利用 PROC FROMAT 擷取資料 – SAS 程式

嚴友君

臺北醫學大學 生物統計研究中心 助理研究員

在資料處理與分析過程，我們常常會需要從龐大的原始資料中，擷取一部分符合分析條件的資料後，再做後續的分析或統計計算。如果擷取條件簡單，例如只留下男性，或是例如只留下接受特定科別治療的病人，簡單直觀的方法可以使用 IF 條件式或 WHERE 條件式來達成。但是如果擷取條件比較複雜，例如你手上有一個上千人的 ID 列表，要擷取出這個 ID 列表中的人的資料用以後續分析；或是例如要擷取有特定藥物治療的病人資料，而符合分析的藥品代碼有上百個。這個時候使用 IF 或 WHERE 條件式就不是一個好的選擇，一方面程式會是一長串 IF 或 WHERE 條件式而非常冗長，冗長的程式就很可能有輸入/打字錯誤；另一方面一但擷取的條件有些微的修改，就必需從冗長的程式中找到要修改的地方，不利於程式的管理。所以在這個情況下，比較好的策略是將符合條件的 ID 例表或藥品代碼列表儲存成一個資料檔，將原始資料檔與條件列表資料利用“SORTED BY MERGE”的方式，從原始資料內篩選出與條件列表資料一致的部分，即可達成擷取資料的目的。除了上述使用 IF 或 WHERE 條件式或“SORTED BY MERGE”的方法，本篇將介紹另一種擷取資料的方法。這個方法是使用 PROC FORMAT 來達成，尤於這個方法不需要 SORT 原始資料，預期會較使用“SORTED BY MERGE”節省時間。

FORMAT 自訂格式

以下利用附錄程式 A(圖 15)模擬之資料為例子說明。模擬資料有 6 個變數，SAS 資料名稱為 source，以下範例只會用到 ID 及 GENDER 兩個變數。其中 ID 為唯一識別碼，資料類型為字元；GENDER 為性別，資料類型為字元，其中 0 為女性、1 為男性，空白為缺失值。

我們先利用 GENDER 這個最簡單的變數為例，瞭解 SAS 中 FORMAT 的使用與 FORMAT 形成後 SAS 自動建立關聯這個特定 FORMAT 的資料格式與內容。

我們可以用以下程式自定義一個 SEX 文字格式，指定 0 為 Female，1 為 Male。列印前 10 筆資料並將 GENDER 這個變數套用 SEX 格式列印，在使用文字格式的

format 時，要在前面加入“$”符號。

```
PROC FORMAT;
  VALUE $ sex
    "0" = "Female"
    "1" = "Male"
    ;
    RUN;
PROC PRINT DATA=source (OBS=10);
    RUN;
PROC PRINT DATA=source (OBS=10);
    FORMAT GENDER $sex.;
    RUN;
```

圖 1

沒有套用 SEX 格式與有套用 SEX 格式之列印結果如下：

觀測值	ID	X1	X2	X3	X4	GENDER
1	ID000000001	393	-0.2722	28	6	0
2	ID000000002	216	-50.6571	-2	6	1
3	ID000000003	436	22.7932	24	9	1
4	ID000000004	363	-17.2978	-24	4	
5	ID000000005	170	43.1708	-159	4	0
6	ID000000006	315	-19.6862	18	8	
7	ID000000007	420	-31.5780	-64	3	1
8	ID000000008	296	27.0547	19	9	0
9	ID000000009	502	0.1234	-44	5	1
10	ID000000010	553	-9.5580	-55	8	

觀測值	ID	X1	X2	X3	X4	GENDER
1	ID000000001	393	-0.2722	28	6	Female
2	ID000000002	216	-50.6571	-2	6	Male
3	ID000000003	436	22.7932	24	9	Male
4	ID000000004	363	-17.2978	-24	4	
5	ID000000005	170	43.1708	-159	4	Female
6	ID000000006	315	-19.6862	18	8	
7	ID000000007	420	-31.5780	-64	3	Male
8	ID000000008	296	27.0547	19	9	Female
9	ID000000009	502	0.1234	-44	5	Male
10	ID000000010	553	-9.5580	-55	8	

