

## <당뇨병 망막증 진행 사항 정리>

작성자 : 이은경

11월 29일 Version

### <What To Do>

: “AMD DR Pts A1c Rlts” excel file과 “Birth” sas7bdat file joint  
-- joint key는 “MRN” 변수

### <Result>

- 1) “AMD DR Pts A1c Rlts” excel file의 unique PERSON의 수는 266명  
-- 데이터 총 관측치 수는 3743개  
-- sas에 import 할 때 “A1c”로 명명

#### [“AMD DR Pts A1c Rlts” excel file import 결과]

mrn	Date of baseline exam	Date of follow up exam	A1c Create TimeStamp	A1c Result	F
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	3/7/2011 11:11 AM	7.2	1
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/19/2011 3:28 PM	6.1	2
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/29/2011 12:36 AM	7.1	3
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	1/13/2012 2:06 PM	7.3	4
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/18/2012 10:59 AM	6.2	5
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	10/19/2012 10:26 AM	7.3	6
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	1/24/2013 3:29 PM	7.2	7
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	4/4/2013 11:08 AM	7.2	8
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/3/2013 2:25 PM	6.8	9
041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	10/2/2013 11:43 AM	6.7	10

: 날짜 변수 형식 맞춰서 잘 들어온 것 확인.

- 2) “Birth” sas table에 담겨있는 총 Patient 수는 401명

- 3) “A1c” table과 “Birth” table joint (joint key는 “MRN”)  
-- joint 결과 관측치 수가 3728개로 줄어듦.  
--> 이는 “Birth” table에 MRN이 “187995”인 개체의 기록이 없어서 발생한 것으로 파악됨.  
-- “Birth\_A1c” tbl / 후에 지속적으로 필요할 것 같아 table 따로 저장

#### [Birth\_A1c table 생성 결과]

PatientID	mrn	Date of baseline exam	Date of follow up exam	A1c Create TimeStamp	A1c Result	F
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	3/7/2011 11:11 AM	7.2	1
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/19/2011 3:28 PM	6.1	2
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/29/2011 12:36 AM	7.1	3
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	1/13/2012 2:06 PM	7.3	4
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/18/2012 10:59 AM	6.2	5
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	10/19/2012 10:26 AM	7.3	6
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	1/24/2013 3:29 PM	7.2	7
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	4/4/2013 11:08 AM	7.2	8
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	7/3/2013 2:25 PM	6.8	9
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	10/2/2013 11:43 AM	6.7	10
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	2/12/2014 12:55 PM	6.8	11
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	6/12/2014 11:32 AM	7	12
ROHA	041262	1/28/2011 12:00 AM	2/2/2017 12:00 AM	10/2/2014 2:41 PM	6	13

## 12월 6일 Version

### <What To Do>

- : 지속적으로 사용하는 table 따로 저장
- : 기존의 “all2” table과 “Birth\_A1c” table joint
  - joint key는 “Patient\_ID” 변수
- : Data check

### <Way & Result>

0. “all2” table에는 기록이 존재하는데 “Birth\_A1c” table에는 기록이 없는 객체들 존재하는지 check  
: Patient\_ID가 “CUJOH”, “PAMAR”인 객체

1. “all2” table과 “Birth\_A1c” table joint(joint key는 “Patient\_ID”)

-- “data\_ver2” tbl

--> “data\_ver2” tbl에 존재하는 기존 변수 “FU\_Duration”, A1c 관련 변수들 모두 제거

-- “data” tbl

check1) “data” tbl에 “Patient\_ID”와 “Replication”이 동일한 관측치 존재

-- 2개의 eye에 대해 각각 따로 측정했기 때문

2. ‘PROC IML’ 이용해 “FU\_Duration” 변수 재정의 --- “mydata” tbl

2-0. 기존의 날짜 변수들을 모두 실수로 변환 (계산의 용이성 위해)

2-①. “Patient\_ID”와 “Replication”이 동일한 관측치(2개의 eye)의 경우 “FU\_Duration” 동일한 값 input

2-②. “Replication”이 1인, 즉, 첫 번째 반복측정자료의 경우

“FU\_Duration” = (“A1c\_Create\_TimeStamp” - “Date\_of\_Baseline\_exam”)/365.25 로 정의

2-③. 나머지 반복측정자료의 경우 “FU\_Duration” = {(현재 시점의 “A1c\_Create\_TimeStamp”) - (이전 시점의 “A1c\_Create\_TimeStamp”)/365.25 로 정의

