3.2.2 IF 문(2) IF (조건) THEN DO; 문장(들); END;

()의 조건이 성립하면 문장들이 실행된다.

학생들의 키(height)가 60인치 보다 큰 관측치(학생)의 경우 몸무게(파운드, weight)는 0.9배, 키는 1.1배 하시오. 변수명은 HEIGHT2와 WEIGHT2으로 하시오.

```
data class6;
    set class;
    if (height>60) then do;
        height2=height*1.1;
        weight2=weight*0.9;
    end;
run;

proc print data=class6;
    var height height2 weight weight2;
run;
```

height	height2	weight	weight2
69.0	75.90	112.5	101.25
56.5	.	84.0	.
65.3	71.83	98.0	88.20
62.8	69.08	102.5	92.25
63.5	69.85	102.5	92.25
57.3	.	83.0	.
59.8	.	84.5	.
62.5	68.75	112.5	101.25

만약 키가 60보다 크지 않은 학생의 경우 새로운 변수 HEIGHT2와 WEIGHT2는 결측치이다. 만약 이 때 새로운 변수의 값을 기존 값으로 하려면 다음과 같이 하면 된다.

```
data class6;
    set class;
        height2=height;
        weight2=weight;
    if (height>60) then do;
        height2=height*1.1;
        weight2=weight*0.9;
    end;
run;
```

Obs	height	height2	weight	weight2
1	69.0	75.90	112.5	101.25
2	56.5	56.50	84.0	84.00
3	65.3	71.83	98.0	88.20
4	62.8	69.08	102.5	92.25
5	63.5	69.85	102.5	92.25
6	57.3	57.30	83.0	83.00
7	59.8	59.80	84.5	84.50

연습문제 IF문 사용하기

UNIV.txt 자료 미국 대학 이름, 학교 종류(Liberal Arts, Univ.), SAT 점수, 입학률(acceptance rate), 일년간 학생들이 쓴 비용, 입학생 중 고등학교 때 10% 안에 든 학생 비율(%), 교직원의 박사 학위 소지 비율(%), 졸업하는 비율(%)이다.

①우선 SAS 데이터(이름은 UNIV)를 만드시오. 변수명은 출력 결과를 참조하시오.

```
proc print data=univ;
run;
```

NAME	TYPE	SAT	ACCEPT	SPEND	TOP10	PHD	GRADUATE
Amherst	Lib Arts	1315	22	26636	85	81	93
Swarthmore	Lib Arts	1310	24	27487	78	93	88
Williams	Lib Arts	1336	28	23772	86	90	93
Washington&Lee	Lib Arts	1234	29	17998	61	89	78
Grinnell	Lib Arts	1244	67	22301	65	79	73
Mount Holvoke	Lib Arts	1200	61	23358	47	83	83

출력 결과를 저장하시오. A:\APR02 폴더에 이름은 **OUT1.lst**로 하시오.

②SAT 성적이 1350 이상이거나 입학생 중 고등학교 때 10% 안에 든 학생 비율(TOP10)이 85% 이상이면 우수 학교라 하고 그렇지 않으면 비우수 학교라 정의한다. 우수 여부를 분류하는 변수를 하나 만드시오. 변수명은 GROUP1 이라 한다.

프로그램을 A:\APR02 폴더에 **PGM1.sas**로 저장하시오.

```
data univ1;
    set univ;
     이 부분을 채우시오.
run;

proc print data=univ1;
    var name sat top10 group1;
run;
```

▶

Obs	NAME	SAT	TOP10	group1
1	Amherst	1315	85	비우수
2	Swarthmore	1310	78	비우수
3	Williams	1336	86	우수
4	Washington&Lee	1234	61	비우수
5	Grinnell	1244	65	비우수
6	Mount Holyoke	1200	47	비우수
7	Colby	1200	52	비우수
8	Hamilton	1215	51	비우수
9	Bates	1240	58	비우수
10	Haverford	1285	71	비우수
11	Harvard	1370	90	우수

②변수 TOP10, PHD, GRADUATE 값의 평균을 구하시오. 변수명은 AVG1으로 하시오. 만약 AVG1의 값이 85이상이면 TOP10은 90, PHD는 85로 하시오. AVG1 변수가 출력될 때 아래와 같이 정수로 출력되게 하시오.

```
data univ1;
    set univ;
  이 부분을
  채우시오.
run;

proc print data=univ1;
    var name top10 phd avg1;
run;
```

Obs	NAME	TOP10	PHD	avg1
1	Amherst	90	85	86
2	Swarthmore	90	85	86
3	Williams	90	85	90
4	Washington&Lee	61	89	76
5	Grinnell	65	79	72
6	Mount Holyoke	47	83	71
7	Colby	52	75	70
8	Hamilton	51	86	74

프로그램을 A:\APR02 폴더에 **PGM2.sas**로 저장하시오.

