실험계획과 분석

심송용(한림대학교 데이터과학스쿨)

http://jupiter.hallym.ac.kr

SAS 기초

위의 SAS 프로그램에서 데이터 세트 B의 변수 이름을 둘 다 X와 Y라고 하고(이 경우 두 데이터 세트의 변수의 이름이 모두 같아진다.) 이들 변수 값을 출력하여 보아라.

MERGE 명령

SET 명령이 둘 이상의 자료를 합하여 하나로 만들 때 자료의 숫자가 늘어나는 방향으로 자료가 합해지는데 MERGE 명령은 둘 이상을 합하여 변수의 개수가 늘어 나는 경우에 사용한다. 다음 예를 보자.

```
/* mergetest1.sas */
DATA SCORE;
 INPUT ID GRADE$ @@;
 CARDS;
 1 A 2 A 3 B 4 C 5 D 6 F
;
DATA GENDER;
```

SAS 기초

```
 INPUT ID SEX$ @@;
 CARDS;
 1 M 2 M 3 M 4 F 5 F 6 F
;
DATA ALL;
 MERGE SCORE GENDER;
;
PROC PRINT DATA=ALL;
  VAR ID SEX GRADE;
RUN;
```

OBS	ID	SEX	GRADE
1	1	M	A
2	2	M	A
3	3	M	B
4	4	F	C

5	5	F	D
6	6	F	F

위의 경우 ID의 변숫값이 두 데이터 세트 SCORE와 GRADE에서 같은 순서로 같은 값이 있으므로 MERGE하는데 문제가 없었다. 하지만 만일 자료가 ID에 따라서 정렬이 되어 있지 않거나 일부 ID는 한 데이터 세트에만 있는 등의 경우 위와 같이 MERGE 할 경우에는 문제가 발생한다. 예를 들어

```
/* mergetest2.sas */
DATA SCORE;
 INPUT ID GRADE$ @@;
 CARDS;
 3 B 4 C 5 D 6 F 1 A 2 A
;
DATA GENDER;
 INPUT ID SEX$ @@;
```

SAS 기초

```
 CARDS;
 1 M 2 M 3 M 4 F 5 F 7 F
;
DATA ALL2;
 MERGE SCORE GENDER;
RUN;
PROC PRINT DATA=ALL2;
  VAR ID SEX GRADE;
RUN;
```

인 경우 MERGE의 결과는 앞의 경우와 달라지며, 같은 ID 값에 의한 매치(match)도 생기지 않는다. 이 경우 BY를 사용하여 다음과 같이

```
PROC PRINT DATA=ALL;
  VAR ID SEX GRADE;
  BY ID;
```

RUN;

를 실행한다. 이 때 BY를 사용하기 위해서는 사전에 두 데이터 세트를 sort 하여야 하며 이를 위해서 SORT 프로시저를 추가하여야 한다. 전체 명령은 다음과 같다.

```
/* mergetest3.sas */
DATA SCORE;
 INPUT ID GRADE$ @@;
 CARDS;
 3 B 4 C 5 D 6 F 1 A  2 A
;
DATA GENDER;
 INPUT ID SEX$ @@;
 CARDS;
 1 M 2 M 3 M 4 F 5 F 7 F
;
```

SAS 기초

```
PROC SORT DATA=SCORE; BY ID;
RUN;
PROC SORT DATA=GENDER; BY ID;
RUN;

DATA ALL2;
 MERGE SCORE GENDER;
 BY ID;
;
PROC PRINT DATA=ALL2 ;
  VAR ID  SEX GRADE;
RUN;
```

를 사용하면 다음과 같이 원하는 출력이 얻어진다.

SAS 기초

OBS	ID	SEX	GRADE
1	1	M	A
2	2	M	A
3	3	M	B
4	4	F	C
5	5	F	D
6	6		F
7	7	F	

SAS의 데이터 파일 출력

두 개 또는 그 이상의 데이터 세트를 합한 경우나 새 변수를 계산한 경우 등에서는 새 자료를 별도의 데이터 파일로 저장할 필요가 있는 경우가 있다. 데이터 세트의 자료를 별도의 텍스트 파일로 저장하고자 할 때는 FILE 명령과 PUT 명령의 조합으로 사용하며 이들은

```
FILE 'path\filename' DLM='delimiter' ENCODING='encoding';
PUT var1 var2 ...;
```

SAS 기초

로 사용하며

- path\filename에는 저장할 파일의 경로와 이름을 설정하고
- delimiter는 각 열을 구분할 문자(구분자)를 설정한다. DLM이 생략되면 구분자는 빈 칸이 사용된다.
- encoding에는 인코딩에 사용할 규격을 설정한다. 한국어인 경우 예전엔 'euc-kr', 최근엔 'utf-8'이 사용되는 것이 바람직하다.
- var1 등에는 이 파일에 저장할 변수를 설정한다.

