



실험계획과 분석

심송용(한림대학교 데이터과학스쿨)

<http://jupiter.hallym.ac.kr>

SAS 기초

```
DO X = 1, 5, 8, 11, 57;
```

이라고 하면 X의 값을 1, 5, 8, 11, 57로 바꾸면서 END; 문이 나올 때까지 반복을 하게 된다.

OUTPUT 명령

OUTPUT 명령은 SAS 자료에 현재의 값을 변수값으로 갖는 변수를 만드는 명령이다. 예를 들면

```
/* outputtest1.sas */  
DATA A;  
  DO X = 1 TO 10;  
    IF X ^> 5 THEN Y = 0;  
    ELSE Y = 1;  
    OUTPUT;  
  END;
```

SAS 기초

```
PROC FREQ DATA =A;  
  TABLES X Y;  
RUN;
```

의 결과로 변수 X와 Y가 만들어지며 그 결과는 다음과 같다.

X	Cumulative		Cumulative	
	Frequency	Percent	Frequency	Percent
1	1	10.0	1	10.0
2	1	10.0	2	20.0
3	1	10.0	3	30.0
4	1	10.0	4	40.0
5	1	10.0	5	50.0
6	1	10.0	6	60.0

SAS 기초

7	1	10.0	7	70.0
8	1	10.0	8	80.0
9	1	10.0	9	90.0
10	1	10.0	10	100.0

Y	Cumulative		Cumulative	
	Frequency	Percent	Frequency	Percent
0	5	50.0	5	50.0
1	5	50.0	10	100.0

이는 다음과 같이 (특히 실험계획법 자료에서) 하나의 수준에서 여러 개의 관측치를 갖는 경우에 유용하게 사용할 수 있다.

```
/* outputtest2.sas */
DATA CRD;
```

SAS 기초

```
INPUT FACTOR YIELD1-YIELD4;  
YIELD = YIELD1; OUTPUT;  
YIELD = YIELD2; OUTPUT;  
YIELD = YIELD3; OUTPUT;  
YIELD = YIELD4; OUTPUT;  
DROP YIELD1-YIELD4;  
CARDS;  
1  2 3 4 6  
2  3 5 6 7  
3  3 4 5 6  
;  
PROC PRINT DATA=CRD;  
  VAR FACTOR YIELD;  
RUN;
```

의 결과는

SAS 기초

OBS	FACTOR	YIELD
1	1	2
2	1	3
3	1	4
4	1	6
5	2	3
6	2	5
7	2	6
8	2	7
9	3	3
10	3	4
11	3	5
12	3	6

이다. 위의 예는 세 개의 수준에서 반복수가 4인 실험자료를 읽어서 하나의 변수로 만든 것

SAS 기초

이다. 여기서 YIELD를 만들고 난 후에는 필요없는 변수인 YIELD1부터 YIELD4는 SAS 데이터 세트에서 없애도 되므로 DROP 명령을 사용하였다. 이와 유사하게 다음과 같이 만들어 볼 수도 있다.

```
/* outputtest3.sas */  
DATA A;  
  DO X = 1 To 3;  
    INPUT Y@@;  
    OUTPUT;  
  END;  
CARDS;  
2 3 3      3 5 4  
4 6 5      6 7 6  
;  
PROC PRINT; VAR X Y; RUN;
```

OBS	X	Y
1	1	2
2	2	3
3	3	3
4	1	3
5	2	5
6	3	4
7	1	4
8	2	6
9	3	5
10	1	6
11	2	7
12	3	6

SAS 기초

두 개의 SAS 자료로 따로 출력을 만들 수도 있다.

```
/* outputtest4.sas */
```

```
DATA A B;
```

```
DO X = 1 TO 10;
```

```
IF X ^> 5 THEN OUTPUT A;
```

```
ELSE OUTPUT B;
```

```
END;
```

```
PROC PRINT DATA=A;
```

```
RUN;
```

```
PROC PRINT DATA=WORK.B;
```

```
RUN;
```

OBS	X
1	1
2	2
3	3
4	4
5	5

OBS	X
1	6
2	7
3	8
4	9
5	10

SAS 기초

라고 하면 SAS 자료 A는 1부터 5까지, B는 6부터 10까지의 값을 갖는 변수 X를 포함하게 된다.

DELETE와 DROP 명령

DELETE와 DROP은 각각 필요없는 관측치와 변수를 삭제하는 명령이다. 예를 들어

```
IF X > Y THEN DELETE;
```

라고 하면 X의 값이 Y의 값보다 큰 경우는 모두 삭제한다.

DROP 문은 특정한 변수를 삭제하는 명령문이다. 예를 들어

```
DROP X1-X10;
```

의 결과는 SAS 자료에서 열개의 변수 X1에서 X10까지를 삭제하는 것이다.