圖 2

如果我們要利用 PROC FORMAT 來擷取 source 資料集內的男性 (雖然在這個例子以 IF 或 WHERE 條件式即可)，可以用下面這個方法。1. 建立一個自訂義文字格式 include，1 為男性定義為“YES” (要納入之意)，其他定義為“NO” (不要納入之意)。利用一個 IF 條件式結合 PUT，將“YES”留下，儲存於新資料集 include 中。最後可以利用 PROC FREQ 檢視一下男性人數與 include 資料集中的人數是否一致。

範例程式如下：

```
PROC FORMAT;
  VALUE $ include
    "1" = "YES"
    other = "NO "
    ;
    RUN;
DATA include;
    SET source;
    IF PUT(GENDER, $include.)="YES";
    RUN;

PROC FREQ DATA=source;
    TABLES GENDER / LIST MISSING;
    FORMAT GENDER $sex. ; run;
```

圖 3

GENDER	次數	百分比	累積次數	累積百分比
	2000099	20.00	2000099	20.00
Female	4002054	40.02	6002153	60.02
Male	3997847	39.98	10000000	100.00

圖 4

```
42   DATA include;
43       SET source;
44       IF PUT(GENDER, $include.)="YES";
45       RUN;

NOTE: There were 10000000 observations read from the data set WORK.SOURCE.
NOTE: The data set WORK.INCLUDE has 3997847 observations and 6 variables.
```

圖 5

PROC FORMAT 的資料格式與內容

假設我們有一個 ID 列表 [利用附錄程式 B(圖 16)模擬之資料]，想仿照上面擷取男性資料的方法使用 PROC FORMAT。首先需要將這個 ID 列表建立一個自定義 format，在這個 ID 列表內的 format 定義為 “YES”，其他為 “NO”，後續就可以利用 IF 條件式與 PUT 來篩選出資料。但是在將 ID 列表自定義 format 時，我們可以利用 PROC FORMAT 中 CNTLIN 這個語句來建立，而不需要將所有 ID 以 VALUE 的方法一個一個謄打在 SAS 程式中。首先我們可以利用 PROC FORMAT 中

的 CNTLOUT 語句輸出 format 的資料格式與內容，瞭解一下 SAS 中 format 資料的固定所需變數與格式。

以下程式將輸出前面例建立的 include 格式於 include_fmt 資料集，之後列印出這個 include_fmt 資料集。

```
PROC FORMAT cntlout=include_fmt;
  SELECT $include;
PROC PRINT DATA=include_fmt;
  RUN;
```

圖 6

觀測值	FMTNAME	START	END	LABEL	MIN	MAX	DEFAULT	LENGTH	FUZZ	PREFIX	MULT	FILL	NOEDIT	TYPE	SEXCL	EEXCL	HLO
1	INCLUDE	1	1	YES	1	40	3	3	0		0		0	C	N	N	
2	INCLUDE	**OTHER**	**OTHER**	NO	1	40	3	3	0		0		0	C	N	N	O

圖 7

圖 7 中截取 include_fmt 資料集的內容，其中以下幾個變數為利用 CNTLIN 自定義 FORMAT 的重要變數：

FMTNAME – 在使用這個 format 會引用的名字

START – 資料內容的起始值。在文字格式，與 END 會相同。

END – 資料內容的結束值。在文字格式，與 START 會相同。

LABEL – 當套用格式後，START 至 END 的值會以 LABLE 內型式呈現。

TYPE – “C” 代表為文字格式。

HLO – “O” 代表其他沒有包含在所有 START / END 內容內的值，若 HLO= “(字母)O”，SAS 會自動將 START 變項及 END 變項以 other 字串註記。