앞에서 만든 데이터 세트 ALL2를 파일로 인쇄해보자.

```
/* fileouttest1.sas  앞부분은 mergetest3.sas와 같음 */
DATA _NULL_;
 SET ALL2;
 FILE '/folders/myfolders/mydata/textouttest1.txt' DLM=',';
 PUT ID GRADE SEX;
```

SAS 기초

RUN;

이 명령의 결과는 새 파일 textout.txt 파일이 설정된 경로에 만들어지며 이 파일의 내용은

1,A,M

2,A,M

3,B,M

4,C,F

5,D,F

6,F,

7, ,F

이다. 이와 같이 콤마(,)로 값이 구분된 자료의 형식을 CSV(Comma Separated Values)라고 하며 대개 확장자로 .csv를 사용한다.

참고

SAS UniversityEdition의 경우 위 파일의 경로는

SASUniversityEdition/myfolders/mydata/textouttest1.txt

임을 기억하자.

SAS 기초

레이블링(labeling)

변수의 이름과 값에 따로 변수의 설명 및 각각의 값에 대한 설명을 추가할 수 있다.

변수 레이블링

변수 레이블링(labeling)이란 변수이름에 변수의 설명을 추가해주는 기능이다. 이 기능을 사용하여 변수이름에 설명을 설정할 때는 DATA 스텝에서 LABEL 명령으로 변수명 = 변수 설명을 아래의 예와 같이 나열하여 실행한다.

```
DATA  auto2;
  LABEL ID  = "Student ID Number"
        SEX = "Gender Reported"
        GRADE = "Grade s/he got";
RUN;
```

SAS 기초

변숫값 레이블링

변숫값 레이블링은 주로 범주형 자료에서 자료의 각 값에 대한 설명을 추가하는 것을 말한다. 변숫값 레이블링은 PROC FORMAT에서 VALUE 명령으로 아래의 예와 같이 설정한다.

```
PROC FORMAT;
   VALUE   sex 0 ="Male"
               1 = "Female";
   VALUE   $model "Cad."    ="Cadillac (GM)"
                  "Chev."   ="Chevrolet (GM)"
                  "Datsun"  ="Datsun (Nissan)";
RUN;
```

위의 설정에서 보는 것과 같이 수치형 자료는 포맷 형식에 $가 없지만 문자형 변수는 $로 시작한다. 이 FORMAT 프로시저만으로는 출력에 변숫값 레이블링이 반영되지 않으며 변숫값 레이블링을 반영하려면 각 프로시저에서 FORMAT 명령을 주어서 어떤 변수에 어떤 포맷을 사용할지 설정하여야 한다.

SAS 기초

예를 들어 gender 변수엔 FORMAT 프로시저에서 정의한 sex를 사용한다면(마지막의 마침표가 필요함에 주의)

FORMAT gender sex.

로 FORMAT 문을 사용하여야 한다.

앞의 데이터 세트 ALL2를 사용하여 변수명 및 변숫값 레이블링을 다음과 같이 정의하고 마지막에 교차표를 얻어보면

```
/* labeltest1.sas  앞부분은 앞에서 만든 ALL2가 필요*/
DATA  ALL3;
 SET ALL2;
  LABEL  ID   = "Student ID Number"
         SEX  = "Gender Reported"
         GRADE = "Grade s/he got";
RUN;
```

SAS 기초

```
PROC FORMAT;
   VALUE  IDlbl 7 = "Graduated";
   VALUE  $sexlbl "M" ="Male"
                  "F" = "Female";
   VALUE  $gradelbl "F"    ="Failed";
RUN;
PROC FREQ DATA=ALL3;
 TABLE sex * grade;
 FORMAT sex   $sexlbl.
        grade  $gradelbl.;
RUN;
```

결과는 다음과 같다.

SAS 기초

FREQ 프로시저

빈도
백분율
행 백분율
칼럼 백분율

테이블 SEX * GRADE

SEX(Gender Reported)	GRADE(Grade s/he got) A	B	C	D	Failed	합계
Female	0 0.00 0.00 0.00	0 0.00 0.00 0.00	1 20.00 50.00 100.00	1 20.00 50.00 100.00	0 0.00 0.00 .	2 40.00
Male	2 40.00 66.67 100.00	1 20.00 33.33 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 .	3 60.00
합계	2 40.00	1 20.00	1 20.00	1 20.00	0 0.00	5 100.00

결측값 빈도 = 2

SAS 기초

요약하면

1. 변수명 레이블링는 DATA 스텝에서
2. FORMAT 프로시저에서 변숫값 레이블링을 정의하고
3. 실제 분석 프로시저(이 경우 FREQ)에서 FORMAT 문을 사용하여
4. 출력에서 변수명 및 변숫값 레이블링 얻어짐을 알 수 있다.