### [mydata 생성 결과]

Date of baseline exam_2	Date of follow up exam_2	A1c Create TimeStamp_2	A1c Result	Replication	FU_Duration
38156	43196,541667	38170,64713	7,4	1	0,0401016562
38156	43196,541667	38170,64713	7,4	1	0,0401016562
38156	43196,541667	38299,474513	8,3	2	0,3527101513
38156	43196,541667	38299,474513	8,3	2	0,3527101513
38156	43196,541667	38427,698557	9	3	0,3510583006
38156	43196,541667	38427,698557	9	3	0,3510583006
38156	43196,541667	38495,709654	9,5	4	0,1862042365
38156	43196,541667	38495,709654	9,5	4	0,1862042365
38156	43196,541667	38547,808515	8	5	0,1426389076
38156	43196,541667	38547,808515	8	5	0,1426389076
38156	43196,541667	38642,758677	7,7	6	0,2599593753
38156	43196,541667	38642,758677	7,7	6	0,2599593753
38156	43196,541667	38721,738597	9,5	7	0,216235236
38156	43196,541667	38721,738597	9,5	7	0,216235236

## 12월 9일 Version

NOTE) 12월 6일 Version을 수정 + 추가.

### <What To Do>

#### 1) "FU\_Duration" 변수 수정

[CODING 방안]

Date of Baseline exam을 " $t_0$ "라 하면,

ID	Replication	FU_Duration	A1c	Event
1	1	$t_2 - t_1$	$z_1$	0
.	2	.	.	.
.	.	.	.	.
.	.	.	.	.
1	T	$t_{T+1} - t_T$	$z_T$	0/1

Date of Follow-up exam

last A1c Create TimeStamp

Event 발생한 사람의 경우, event 발생 시점이 last follow up time 이다.

Last 반복관찰치 coding 방법이 바뀜!

#### 2) Event 관련 변수들 재정의("bad", "prog", "pdr")

ex) bad = 1 / prog = 0 / pdr = 1인 경우 해당 객체의 반복측정자료 중 마지막 관측치

전까지는 세 변수 값이 모두 0이었다가 마지막 record에 bad = 1 / prog = 0 / pdr = 1로 coding

-- 단,  $t_{T+1} = t_T$  (즉, day가 같은 날)인 사람들의 경우 마지막 Replication obs는 제거해야 한다.

### <Way & Result>

#### 1. Grouping & PROC IML 이용해 "FU\_Duration" 변수 재정의

##### 1-①. "Patient\_ID", "Eye" 변수로 grouping ("Eye" 변수가 한 개체 당 "OD", "OS"로 나뉨)

-- 이때, last.Eye인 경우

"FU\_Duration" = ("Date\_of\_follow\_up\_exam" - "A1c\_Create\_TimeStamp")/365.25

##### 1-②. 1-① 과정에서 정의된 관측치 제외, 나머지 관측치들에 대해 "i번째 FU\_Duration"를

("(i+1)번째 A1c\_Create\_TimeStamp" - "i번째 A1c\_Create\_TimeStamp")/365.25로 정의

-- PROC IML 이용

-- 결과, 변수가 "baseline exam date", "A1c\_Create\_TimeStamp",

"Replication", "FU\_Duration"만 남음.

-- "data\_FU\_Duration" tbl

##### 1-③. 개체들의 반복 관측치가 담겨 있는 "data" tbl과 "data\_FU\_Duration" tbl joint

-- "mydata" tbl

## 2. Grouping & PROC SQL 이용해 Event 관련 변수("bad", "prog", "pdr") 재정의

2-①. "Patient\_ID", "Eye" 별로 grouping 한 후, "bad", "prog", "pdr" 변수 값 합함.

-- 변수명 "bad\_check", "prog\_check", "pdr\_check"로 정의

2-②. "Patient\_ID", "Eye", "bad", "prog", "pdr" 변수 기준으로 grouping 한 후,

```
IF "bad_check" >= 1 & last.Eye then bad=1 ; ELSE bad=0;
IF "prog_check" >= 1 & last.Eye then prog=1 ; ELSE prog=0;
IF "pdr_check" >= 1 & last.Eye then pdr=1 ; ELSE pdr=0;
```

: 즉, event가 발생한 객체의 경우, 마지막 반복측정관측치 값만 "1"로 정의

-- "newmydata" tbl

## 3. "FU\_Duration" 값이 음수인 관측치(day는 동일하나, 시간이 다른 관측치들임) 삭제

3-①. "Patient\_ID", "Eye"로 grouping 한 후, 마지막 반복관측자료이면서 "FU\_Duration"( $t_{T+1} = t_T$ )이

음수인 관측치 삭제 / 제대로 삭제되었는지 확인하는 절차도 거침.