참고

ID+1;의 의미는 변수 ID를 만들고 관측치마다 1씩 증가시키라는 것이다. ID+2;를 하면 어떻게 될까? 2, 4, 6, …으로 된다.

```
data univ2;
    set univ;id+1;
run;
```

▶

Obs	id	NAME
1	1	Amherst
2	2	Swarthmore
3	3	Williams
4	4	Washington&Lee
5	5	Grinnell
6	6	Mount Holyoke
7	7	Colby
8	8	Hamilton

3.2.3 IF 문(3) **IF (조건) THEN; 문장1(들); ELSE DO; 문장2(들); END;**

()의 조건이 성립하면 문장1을 그렇지 않은 경우 문장2를 실행한다. 그러나 이 문장은 IF문 2번 쓰는 것으로 해결되므로 몰라도 상관이 없다.

IF (조건) **THEN DO;** 문장**1**(들)**; END**
IF (반대 조건) **THEN DO;** 문장**2**(들)**; END;**

3.2.4 DO 문

DO UNTIL, DO WHILE 문이 있으나 사용 빈도가 적거나 고급 프로그램이 주로 사용되므로 본 강의에서는 생략하기로 한다.

DO 변수명=초기치 **TO** 말기치 **(BY** 증가분**);**
문장**(**들**);**
END;

다음 프로그램은 변수 1를 1부터 10까지 1씩 증가시키며 DO loop 안에 있는 문장을 반복 실행한다.

```
data do1;
    do i=1 to 10;
        j=i**3;
    end;
run;
proc print data=do1;
run;
```

▶

Obs	i	j
1	11	1000

근데 왜 마지막 한 개만 나오지. 다음과 같이 프로그램을 변경하면 10개 데이터가 나온다. OUTPUT의 의미는 현재 변수 값들을 데이터에 저장하라는 의미이다.

```
data do1;
    do i=1 to 10;
        j=i**3;
        output;
    end;
run;
proc print data=do1;
run;
```

▶

Obs	i	j
1	1	1
2	2	8
3	3	27
4	4	64
5	5	125
6	6	216
7	7	343
8	8	512
9	9	729
10	10	1000

다음은 일정한 숫자만큼 증가시키지 않는 경우 예제이다.

```
data do2;
    do i=1,2,5;
        j=i**3;
        output;
    end;
run;
proc print data=do2;
run;
```

▶

Obs	i	j
1	1	1
2	2	8
3	5	125

문자열을 사용할 수 있다. ^=의 의미는 아니라는 것이다. 즉 '남자'가 아닐 때...

```
data do2;
    do i='남자', '여자 ';
        if (i='남자') then k=0;
        if (i^='남자') then k=1;
        output;
    end;
run;
```

▶

Obs	i	k
1	남자	0
2	여자	1

세 개의 그룹(1, 2, 3)으로부터 남자 3명 여자 3명을 추출하여 IQ를 조사하였다.

그룹1		그룹2		그룹3	
남자	여자	남자	여자	남자	여자
125, 120, 115	113, 110, 135	123, 118, 130	100, 130, 130	110, 110, 115	115, 120, 110

위의 자료를 다음과 같이 SAS 데이터로 만들어 보시오. 단 DO 문을 사용하지 않는다면 어떻게 할까?

