

健保資料庫分析 – 資料篩選與整理常用的 SAS 語法應用

蔡靜雯 副統計分析師

在執行任何分析前，資料整理是最繁瑣最耗時也是最重要的一個環節，這邊將用範例介紹，提供幾個在健保資料庫分析前，資料整理常用的 SAS 語法。健保資料庫包含許多類型的資料，例如：就醫紀錄檔、承保檔、出生通報檔、死因統計檔、重大傷病檔、癌症相關紀錄檔、交通事故檔、兒童及少年保護通報檔...等。每一種類型的檔案資料量不同，儲存方式也不一樣，資料量比較大的會以一個年度一個檔案或一個月份一個檔案的方式儲存，例如：重大傷病檔 (以年度存檔)，全民健保處方及治療明細檔-門急診 (以月份存檔)；資料量比較小的就會儲存成一個或兩個檔案，例如：癌症登記檔LF。在此將會使用就醫紀錄中的全民健保處方及治療明細門急診檔、全民健保處方及治療明細住院檔、全民健保處方及治療醫令明細門急診檔和全民健保處方及治療醫令明細住院檔，以這四種檔案類型的資料示範，使用 macro 搭配 do...end 語法撈取就醫紀錄和 procsql 語法撈用藥紀錄。接著，再介紹 proc transpose 轉置語法在健保資料整理上的應用和 SAS 日期轉換的指令。

[註] 範例所使用的就醫紀錄檔是百萬歸人檔，仿照健保資料庫的檔名儲存，其檔案內容的欄位名稱和全民健保資料庫不全相同。

在開始範例介紹前，先對這四個就醫紀錄檔案做說明：

檔案	內容	儲存方式	串檔比對欄位
全民健保處方及 治療明細檔-門急 診(H_NHI_OPDTE)	1.一次就醫一筆資料。 2.病人當次就醫之疾病、醫 院及費用等相關資料。	月份	FEE_YM + APPL_TYPE + HOSP_ID + APPL_DATE + CASE_TYPE + SEQ_NO
全民健保處方及	1.一次或一段期間住院一筆	年度	FEE_YM + APPL_TYPE +

治療明細檔-住院 (H_NHI_IPDTE)	資料。 2.病人當次住院之診斷、醫療機構及費用等相關資料		HOSP_ID + APPL_DATE + CASE_TYPE + SEQ_NO
全民健保處方及 治療醫令明細檔- 門急診 (H_NHI_OPDTE)	1.一次門診多筆醫令資料。 2.醫令碼、數量及點數等。 (用藥和治療處置資訊)	月份	無病人ID，以 FEE_YM + APPL_TYPE + HOSP_ID + APPL_DATE + CASE_TYPE + SEQ_NO 與門診明細檔 (H_NHI_OPDTE)對照
全民健保處方及 治療醫令明細檔- 住院 (H_NHI_IPDTE)	1.一次住院多筆醫令資料。 2. 醫令碼、數量及點數 等。(用藥和治療處置資訊)	月份	無病人ID，以 FEE_YM + APPL_TYPE + HOSP_ID + APPL_DATE + CASE_TYPE + SEQ_NO 與住院明細檔 (H_NHI_IPDTE)對照

一、使用 macro 搭配迴圈 do...end 語法撈取就醫紀錄

語法：

```
%macro;
  %do var=/*迴圈起始值*/ %to /*迴圈結束值*/ ;
    } /*重覆動作的指令*/
  %end;
%mend;
```

研究對象以糖尿病為例子，糖尿病的疾病碼(ICD-9)為 250-250.3 250.7。門急診、住院檔的主診斷碼或其他次診斷碼任一碼有糖尿病的疾病碼，即定義為有一次就醫紀錄。

[使用檔案]：全民健保處方及治療明細檔-門急診(H_NHI_OPDTE)

全民健保處方及治療明細檔-住院(H_NHI_IPDTE)

住院就醫紀錄：

由於治療明細住院檔是以年度儲存檔案，若資料區間是 90~100 年，要撈取全部年份的住院檔，就必須重覆讀入篩選 11 次檔案，此時，便可用 macro 搭配迴圈 do ... end 語法，省去重覆 code 書寫的動作。

```
%macro DM_ip;
%do year=90 %to 100;
  data DM_ip&year;
    set h_nhi_ipdte&year(keep=id id_birthday in_date ICD9CM_CODE_0
ICD9CM_CODE_1 ICD9CM_CODE_2 ICD9CM_CODE_3 ICD9CM_CODE_4 id_sex);
    if substr(ICD9CM_CODE_0,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") |
      substr(ICD9CM_CODE_1,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") |
      substr(ICD9CM_CODE_2,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") |
      substr(ICD9CM_CODE_3,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") |
      substr(ICD9CM_CODE_4,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") ;
    run;
%end;
%mend ;
```