利用 SAS 資料集及 PROC FORMAT 的 CTLIN 語句自定義 format 格式

以下例子為，我們有一個 ID 列表 [利用附錄程式 B(圖 16)模擬之資料 id_list]，希望將 source 資料集中符合這個 ID 列表的資料擷取出來。第一步是製作一個 format 制式格式的資料，如前一節的說明，這個資料必需包含以下變數：FMTNAME、START、END、LABEL、TYPE、HLO。其中 START 及 END 即為 ID 列表中的 ID。參考程式如下：

```
DATA id_fmt1 (keep=fmtname type start end label);
    set id_list;
    fmtname="include_id";
    type="C";
    start=ID;
    end=ID;
    label="YES";
    RUN;
DATA other;
    fmtname="include_id";
    type="C";
    hlo="O";
    label="NO ";
    RUN;
data id_fmt;
    set id_fmt1 other;
    RUN;
```

圖 8

製作好的 format 格式資料為 id_fmt，資料筆數應較 id_list 資料多 1 筆，這多出來的 1 筆為註記 others 的 label。接下來很重要的一個步驟為確保 format 格式資料沒有重覆，可利用 PROC SORT NODUPKEY 去除重覆。確保沒有重覆後，再利用 PROC FORMAT CNTLIN 讀入 id_fmt 資料，即可成功自訂義 include_id 這個格式。

```
PROC SORT DATA=id_fmt NODUPKEY;
    BY start; RUN;
PROC FORMAT CNTLIN=id_fmt; run;
```

圖 9

```
89   PROC FORMAT CNTLIN=id_fmt;
NOTE: 格式 $INCLUDE_ID 已經輸出。
89 !                           run;

NOTE: PROCEDURE FORMAT used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

NOTE: There were 1001 observations read from the data set WORK.ID_FMT.
```

圖 10

可以利用 PROC FORMAT 的 SELECT 語句確認自訂義文字格式 include_id，將詳細

格式內容列印在輸出中。

```
PROC FORMAT;
    SELECT $include_id; RUN;
```

圖 11

```
----------------------------------------------------------------------------
|                    FORMAT NAME: $INCLUDE_ID LENGTH: 3                    |
|   MIN LENGTH:   1  MAX LENGTH:  40  DEFAULT LENGTH:   3  FUZZ:        0  |
|--------------------------------------------------------------------------|
|START           |END             |LABEL  (VER. 9.4     16JUN2019:21:05:30)|
|----------------+----------------+-----------------------------------------|
|ID000009397     |ID000009397     |YES                                      |
|ID000010174     |ID000010174     |YES                                      |
|ID000016876     |ID000016876     |YES                                      |
|ID000025117     |ID000025117     |YES                                      |
|ID000026933     |ID000026933     |YES                                      |
|ID000029065     |ID000029065     |YES                                      |
|ID000037563     |ID000037563     |YES                                      |
|ID000046066     |ID000046066     |YES                                      |
|ID000047440     |ID000047440     |YES                                      |
|ID000050773     |ID000050773     |YES                                      |
|                                                                          |
|ID009916212     |ID009916212     |YES                                      |
|ID009923758     |ID009923758     |YES                                      |
|ID009952859     |ID009952859     |YES                                      |
|ID009960338     |ID009960338     |YES                                      |
|ID009969550     |ID009969550     |YES                                      |
|ID009975193     |ID009975193     |YES                                      |
|ID009984606     |ID009984606     |YES                                      |
|**OTHER**       |**OTHER**       |NO                                       |
----------------------------------------------------------------------------
```

圖 12

利用自訂義 format 格式及 IF 條件式及 PUT 篩選資料

上一節自訂義文字格式 include_id，定義將符合條件的 ID 標示為 “YES”，其他為 “NO”。利用 PUT 及這個格式，將 “YES” 的人篩選出來，就可以得到原始 source 中，符合 id_list 中 ID 的資料。範例程式如下：

```
DATA source_include;
    SET source;
    IF PUT(ID, $include_id.)="YES";
    RUN;
```