Obs	group	gender	r	iq
1	1	남자	1	125
2	1	남자	2	120
3	1	남자	3	115
4	1	여자	1	113
5	1	여자	2	110
6	1	여자	3	135
7	2	남자	1	123
8	2	남자	2	118
9	2	남자	3	130
10	2	여자	1	100
11	2	여자	2	130
12	2	여자	3	130
13	3	남자	1	110
14	3	남자	2	110
15	3	남자	3	115
16	3	여자	1	115
17	3	여자	2	120
18	3	여자	3	110

```
data do3;
    do group=1 to 3;
    do gender='남자', '여자 ';
    do r=1 to 3;
        input iq @@; output;
    end;
    end;
    end;
cards;
125 120 115 113 110 135 123 118 130
100 130 130 110 110 115 115 120 110
run;
proc print data=do3;
run;
```

연습문제 DO문 사용하기

아래와 같이 출력되게 DO 문을 이용하여 SAS 프로그램을 작성하시오. (1)을 출력하는 프로그램은 A:\APR08 폴더에 **PGM1.sas**로 저장하시오. (2)를 출력하는 프로그램은 A:\APR02 폴더에 **PGM2.sas**로 저장하시오.

(1)

Obs	i	j	k	l
1	2	a	1	1
2	2	a	2	4
3	2	a	3	27
4	2	b	1	1
5	2	b	2	4
6	2	b	3	27
7	2	c	1	1
8	2	c	2	4
9	2	c	3	27
10	4	a	1	1
11	4	a	2	4
12	4	a	3	27
13	4	b	1	1
14	4	b	2	4
15	4	b	3	27
16	4	c	1	1
17	4	c	2	4
18	4	c	3	27

(2)

Obs	i	k	l
1	1	1	1
2	3	1	1
3	3	2	4
4	3	3	27
5	7	1	1
6	7	2	4
7	7	3	27
8	7	4	256
9	7	5	3125
10	7	6	46656
11	7	7	823543

3.2.5 RETAIN 문 사용하기

SAS 데이터에서 새로운 변수에 초기치를 할당하기 위하여 사용된다. 만약 초기치를 할당하지 않으면 결측치(.)로 배정된다.

```
data sample1;
    input a @@;
    total=a+total;
    cards;
    1 3 5 7
run;
```

▶

Obs	a	total
1	1	.
2	3	.
3	5	.
4	7	.

다음과 같이 RETAIN 문을 사용하면 된다.

```
data sample1;
    input a @@;
    retain total 0;
    total=a+total;
    cards;
    1 3 5 7
run;
```

▶

Obs	a	total
1	1	1
2	3	4
3	5	9
4	7	16

연습문제 TETAIN문 사용하기

다음 자료의 평균을 구하는 프로그램을 RETAIN 문을 이용하여 작성하시오. 아래 프로그램은 A:\APR08 폴더에 **PGM3.sas**로 저장하시오.

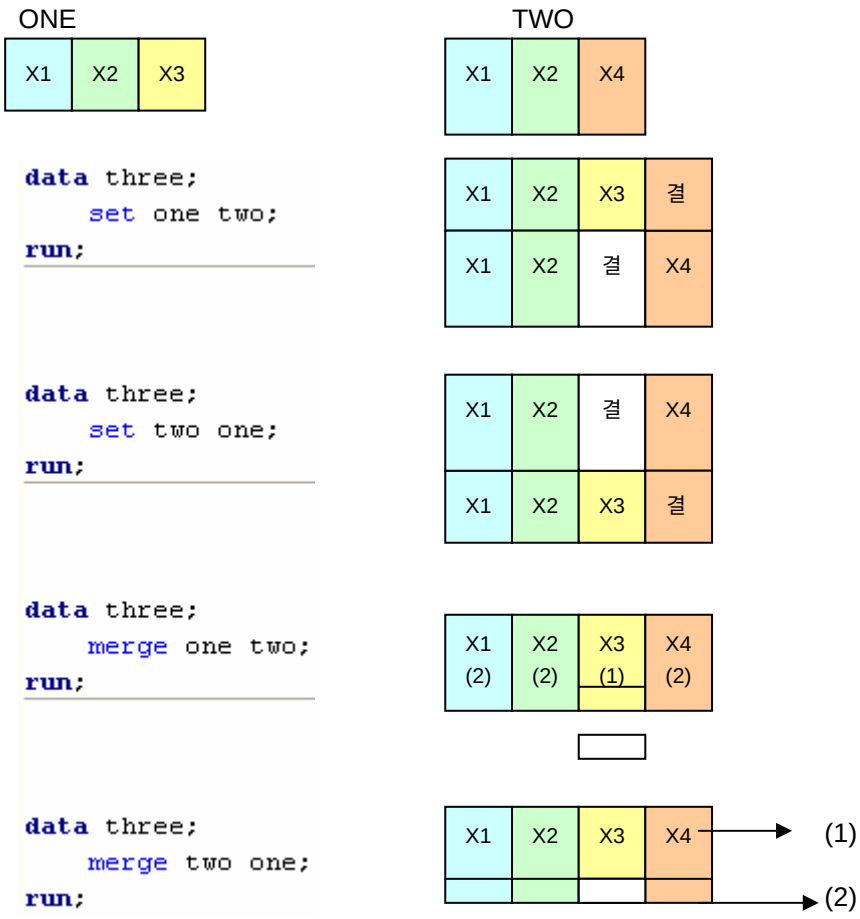
```
data sample2;
    input x @@; n+1;
  이 부분을 채우시오.


    cards;
    1 3 5 7 9 11 12 13 14 15
run;

proc print data=sample2;
run;
```