SAS 기초

KEEP 명령

KEEP 명령은 DROP 명령과 반대로 작용하는 것으로 KEEP 명령에서 준 변수를 제외한 나머지 변수는 모두 DROP하게 하는 명령이다. 예를 들어

```
DATA C;  
SET A;  
KEEP X1-X5
```

라고 하면 만들어지는 SAS 데이터 세트 C는 데이터 세트 A에서 X1부터 X5 이외의 변수는 모두 삭제한 자료가 된다.

SET 명령

SET 명령은 이미 만들어져 있는 다른 SAS 데이터 세트를 불러오는 명령이다.
사용법은

SAS 기초

```
SET sasdata;
```

인데 *sasdata*에는 불러올 SAS 데이터 세트의 이름을 준다. 만일 이름을 주지 않으면 가장 최근에 만든 SAS 데이터 세트를 불러온다.

둘 이상의 SAS 자료를 불러올 수도 있는데 그런 경우에는 자료를 이름이 나오는 순서대로 위에서 아래로 붙여서 새 자료를 만든다. 따라서 두 개의 자료가 n_1 과 n_2 개의 관측치로 되어 있다면 이 둘을 SET 명령으로 불러 온 결과로 만들어진 자료는 원칙적으로 $n_1 + n_2$ 개의 관측치를 갖는다.

다만 변수 이름이 같은 것이 있을 경우 같은 변수는 하나의 변수 이름으로 저장하고 서로 다른 변수 이름이 있는 경우에는 각각의 변수를 만든다. 이때 변수가 없는 쪽에는 결측치로 취급한다. 예를 들어

SAS 기초

```
/* settest1.sas */  
DATA A;  
  INPUT X Y @@;  
  CARDS;  
  1 2 1 3 1 4 1 5 1 6  
;  
DATA B;  
  INPUT X Z @@;  
  CARDS;  
  2 6 2 7 2 9 2 10  
;  
DATA C;  
  SET A B;  
  
PROC PRINT DATA =C;  
RUN;
```

SAS 기초

의 결과로 만들어진 SAS 출력의 결과는 다음과 같다.

데이터 세트 A	데이터 세트 B	데이터 세트 C																																																																						
<table><tr><td>1</td><td>1</td><td>2</td></tr><tr><td>2</td><td>1</td><td>3</td></tr><tr><td>3</td><td>1</td><td>4</td></tr><tr><td>4</td><td>1</td><td>5</td></tr><tr><td>5</td><td>1</td><td>6</td></tr></table>	1	1	2	2	1	3	3	1	4	4	1	5	5	1	6	<table><tr><th>OBS</th><th>X</th><th>Z</th></tr><tr><td>1</td><td>2</td><td>6</td></tr><tr><td>2</td><td>2</td><td>7</td></tr><tr><td>3</td><td>2</td><td>9</td></tr><tr><td>4</td><td>2</td><td>10</td></tr></table>	OBS	X	Z	1	2	6	2	2	7	3	2	9	4	2	10	<table><tr><th>OBS</th><th>X</th><th>Y</th><th>Z</th></tr><tr><td>1</td><td>1</td><td>2</td><td>.</td></tr><tr><td>2</td><td>1</td><td>3</td><td>.</td></tr><tr><td>3</td><td>1</td><td>4</td><td>.</td></tr><tr><td>4</td><td>1</td><td>5</td><td>.</td></tr><tr><td>5</td><td>1</td><td>6</td><td>.</td></tr><tr><td>6</td><td>2</td><td>.</td><td>6</td></tr><tr><td>7</td><td>2</td><td>.</td><td>7</td></tr><tr><td>8</td><td>2</td><td>.</td><td>9</td></tr><tr><td>9</td><td>2</td><td>.</td><td>10</td></tr></table>	OBS	X	Y	Z	1	1	2	.	2	1	3	.	3	1	4	.	4	1	5	.	5	1	6	.	6	2	.	6	7	2	.	7	8	2	.	9	9	2	.	10
1	1	2																																																																						
2	1	3																																																																						
3	1	4																																																																						
4	1	5																																																																						
5	1	6																																																																						
OBS	X	Z																																																																						
1	2	6																																																																						
2	2	7																																																																						
3	2	9																																																																						
4	2	10																																																																						
OBS	X	Y	Z																																																																					
1	1	2	.																																																																					
2	1	3	.																																																																					
3	1	4	.																																																																					
4	1	5	.																																																																					
5	1	6	.																																																																					
6	2	.	6																																																																					
7	2	.	7																																																																					
8	2	.	9																																																																					
9	2	.	10																																																																					

나머지 변수는 해당되는 곳에만 만든다.

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;
```