圖 13

```
NOTE: There were 10000000 observations read from the data set WORK.SOURCE.
NOTE: The data set WORK.SOURCE_INCLUDE has 1000 observations and 6 variables.
```

圖 14

以上介紹的方法，只需要排序 id_list 資料，而不需要排;序 source 資料。如果使用“SORTED BY MERGE”方法，則必需排需 source 與 id_list 資料。若 source 資料龐大，將會耗費較多時間，但是利用 format 則會受限於記憶體的大小。由於 format 格式設定後，會儲存於記憶體內，format 格式內的最大可儲存筆數將受到記憶體大小的限制。本篇文章簡單介紹基礎利用 format 格式擷取資料的方法，提供 SAS 分析資料使用者靈活運用 SAS 程式的一些技巧與想法。

參考閱讀資料

SAS Institute, Inc. (2017), Base SAS 9.4 **Procedures Guide**, Seventh Edition. Cary, NC: SAS Institute, Inc.
http://documentation.sas.com/?docsetId=proc&docsetTarget=p1xidhqypi0fnwn1if8opjpqpbmn.htm&docsetVersion=9.4&locale=en

Jenine Milum, Wells Fargo, Charlotte, NC. “**Proc Format, a Speedy Alternative to Sort / Sort / Merge**.”
http://support.sas.com/resources/papers/proceedings12/428-2012.pdf

John Cohen, AstraZeneca LP, Wilmington, DE. "**Table Lookups: Getting Started With Proc Format**."
https://www.lexjansen.com/nesug/nesug08/cc/cc19.pdf

```
/* 附錄程式A */
DATA source;
   DO i=1 TO 1E7;
   LENGTH ID $12;
   ID=cat("ID", PUT(i, z9.));
   X1=FLOOR(1000*RANUNI(0));
   X2=25*RANNOR(0);
   X3=FLOOR(50*RANNOR(0));
   X4=RANBIN(0, 30, 0.2);
   LENGTH GENDER $1;
   X5=RANUNI(0);
   IF X5 < 0.4 THEN GENDER="0";
      ELSE IF X5 < 0.8 THEN GENDER="1";
      ELSE GENDER=" ";
   OUTPUT;
   END;
   drop i X5;
   RUN;

proc contents data=source varnum; run;
```

