


# 2023.07.03 Counting process format 변환

담당자	 은경 이
상태	완료
마감일	@2023년 7월 4일
태그	My work

## What TO DO)

: 폐암 Outcome을 대상으로 “Outcome Category”, “Outcome date”, “between\_yes” 변수 추가 생성

: Outcome이 폐암으로 설정한 데이터를 Counting-process format으로 변환

## Share & Result)

### 1) Outcome이 폐암인 반복 측정 자료 생성

: (Demographic Info + 최초 고용 보험 등록일 + Outcome) 정보만 있는 “N1\_Raw” tbl과 사업장 정보 가 담긴 “Company\_Info” tbl을 ‘INDI\_ID’ 기준으로 MERGE

: 추가로 “*Outcome\_Category*”(범주가 lung\_cancer / Death / Last\_follow인 변수), “*Outcome\_date*” (Outcome 종류 별 특정 event가 발생한 날짜), “*between\_yes*”(각 반복 측정치의 [취득일, 상실일] 사이에 Outcome\_date가 속하는지의 여부)

→ *mine.lung\_raw* tbl

```
/* 사업장 정보만 담긴 tbl get */
DATA mine.company_info;
SET dir.Preproc1_new;
DROP fdx1-fdx6 tcode1-tcode6 mcode1-mcode6 method1-method6 icd10_1-icd10_6
seer_grp1-seer_grp6 REAL_AUTH_CODE1 DTH_DATE1-DTH_DATE3 BYEAR BMONTH BDAY AGE DURATION;
run;

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 폐암인 Data generate */
DATA mine.lung_raw;
MERGE mine.company_info mine.N1_raw;
BY INDI_ID;
IF lung_cancer=1 then do;
    Outcome_Category="lung_cancer";
    Outcome_date = lung_cancer_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 leukemia leukemia_date
Leukemia_Cancer_History;
run;

```

## 2) 경우에 따라 나누어 Counting-process format 생성

2)-1 “between\_yes” = 1 + 반복 측정치가 1개 (first\_yes tbl)

: “first\_yes” tbl을 counting-process format으로 변환 (last\_yes tbl)



```

/** 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.lung_raw
where INDI_ID in (select distinct INDI_ID from mine.lung_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;

```

**2)-2** 반복 측정치가 여러 개 + “between\_yes” = 1인 경우가 존재(between\_yes\_much tbl)

: “between\_yes\_much” tbl을 counting-process format으로 변환(temp2 / last tbl)

(Raw data)

ID	NO	ECNY_DT	OUT_DT	Outcome date
1	1	20070607	20070922	20150120
1	2	20100701	20100918	20150120
1	3	20110103	20110319	20150120
1	4	20110404	20120930	20150120
1	5	20121001	20131113	20150120
1	6	20131113	20150129	20150120

(Output 1)

ID	NO	ECNY_DT	OUT_DT
1		20070922	20100701
1		20100918	20110103
1		20110319	20110404
1		20120930	20121001
1		20131113	20131113

(Output 2)

ID	NO	ECNY_DT	OUT_DT
1	6	20131113	20150120

: 위 그림 형태로 데이터를 rbind 한 후, “OUT\_DT > Outcome\_date” 혹은 “ECNY\_DT = OUT\_DT” and “NO=” ’ “인 관측치는 삭제 진행

```

/* 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;

```

\* 2)-1 + 2)-2 rbind 통해 “between\_yes” = 1 관측치를 가지는 객체 전부 대상으로 counting-process format 형성 → *mine.final\_between\_yes* tbl

```

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;

```

**2)-3** “between\_yes” = 0 + 반복 측정치가 1개(*first\_no* tbl)

: *first\_no* tbl을 counting-process format으로 변환(*no\_first* tbl)

NO	ECNT_DT	OUT_DT	Outcome_date
1	20150123	20161201	20181231



NO	ECNT_DT	OUT_DT
1	20150123	20161201
	20161201	20181231

```

/* Get all INDI_ID's record who don't have "between_yes=1" */
proc sql;
create table between_no_all as select * from mine.lung_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;
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;

```

**2)-4** 반복 측정치가 여러 개 + 모든 측정치가 “*between\_yes*” = 0인 경우  
(*between\_no\_much* tbl)

: *between\_no\_much* tbl을 counting-process format으로 변환(*temp2*, *last* tbl)

(Raw data)

ID	NO	ECNY_DT	OUT_DT	Outcome date
1	1	20070607	20070922	20150120
1	2	20100701	20100918	20150120
1	3	20110103	20110319	20150120
1	4	20110404	20120930	20150120
1	5	20121001	20131113	20150120
1	6	20131113	20150129	20181231

(Output 1)

ID	NO	ECNY_DT	OUT_DT
1		20070922	20100701
1		20100918	20110103
1		20110319	20110404
1		20120930	20121001
1		20131113	20131113



(Output 2)

ID	NO	ECNY_DT	OUT_DT
1		20150129	20181231

: 위 그림 형태로 데이터를 rbind 한 후, “ECNY\_DT = OUT\_DT” and “NO=’ ’”인 관측치는 삭제 진행

```
/* 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;
```

\* 2)-3 + 2)-4 rbind 통해 모든 반복 측정치가 “between\_yes” = 0인 객체 전부 대상으로 counting-process format 형성 → **mine.final\_between\_no** tbl

```

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;

```

### 3) 2) 과정 통해 생성한 데이터 모두 병합 + [Start, Stop) 형태 변환

: “NO”가 결측인 관측치(무직 기간)의 사업장 정보 변수는 모두 결측으로 대체

: “Outcome\_Category” = ‘lung\_cancer’ 이면서 Stop 시점이 Outcome\_date와 동일한 경우 “Event” 변수 값에 1 부여

→ **mine.final\_lung** tbl

```

/* 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="lung_cancer" then Event=1;
ELSE Event=0;
RUN;

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

/* [Start, Stop) 형태 변환 */
DATA mine.final_lung;
SET mine.final_format;
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.);
stop = input(OUT_DT, yymmdd8.) - input(first, yymmdd8.);

```

```
DROP first;  
run;
```