인 암 이력이 존재하는 객체("*leukemia_cancer_history*" = 1)의 기록 가져와 "*between_yes"* 변수 이전과 동일한 방식으로 생성

: "*between_yes*" 변수 이용해 이전과 동일한 형태로 case 나눔

(가능한 경우)

- "*between_yes*" = 1인 관측치가 존재하는 경우
- 개인 암 이력은 있지만 모든 반복 측정치에 대해 "*between_yes*" = 0인 경우
- 개인 암 이력이 존재하지 않는 경우

→ 경우에 따라 나누어 counting process format 다시 변환한 뒤, 병합 + *(Start, Stop]* 변수 추가 + "*NO3*" 변수 (각 obs의 unique key / obs number로 정의) 추가 — ***mine.final_leukemia_time_varying_NO3*** tbl

```
/* Get distinct INDI_ID who have cancer history for leukemia cancer outcome */
proc sql;
create table temp as select distinct INDI_ID from mine.final_leukemia_no2
where leukemia_cancer_history=1;
quit;

/* Get record who is in temp tbl */
proc sql;
create table check_cancer as select distinct INDI_ID, fdx1 from mine.My_raw
where INDI_ID in (select * from temp);
quit;

/* joint counting-process format for leukemia cancer outcome tbl and check_cancer tbl */
DATA final_leukemia_no2;
SET mine.final_leukemia_no2;
DROP leukemia_cancer_history start stop;
RUN;

proc sql;
create table mine.temp2 as select * from final_leukemia_no2 as a join check_cancer as b
on a.INDI_ID = b.INDI_ID;
quit;

/* Define "between_yes" variable */
DATA temp3;
SET mine.temp2;
IF (fdx1> ECNY_DT) and (fdx1 < OUT_DT) then between_yes=1;
ELSE between_yes=0;
run;

/* Get records who have "between_yes" = 1*/
proc sql;
```

```
create table yes_tbl as select * from temp3
where INDI_ID in (select distinct INDI_ID from temp3 where between_yes=1);
quit;

/* Plus the obs cancer history for leukemia_cancer outcome */
DATA yes_1;
SET yes_tbl;
IF between_yes=1;
run;

DATA first last;
SET yes_1;
ECNY_DT2 = ECNY_DT;
OUT_DT2 = fdx1;
DROP ECNY_DT OUT_DT;
RENAME ECNY_DT2 = ECNY_DT OUT_DT2 = OUT_DT;
Output first;
ECNY_DT2 = fdx1;
OUT_DT2 = OUT_DT;
DROP ECNY_DT OUT_DT;
RENAME ECNY_DT2 = ECNY_DT OUT_DT2 = OUT_DT;
Output last;
run;

proc sql;
create table except_yes as select * from yes_tbl
where NO2 not in (select NO2 from yes_1);
quit;

DATA yes_all;
SET except_yes first last;
DROP between_yes;
run;

proc sort data = yes_all;
BY INDI_ID ECNY_DT OUT_DT;
run;

/* Get all records who don't have "between_yes" = 1 */
proc sql;
create table check_ID as select * from temp3
where INDI_ID not in (select distinct INDI_ID from yes_tbl);
quit;

/* MERGE data who have cancer history for leukemia cancer outcome */
DATA mine.final_history_yes;
SET yes_all check_ID(DROP = between_yes);
IF ECNY_DT >= fdx1 then leukemia_cancer_history=1;
ELSE leukemia_cancer_history=0;
run;

/* MERGE all data */
proc sql;
create table final_history_not as select * from mine.final_leukemia_no2
where INDI_ID not in (select distinct INDI_ID from mine.final_history_yes);
quit;

DATA final_history_not2;
SET final_history_not;
leukemia_cancer_history=0;
run;

DATA mine.final_time_varying_cancer;
SET mine.final_history_yes(DROP = fdx1) final_history_not(DROP = start stop);
run;

proc sort data = mine.final_time_varying_cancer;
BY INDI_ID ECNY_DT OUT_DT;
run;

/* Add (Start, Stop] and "NO3" variable */
DATA mine.final_leukemia_time_varying_NO3;
SET mine.final_time_varying_cancer;
BY INDI_ID ECNY_DT OUT_DT;
RETAIN first;
IF first.INDI_ID then first = ECNY_DT;
start = (input(ECNY_DT, yymmdd8.) - input(first, yymmdd8.))/365.25;
stop = (input(OUT_DT, yymmdd8.) - input(first, yymmdd8.))/365.25;
NO3 = _n_;
DROP first;
run;
```

6) 백혈병 발생이 Outcome인 데이터 대상, N1 model(개인 암 이력은 시간 가변 공변량 형태) Cox-ph 적합

: “*COVSANDWICH*” option 추가

```
%macro split_data_fitted_Cox(input_dataset, output_prefix);
  %do year = 1940  %to 1984 %by 5;
         DATA mine.&output_prefix._&year._M;
         SET &input_dataset(DROP = Outcome_Category Outcome_date);
         WHERE BYEAR between &year and (&year + 4) and SEX = "남";
         run;

      proc phreg data=mine.&output_prefix._&year._M COVSANDWICH FAST;
      class Event(ref='0') Entry_2004(ref='0') Entry_2009(ref='0') Entry_2014(ref='0')
            Entry_2018(ref='0') Leukemia_Cancer_History(ref='0');
      model (start, stop) * Event(0) = Entry_2004 Entry_2009 Entry_2014 Entry_2018
                                       Leukemia_Cancer_History /rl;
      run;

      DATA mine.&output_prefix._&year._F;
          SET &input_dataset(DROP = Outcome_Category Outcome_date);
          WHERE BYEAR between &year and (&year + 4) and SEX = "여";
          run;

      proc phreg data=mine.&output_prefix._&year._F COVSANDWICH FAST;
      class Event(ref='0') Entry_2004(ref='0') Entry_2009(ref='0') Entry_2014(ref='0')
            Entry_2018(ref='0') Leukemia_Cancer_History(ref='0');
      model (start, stop) * Event(0) = Entry_2004 Entry_2009 Entry_2014 Entry_2018
                                       Leukemia_Cancer_History /rl;
      run;
    %end;
%mend;

%split_data_fitted_Cox(input_dataset=mine.final_Leukemia_time_varying_NO3,
                       output_prefix=dt2);
```