


2023.07.07 개인 암 이력 time-varying covariate 변환

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

What TO DO)

: 폐암 발생 Outcome 대상, time-invariant covariate에서 “time-varying” covariate로 변환
→ 데이터도 counting-process format 재 형성

Share & Result

1) 이전에 생성한 “*mine.final_lung_plus_no2*” tbl에서 *lung_cancer_history=1*, 즉, 개인 암 이력이 존재하는 객체의 distinct INDI_ID와 첫 암 진단 일자(fdx1)가져오기

: 해당 객체는 933,944명임.

```
/* Get distinct INDI_ID who have cancer history for lung cancer outcome */
proc sql;
create table temp as select distinct INDI_ID from mine.final_lung_plus_no2
where lung_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;
```

2) Cancer history가 있는 객체들의 모든 반복 측정치 가져오기

: 기존에 생성한 폐암 발생이 Outcome인 counting-process format과 “*check_cancer*” tbl joint

```

/* joint counting-process format for lung cancer outcome tbl and check_cancer tbl */
DATA final_lung_no2;
SET mine.final_lung_plus_no2;
DROP lung_cancer_history start stop;
RUN;

proc sql;
create table mine.temp2 as select * from final_lung_no2 as a join check_cancer as b
on a.INDI_ID = b.INDI_ID;
quit;

```

3) “*between_yes*” 변수(각 반복 측정치의 (취득일, 상실일) 사이에 암 진단 일자가 포함되는지 나타내는 변수) 정의

: 개인 암 이력이 존재하는 객체들 만을 대상으로 정의함.(*temp2* tbl)

: 이전과 동일한 방식으로 진행

```

/* 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;

```

4) “*between_yes*”=1인 반복 측정치를 가지고 counting-process format으로 변환

4)-1. “*between_yes*”=1인 반복 측정치를 아래 그림과 같은 형태로 tbl을 따로 생성함

→ “*first*”, “*last*” tbl

INDI_ID	ECNY_DT	OUT_DT	fdx1
100	20050101	20150201	20100715



INDI_ID	ECNY_DT	OUT_DT	fdx1
100	20050101	20100715	20100715
100	20100715	20150201	20100715

```

/* 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 lung_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;

```

4)-2 “between_yes”=1인 측정치 삭제(주황색 tbl)하고 4)-1에서 생성한 tbl(노란색 tbl)을 기존 tbl과 rbind

: unique key “NO2” 변수 이용

(삭제해야 하는 행)

INDI_ID	ECNY_DT	OUT_DT	fdx1
100	20050101	20150201	20100715

(새로 합쳐야 하는 행)

INDI_ID	ECNY_DT	OUT_DT	fdx1
100	20050101	20100715	20100715
100	20100715	20150201	20100715

```

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;

```

5) 개인 암 이력이 존재하나 모든 반복 측정치가 “between_yes” = 0인 객체 (즉, (취득일, 상실일) 사이에 암 진단 일자가 존재하지 않는 case) 기록 가져오기

(존재하는 case)

- 사망 일자와 암 진단 일자가 동일한 경우
- “OUT_DT”와 암 진단 일자가 동일한 경우

취득일	상실일	INDI_ID	No2	fdx1
20060501	20120209	1000000760021	12131	20170506
20120209	20170506	1000000760021	12132	20170506
19950701	20000118	1000000763428	25705	20111003
20000118	20040101	1000000763428	25706	20111003
20040101	20041112	1000000763428	25707	20111003
20041112	20080501	1000000763428	25708	20111003
20080501	20081230	1000000763428	25709	20111003
20081230	20090101	1000000763428	25710	20111003
20090101	20111003	1000000763428	25711	20111003
20111003	20181231	1000000763428	25712	20111003

- last follow up date(2018.12.31)와 암 진단 일자가 동일한 경우

취득일	상실일	INDI_ID	No2	fdx1
19950701	19970801	1000000770610	54821	20181231
19970801	20181231	1000000770610	54822	20181231

→ 해당 case는 굳이 행을 변환할 필요가 없다고 판단함.

```

/* 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;
```

6) 개인 암 이력이 존재하는 객체 전부 대상, counting-process format으로 변환한 tbl 모두 rbind + time-varying “lung_cancer_history” covariate 변수 재정의

: time-varying “lung_cancer_history” covariate는 첫 암 진단 일자 이후 모두 1의 값을 가짐.

— “final_history_yes” tbl

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

7) 개인 암 이력이 아예 없는 객체들의 자료와 6)에서 생성한 “final_history_yes” tbl rbind

: 개인 암 이력이 없는 객체들에 대해서는 “lung_cancer_history” 변수 값이 모두 0이 되도록 생성

— “mine.final_time_varying_cancer” tbl

```
/* MERGE all data */
proc sql;
create table final_history_not as select * from mine.final_lung_plus_no2
where INDI_ID not in (select distinct INDI_ID from mine.final_history_yes);
quit;

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

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

8) 재 변환한 counting-process format data에 “(Start, Stop]” 변수 추가 + unique key “NO3” 변수 추가 생성

: “(Start, Stop]”는 단위가 year이 되도록 생성 + “NO3” 변수는 obs number로 정의

— 최종 table : “*mine.final_lung_time_varying_NO3*”

```
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_lung_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;
```