其中，substr 為截取字串的函數。

語法： substr(字串, 截取字串的起始位置, 截取幾個字元)

門急診就醫紀錄：

治療明細門急診檔是以月份儲存檔案，若資料區間是 90~100 年，要撈取全部月份的門急診檔，就必須重覆讀入篩選 132 次檔案，此時，同樣可用 macro

搭配迴圈 do ... end 語法，以年份+月份方式，省去重覆 132 次 code 書寫的動作。但這邊 macro 搭配 do ... end 的語法和撈取住院就醫紀錄方式不太一樣，要分成兩個步驟執行：

- 步驟 1. 先寫函數形式的 macro - 指定要撈出糖尿病就診的年度月份。

```
%macro DM_op(year, month);
/*year - 指定撈取資料的年度
/*month - 指定撈取資料的月份*/
data DM_op&year.&month;
set h_nhi_opdte&year.&month(keep=id id_birthday func_date
ACODE_ICD9_1 ACODE_ICD9_2 ACODE_ICD9_3 id_sex);
if substr(ACODE_ICD9_1,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") |
substr(ACODE_ICD9_2,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") |
substr(ACODE_ICD9_3,1,4) in
("2500","2502","2503","2504","2505","2506","2507","2508","2509") ;
run;
%mend ;
```

- 步驟 2. 再將步驟 1 的 macro 函數包在迴圈形式的 macro 內 - 利用迴圈將所有年度月份的糖尿病就診紀錄都撈出來。

```
%macro DM_opall(starty, endy);
/*starty - 撈取資料的起始年度*/
/*endy - 撈取資料的結束年度*/
%do year=&starty %to &endy;
%do month=1 %to 9; %DM_op(&year, 0&month); %end;
%do month=0 %to 2; %DM_op(&year, 1&month); %end;
%end;
%mend ;
%DM_opall(starty=90, endy=100);
```

二、使用 proc sql 撈用藥紀錄

撈取有使用 metformin 用藥的 ID。

[使用檔案]：全民健保處方及治療明細檔-門急診(H_NHI_OPDTE)

全民健保處方及治療明細檔-住院(H_NHI_IPDTE)

全民健保處方及治療醫令明細檔-門急診(H_NHI_OPDIO)

全民健保處方及治療醫令明細檔-住院(H_NHI_IPDIO)

用藥資訊放於全民健保處方及治療醫令明細檔內，但醫令明細檔沒有 ID 欄位，

所以必須用串檔比對欄位 FEE_YM+APPL_TYPE+HOSP_ID+APPL_DATE+

CASE_TYPE+SEQ_NO 串回全民健保處方及治療明細檔。

- 步驟 1. 先從醫令檔中抓取有 metformin 的藥碼，其中 metformin 藥碼的檔

案欄位內容如下：

	ATC_code	drug_id	contain	contain_unit	drug_status
1	A10BA02	A034096100	500 MG		metformin
2	A10BA02	A0340961G0	500 MG		metformin
3	A10BA02	A034549100	500 MG		metformin
4	A10BA02	A035324100	500 MG		metformin
5	A10BA02	A0353241G0	500 MG		metformin
6	A10BA02	A035665100	250 MG		metformin
7	A10BA02	A0356651G0	250 MG		metformin
8	A10BA02	A036201100	500 MG		metformin
9	A10BA02	A0362011G0	500 MG		metformin
10	A10BA02	A036204100	500 MG		metformin
11	A10BA02	A0362041G0	500 MG		metformin
12	A10BA02	A036348100	250 MG		metformin
13	A10BA02	A036349100	500 MG		metformin
14	A10BA02	A0363491G0	500 MG		metformin
15	A10BA02	A036820100	500 MG		metformin
16	A10BA02	A0368201G0	500 MG		metformin

```

/*抓取門急診醫令檔中 metformin 藥碼*/
proc sql;
create table drug_op9001 (keep=hosp_id appl_date fee_ym appl_type
case_type seq_no drug_id ATC_CODE contain drug_status ) as
select a.*, b.*
from drug /*metformin 健保藥碼*/ as a, h_nhi_opdto9001 /*門急診醫令檔*/
as b
where a.drug_id=b.drug_no;
quit;