Obs	x	n	total	avg
1	1	1	1	1.00000
2	3	2	4	2.00000
3	5	3	9	3.00000
4	7	4	16	4.00000
5	9	5	25	5.00000
6	11	6	36	6.00000
7	12	7	48	6.85714
8	13	8	61	7.62500
9	14	9	75	8.33333
10	15	10	90	9.00000

3.3 데이터 합치기

3.3.1 SET 과 MERGE

SAS 데이터를 합치는 방법으로 SET 문과 MERGE 문이 있다. SET은 두 데이터를 세로로 합치는 것이고 MERGE 문은 가로로 합치게 된다.



연습문제 데이터 합치기

```
data one;
    input x1 x2 x3;
    cards;
    1 2 3
    4 5 6
    7 8 9
run;
```

```
data two;
    input x1 x4;
    cards;
    0 1
    2 3
    4 5
    6 7
run;
```

```
data three;
    input x1 x2 x5;
    cards;
    0 1 0
    2 1 2
run;
```

SAS를 실행하지 말고 다음 결과를 예상해 보자. 그리고 SAS를 이용하여 확인하자. 세 개 이상의 데이터 합치기도 동일하게 실행된다.

```
data four;
    set one two;
run;

data four;
    set one two three;
run;

data four;
    set three one;
run;
```

```
data four;
    merge one two;
run;

data four;
    merge one two three;
run;

data four;
    merge three one;
run;
```

3.3.2 같은 변수에 의해 합치기

두 데이터에 동일한 변수(예: ID)가 있어 이 변수에 의해 두 데이터를 합치려고 한다면 MEREGE 문을 사용하려고 할 것이다.

```
data one;
    input id iq;
    cards;
    1 135
    2 110
    5 120
    7 150
run;
```

```
data two;
    input id gender $ income;
    cards;
    2 m 15
    1 f 30
    4 m 20
    7 f 25
    5 f 40
run;
```

그러나 MERGE 문을 사용하면 동일한 변수명을 가지게 되면 뒤에 데이터로 엎어 버린다.

```
data three;
    merge one two;
run;

proc print data=three;
run;
```

Obs	id	iq	gender	income
1	2	135	m	15
2	1	110	f	30
3	4	120	m	20
4	7	150	f	25
5	5	.	f	40

이런 경우 BY 문을 사용하면 된다.

```
data three;
    merge one two;
    by id;
run;
```

그러나 출력 결과에 변화가 없다. 이상하다. 언제나 로그(log) 창을 살펴보는 습관을 갖자. 오류 메시지는 데이터 TWO가 정렬(sort)되어 있지 않았다고 한다.

```
1059  data three;
1060      merge one two;
1061      by id;
1062  run;

ERROR: BY variables are not properly sorted on data set WORK.TWO.
id=2 iq=110 gender=m income=15 FIRST.id=1 LAST.id=1 ERROR=1 N=2
```

SAS에서 BY 문을 쓰려면 BY 문에 지정된 변수에 의해 그 데이터는 정렬되어 있어야 한다.그러므로 다음 절차를 따르면 문제가 해결된다. PROC SORT에 대해서는 다음에 자세히 다루기로 한다.

```
proc sort data=one;
    by id;
run;
proc sort data=two;
    by id;
run;
data three;
    merge one two;
    by id;
run;
proc print data=three;
run;
```

Obs	id	iq	gender	income
1	1	135	f	30
2	2	110	m	15
3	4	.	m	20
4	5	120	f	40
5	7	150	f	25

연습문제 MERGE문 (by)