∴ "FU\_Duration", "bad", "prog", "pdr"변수들 재정의, 정리한 최종 data -- "newdata" tbl

Patient_ID	Eye	bad	prog	pdr	Baseline_date	Follow-up date	A1c_Create	TimeStamp	Repl	FU_Duration
CAJO	OD	0	0	0	39423.395833	42822.642072	39434.426428		1	0.386001
CAJO	OD	0	0	0	39423.395833	42822.642072	39575.413335		2	0.575104
CAJO	OD	0	0	0	39423.395833	42822.642072	39785.470201		3	0.290338
CAJO	OD	1	0	1	39423.395833	42822.642072	39891.51609		4	8.024986
CAJO	OS	0	0	0	39423.395833	42822.642072	39434.426428		1	0.386001
CAJO	OS	0	0	0	39423.395833	42822.642072	39575.413335		2	0.575104
CAJO	OS	0	0	0	39423.395833	42822.642072	39785.470201		3	0.290338
CAJO	OS	1	0	1	39423.395833	42822.642072	39891.51609		4	8.024986

## 12월 15일 Version

NOTE) 12월 9일 Version을 수정 + 추가.

### <What TO DO>

- 1) 기존 “data” tbl(“all2” tbl과 “Birth\_A1c” tbl joint + 기존 A1c 관련 변수 제거)에서 “FU\_Duration” 변수 / Event 관련 변수 재생성
- 2) “FU\_Duration” 변수가 0인 관측치 삭제 & Replication number 조정
- 3) “FU\_Duration” 변수가 -1과 0 사이(즉, “A1c\_Create\_TimeStamp” > “Date\_of\_follow\_up”인데 같은 날이고 시간만 다른 경우)인 관측치 확인 후, 제거

---

### <Way & Result>

#### 1. “FU\_Duration” 변수 / Event 관련 변수(“bad”, “prog”, “pdr”) 재생성

- 1-①. 기존 “data” tbl에서 각 Patient\_ID마다 unique number ID 부여(“ID”가 변수명)  
& Eye 범주와 대응되는 numerical value 가지는 변수 생성 (“Eye\_num”이 변수명)  
-- “data\_ver0” tbl
- 1-②. 기존의 FU\_Duration 변수 생성 규칙 이용해 변수 생성  
-- “data\_FU\_Duration” tbl
- 1-③. “data2” tbl(“data” tbl에서 마지막 반복 측정치의 FU\_Duration 변수 값만 정의된 tbl)와  
“data\_FU\_Duration” tbl을 “ID”, “Replication”, “Eye\_num” 기준으로 joint  
--- “mydata\_ver0” tbl
- 1-④. 기존에 Event 관련 변수 정리하는 규칙 이용해 변수들 정리  
--- “newmydata” tbl

#### 2. “FU\_Duration” 변수가 0인 관측치 삭제 & Replication number 조정

- 2-①. “newmydata” tbl에 Patient\_ID, Eye당 반복측정된 총횟수(“N\_count”가 변수명),  
(총 반복측정횟수 - 관측치 반복측정횟수) 뜻하는 변수(“ow\_n”이 변수명) 생성  
--- “new\_add\_N” tbl

- 2-②. PROC IML 이용해 Replication number 변경 --- “new” tbl

#### **PROC IML;**

```
USE new_add_N var {ID Eye_num Date_of_baseline_exam_2 A1c_Create_TimeStamp_2 Replication Replication2 N_count ow_n FU_Duration};  
READ ALL into result2[colname=varNames];
```

```
DO i=1 TO NROW(result2);  
  IF (result2[i,"FU_Duration"] = 0) & (result2[i,"ID"] = result2[(i+1),"ID"]) & (result2[i,"Eye_num"] = result2[(i+1),"Eye_num"]) THEN do;  
    result2[(i+1), "Replication2"] = result2[i, "Replication2"];  
    result2[(i+2):(i+1+result2[(i+1),"ow_n"]), "Replication2"] = result2[(i+2):(i+1+result2[(i+1),"ow_n"]), "Replication"] - 1;  
  END;  
END;  
create newdata_ver1_2 FROM result2[colname=varNames];  
APPEND FROM result2;  
CLOSE new_add_N;  
QUIT;
```

---

3. “FU\_Duration” 변수가 -1과 0 사이(즉, “A1c\_Create\_TimeStamp” > “Date\_of\_follow\_up”인데 같은 날이고 시간만 다른 경우)인 관측치 확인 후, 제거  
: 마지막으로, “FU\_Duration” 변수 값을 365.25로 나누어 scale 조정함.  
-- “mynewdata” tbl (저장함.)