```

- 步驟 2. 以串檔比對欄位 FEE_YM+APPL_TYPE+HOSP_ID+APPL_DATE+CASE_TYPE+SEQ_NO 將有 metformin 資訊的醫令檔串回明細檔，即可找出有使用 metformin 的 ID。

```

/*將門急診明細檔與門急診醫令檔合併*/
proc sql;
create table temp.drug2_&from.&year.&month(keep=id id_birthday
func_date drug_id ATC_CODE ) as
select *
from H_nhi_opdte9001 /*門診明細檔*/ as a
inner join drug_op9001 /*抓好metformin的醫令檔*/
( rename=(fee_ym=to_fee_ym appl_type=to_appl_type
appl_date=to_appl_date case_type=to_case_type seq_no=to_seq_no
hosp_id=to_hosp_id )) as b
on a.fee_ym=b.to_fee_ym and a.appl_type=b.to_appl_type and
a.appl_date=b.to_appl_date and a.case_type=b.to_case_type and
a.seq_no=b.to_seq_no and a.hosp_id=b.to_hosp_id ;
quit;

```

三、proc transpose 轉置語法的應用

語法：

```

proc transpose data=/*要轉置的檔案*/
               out=/*轉置後輸出的檔案*/;
var /*要轉置的變數*/;
by /*根據哪個變數來依序轉置*/ ;
run;

```

健保資料分析中，常會需要知道同一個 ID 不同治療方式的先後順序，這時候就可以使用 transpose 語法將同一 ID 多筆不同治療資料轉置在同一列，找出治療順序組合，再做後續的比較。

以肺癌病人為例，若已知肺癌病人化學治療(CT)、標靶治療(TKI)、手術治療(surgery)和放射治療(RT)的最早治療日期，檔名分別為 CT_treat、TKI_treat、surgery_treat、RT_treat，檔案內容和執行步驟如下：

先讀入治療檔案

```

/*肺癌病人的最早化學治療日期*/
data CT_treat;
input id $ first_treat_date $ treat $ @@;
cards;
C 20090816 CT D 20090326 CT F 20111001 CT H 20090529 CT
J 20120923 CT K 20100623 CT ;
run;

/*肺癌病人的最早標靶治療日期*/
data TKI_treat;
input id $ first_treat_date $ treat $ @@;
cards;
J 20101228 TKI F 20090816 TKI B 20100106 TKI A 20091221 TKI
G 20081001 TKI H 20110529 TKI I 20081111 TKI C 20091224 TKI
D 20120923 TKI K 20111123 TKI
;
run;

/*肺癌病人的最早手術治療日期*/
data surgery_treat;
input id $ first_treat_date $ treat $ @@;
cards;
B 20121228 surgery F 20100616 surgery E 20091221 surgery
I 20110525 surgery C 20100817 surgery J 20111001 surgery
;
run;

/*肺癌病人的最早放射治療日期*/
data RT_treat;
input id $ first_treat_date $ treat $ @@;
cards;
E 20110512 RT B 20101228 RT G 20101228 RT
I 20110326 RT D 20121111 RT C 20120529 RT
A 20110923 RT
;
run;

```

- 步驟 1. 將放射治療、化學治療、標靶治療和手術治療的檔案合併。

```
data treat_all;
set ct_treat
    rt_treat
    surgery_treat
    TKI_treat;
run;
```

- 步驟 2. 所有治療合併後，將同一 ID 的治療日期由小到大排序。

```
proc sort data=treat_all;
by id first_treat_date;
run;
```

output

	id	first_treat_date	treat
1	A	20091221	TKI
2	A	20110923	RT
3	B	20100106	TKI
4	B	20101228	RT
5	B	20121228	surgery
6	C	20090816	CT
7	C	20091224	TKI
8	C	20100817	surgery
9	C	20120529	RT
10	D	20090326	CT
11	D	20120923	TKI
12	D	20121111	RT

- 步驟 3. 把同一人的所有治療依序轉置(直轉橫)放在同一列。

```
proc transpose data=treat_all
                out=treat_all_t;
var treat;
by id ;
run;
```

output

	id	先前變數的名稱	COL1	COL2	COL3	COL4
1	A	treat	TKI	RT		
2	B	treat	TKI	RT	surgery	
3	C	treat	CT	TKI	surgery	RT
4	D	treat	CT	TKI	RT	
5	E	treat	surgery	RT		
6	F	treat	TKI	surgery	CT	
7	G	treat	TKI	RT		
8	H	treat	CT	TKI		
9	I	treat	TKI	RT	surgery	
10	J	treat	TKI	surgery	CT	
11	K	treat	CT	TKI		

- 步驟 4. 最後以 catx 指令將同一人的所有治療合併在同一欄位(串連字串)