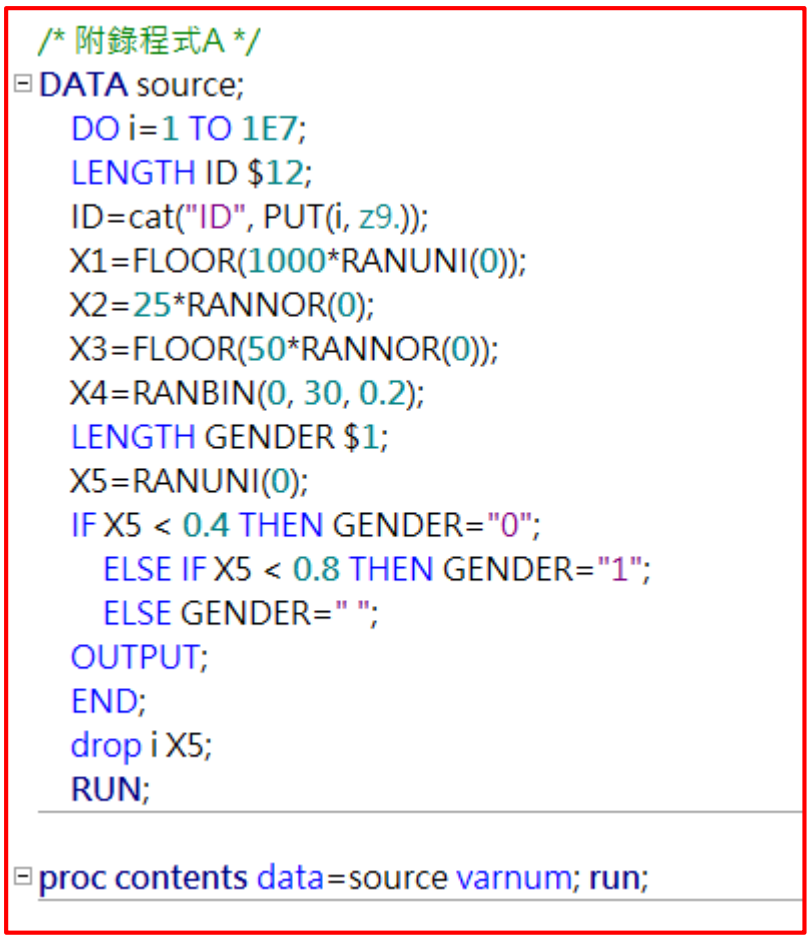

圖 15

```
/* 附錄程式B */
DATA _example;
      DO i=1 TO 1000;
      X1=FLOOR(1E7*RANUNI(0));
      LENGTH ID $12;
      ID=cat("ID", PUT(X1, z9.));
   OUTPUT;
   END;
   IF 0 < X1 < 100;
   drop i X1;
   RUN;
PROC SORT DATA=_example NODUPKEY OUT=id_list;
   BY ID;
   RUN;
data subset.id_list;
   set id_list;
   run;
```

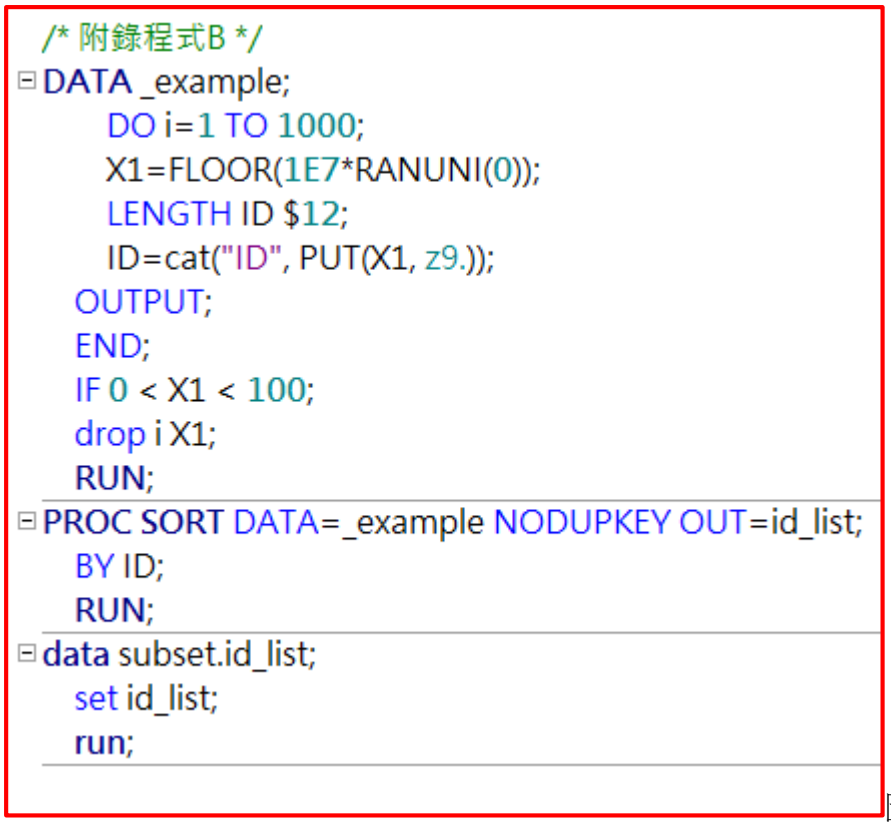

圖 16