---

## 12월 19일 Version

NOTE) 12월 15일 Version에서 “FU\_Duration” 변수 정의, Event 관련 변수 정의하는 코드 순서 변경

### 〈What TO DO〉

- 1) 기존 code에서 “FU\_Duration” 변수 생성 & 정의하는 code, Event 관련 변수 정의하는 코드 순서 변경
- 2) “FU\_Duration” 변수가 -1과 1 사이(즉, “A1c\_Create\_TimeStamp” > “Date\_of\_follow\_up” 혹은 “A1c\_Create\_TimeStamp” < “Date\_of\_follow\_up”인데 같은 날이고 시간만 다른 경우)인 관측치 확인 후, 제거

### 〈Way & Result〉

1. “FU\_Duration” 변수 생성 후 “FU\_Duration” 변수가 0인 관측치 삭제 & Replication number 조정

- 1-①. 기존 “data” tbl에서 각 Patient\_ID마다 unique number ID 부여(“ID”가 변수명)  
& Eye 범주와 대응되는 numerical value 가지는 변수 생성 (“Eye\_num”이 변수명)  
-- “data\_ver0” tbl

- 1-②. 기존의 FU\_Duration 변수 생성 규칙 이용해 변수 생성  
-- “data\_FU\_Duration” tbl

- 1-③. “data2” tbl(“data” tbl에서 마지막 반복 측정치의 FU\_Duration 변수 값만 정의된 tbl)와  
“data\_FU\_Duration” tbl을 “ID”, “Replication”, “Eye\_num” 기준으로 joint  
--- “mydata\_ver0” tbl

- 1-④. “mydata\_ver0” tbl에 Patient\_ID, Eye당 반복측정된 총횟수(“N\_count”가 변수명),  
(총 반복측정횟수 - 관측치 반복측정횟수) 뜻하는 변수(“ow\_n”이 변수명) 생성  
--- “new\_add\_N” tbl

- 1-⑤. PROC IML 이용해 Replication number 변경 --- “new” tbl

PROC IML;

```
USE new_add_N var {ID Eye_num Date_of_baseline_exam_2 A1c_Create_TimeStamp_2 Replication Replication2 N_count ow_n FU_Duration};  
READ ALL into result2[colname=varNames];
```

```
DO i=1 TO NROW(result2);
```

```
IF (result2[i,"FU_Duration"] = 0) & (result2[i,"ID"] = result2[(i+1),"ID"]) & (result2[i,"Eye_num"] = result2[(i+1),"Eye_num"]) THEN do;  
result2[(i+1) , "Replication2"] = result2[i, "Replication2"];  
result2[(i+2):(i+1+result2[(i+1),"ow_n"]), "Replication2"] = result2[(i+2):(i+1+result2[(i+1),"ow_n"]), "Replication"] - 1;  
END;
```

```
END;
```

```
create newdata_ver1_2 FROM result2[colname=varNames];
```

```
APPEND FROM result2;
```

```
CLOSE new_add_N;
```

```
QUIT;
```

---

1-⑥. “FU\_Duration” 변수가 -1과 1 사이(즉, “A1c\_Create\_TimeStamp” > “Date\_of\_follow\_up” 혹은 “A1c\_Create\_TimeStamp” < “Date\_of\_follow\_up”인데 같은 날이고 시간만 다른 경우)인 관측치 확인 후, 제거

: 마지막으로, “FU\_Duration” 변수 값을 365.25로 나누어 scale 조정함.

-- “mydata\_ver4” tbl

## 2. 기존에 Event 관련 변수 정리하는 규칙 이용해 변수들 정리

--- “newmydata” tbl (저장함.)

---

## 12월 26일 Version

### 〈What TO DO〉

1) model 적합하기 위해 Data의 time 변수 형태를 “(start, stop)” 형태로 변환하기

2) “PROG PHREG” 적합하기

### 〈Way & Result〉

1) Data 형태 “(start, stop)” 추가 (time covariate를 coxph에 적용하기 위해)

1)-①. “FU\_Duration” 변수 누적합 계산한 새로운 변수 생성

--- “cumulation\_FU\_Duration” 변수명으로 생성

1)-②. “(start, stop)” 생성

: Patient\_ID와 Eye 기준으로 데이터 grouping 한 후, first.Eye일 때는 “start”=0  
나머지 반복 측정치의 경우 이전 반복 측정치의 “cumulation\_FU\_Duration”을 대입  
“stop”은 모든 반복 측정치에 대해 “cumulation\_FU\_Duration”을 대입

1)-③. “cumulation\_FU\_Duration” 변수명 수정

: “cumulation\_FU\_Duration” 변수명을 “stop”으로 변경

-- “PHREG\_data” tbl

## 2) “PHREG” Procedure 적용하기

: 이때, PS model에서 기존 “A1c\_Baseline” 변수를 “A1c\_Result” 변수로 바꾸어야 하는데 Patient\_ID가 “DEBE” 인 관측치에서 “A1c\_Result”가 “see note”인 경우가 있어 Patient\_ID가 “DEBE” 인 관측치 전부 제거.

: “A1c\_Result” 변수 속성을 문자형에서 수치형으로 변환