語法： catx(字串間的分隔符號, 字串 1, 字串 2, ..., 字串 n)

```
data treat_all_ok;
set treat_all_t;
sequence=catx(" -> ", col1, col2, col3, col4);
run;
```

output

	id	先前變數	COL1	COL2	COL3	COL4	sequence
1	A	treat	TKI	RT			TKI -> RT
2	B	treat	TKI	RT	surgery		TKI -> RT -> surgery
3	C	treat	CT	TKI	surgery	RT	CT -> TKI -> surgery -> RT
4	D	treat	CT	TKI	RT		CT -> TKI -> RT
5	E	treat	surgery	RT			surgery -> RT
6	F	treat	TKI	surgery	CT		TKI -> surgery -> CT
7	G	treat	TKI	RT			TKI -> RT
8	H	treat	CT	TKI			CT -> TKI
9	I	treat	TKI	RT	surgery		TKI -> RT -> surgery
10	J	treat	TKI	surgery	CT		TKI -> surgery -> CT
11	K	treat	CT	TKI			CT -> TKI

四、其他 SAS 常用指令

- ✓ 日期函數

語法： mdy(month, day, year)

在健保資料中的日期欄位都是以文字儲存，若要計算就醫時年齡，就需要使用日期函數轉換成 SAS 的內部數字。

範例：aa 檔案中的 birthday 和 func_date 分別為生日和就醫日期，且都以

文字格式儲存。

```
data aa;
input id $ birthday $ func_date $;
cards;
A 19511021 20010921
B 19800725 20120212
C 19950519 20091207
D 19600926 20050428
E 20020729 20120831
;
run;
```

- 步驟 1. 將 birthday 和 func_date 用 mdy 日期函數轉換。
- 步驟 2. 日期轉換後的 birthday 和 func_date 相減，再除以 365.25(一年)即可得到就醫時年齡。

```
data aal;
set aa;
/*步驟 1*/
birthday_ok=mdy(substr(birthday,5,2), substr(func_date,7,2),
                 substr(birthday,1,4));
func_date_ok=mdy(substr(func_date,5,2),
                 substr(func_date,7,2), substr(func_date,1,4));
/*步驟 2*/
age=(func_date_ok-birthday_ok)/365.25;
run;
```

其中，substr(birthday,5,2) 為截取 birthday 的月份，

substr(birthday,7,2) 為截取 birthday 的日，

substr(birthday,1,4) 為截取 birthday 的年度。

output

	id	birthday	func_date	birthday_ok	func_date_ok	age
1	A	19511021	20010921	-2994	15239	49.919233402
2	B	19800725	20120212	7498	19035	31.586584531
3	C	19950519	20091207	12910	18238	14.587268994
4	D	19600926	20050428	271	16554	44.580424367
5	E	20020729	20120831	15552	19236	10.0862423

✓ 日期格式

語法： `format` 要轉換的變數 日期格式

日期字串經由 `mdy(month, day, year)` 轉換為 SAS 的內部數字，在呈現上不易觀看，可用 `format` 轉換成日期格式較易觀察，但實際上還是 SAS 的內部數字。

接續上一個例子，只要再加上 `format` 指令，並指定要轉換的日期格式即可。

```
data aa2(drop=func_date func_date_ok age);
set aa1;
birthday1=birthday_ok;
birthday2=birthday_ok;
birthday3=birthday_ok;
birthday4=birthday_ok;
birthday5=birthday_ok;
birthday6=birthday_ok;
format birthday1 yymmdd10.
      birthday2 mmddyy10.
      birthday3 mmddyy8.
      birthday4 date9.
      birthday5 worddate.
      birthday6 weekdate.;
run;
```

output

	id	birthday	birthday_ok	birthday1	birthday2	birthday3	birthday4	birthday5	birthday6
1	A	19511021	-2994	1951-10-21	10/21/1951	10/21/51	21OCT1951	October 21, 1951	Sunday, October 21, 1951
2	B	19800725	7498	1980-07-12	07/12/1980	07/12/80	12JUL1980	July 12, 1980	Saturday, July 12, 1980
3	C	19950519	12910	1995-05-07	05/07/1995	05/07/95	07MAY1995	May 7, 1995	Sunday, May 7, 1995
4	D	19600926	271	1960-09-28	09/28/1960	09/28/60	28SEP1960	September 28, 1960	Wednesday, September 28, 1960
5	E	20020729	15552	2002-07-31	07/31/2002	07/31/02	31JUL2002	July 31, 2002	Wednesday, July 31, 2002

參考資料

1. SAS Programming1: Essentials.
2. SAS Programming2: Data Manipulation Techniques.