DATA0301.txt은 3월 1일 조사 데이터이고(이름과 생년월일) DATA0401은 4월 1일 조사 데이터이다. 4월 1일 자료는 3월 1일에 조사된 내용 중에 수정할 부분만을 조사한 자료이다. 적절한 프로그램을 적성하고 A:\APR08 폴더에 **PGM4.sas**로 저장하시오.

```
data one;
     이 부분을 채우시오.

cards;
You     12/31/1978
Kim2    04/17/1978
run;
```

```
data two;
    이 부분을 채우시오

cards;
Kwon 01/03/1961
Kim 04/05/1981
You 10/13/1978
Kim2 07/17/1978
run;
```

두 데이터를 적절히 합쳐 DATA THREE를 만드시오. DATA Three는 다음과 같다.

```
options nodate nonumber;
title 'DATA three';
proc print data=three;
run;
```

DATA three

Obs	bithday	name
1	04-05-81	Kim
2	04-17-19	Kim2
3	01-03-61	Kwon
4	12-31-01	You

OPTIONS 문장의 nodate는 출력 창에 날짜 출력하지 말라는 것이고 nonumber는 페이지 번호 출력하지 말라는 것이다.

TITLE 문장은 출력 결과의 제목을 지정한다. 여러 줄을 사용하고자 할 때는 TITLE1, TITLE2, … 사용하면 된다. 제목을 없애려면 TITLE;을 사용하면 된다.

설문데이터

설문조사 데이터의 경우 문항 당 한자리 입력이 정형화 되어 있으므로 공백 없이 입력하는 것이 편리하고 코딩의 오류가 없다.

Q1 □여자(1) 남자(2)

Q2 □재수하지 않음(1) 재수 이상(2)

선택 보기 문항▶각 한 자리씩 부여

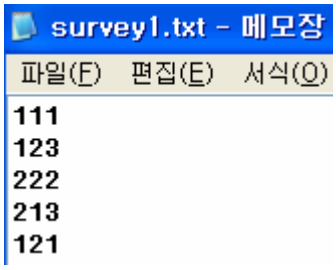
Q3 □출신 고등학교 소재지? 대전(1) 충남(2) 서울(3) 경기(4) 그 외 지역(5)

이런 자료의 경우 다음과 같이 읽는 것이 일반적이다.

```
data one;
    input (q1-q3) (1.);
    cards;
111
123
222
213
121
run;

proc print data=one;
run;
```

▶

Obs	q1	q2	q3
1	1	1	1
2	1	2	3
3	2	2	2
4	2	1	3
5	1	2	1




```
data one;
    infile 'c:\temp\survey1.txt';
    input (q1-q3) (1.);
run;

proc print data=one;
run;
```

3.3.3 필요한 변수만 가져오기

두 데이터에 동일한 변수(예: ID)가 있어 이 변수에 의해 두 데이터를 합치려고 한다면 MEREGE 문을 사용하려고 할 것이다.

```
data one;
    input name $ iq;
cards;
Kwon 150
Kim 120
You 125
Kim2 130
run;
```

```
data two;
    input name $ gender $ income;
cards;
Kwon M 500000
Kim F  350000
You M  400000
Kim2 M 300000
run;
```

아래 프로그램은 DATA two로부터 GENDER라는 변수만 가져와 ONE에 merge한다. 만약 소득

변수도 가져오기 원하면 two(keep=gender income); 이렇게 사용하면 된다. 만약 필요 없는 변수를 제외하려면 two(drop=income); 이렇게 사용하면 된다.

```
data three;
    merge one two(keep=gender);
run;

title;
proc print data=three;
run;
```

▶

Obs	name	iq	gender
1	Kwon	150	M
2	Kim	120	F
3	You	125	M
4	Kim2	130	M

다음 프로그램은 두 개의 데이터를 SET 문을 사용하여 합치는데 two 데이터에서 처음 2개 데이터만 가져올 때 사용된다.

```
data one;
    input name $ iq;
cards;
Kwon 150
Kim 120
You 125
Kim2 130
run;
```

```
data two;
    input name $ iq;
cards;
Kwon2 150
Kim3 120
You2 125
run;
```

```
data three;
    set one two(obs=2);
run;

title;
proc print data=three;
run;
```

▶

Obs	name	iq
1	Kwon	150
2	Kim	120
3	You	125
4	Kim2	130
5	Kwon2	150
6	Kim3	120

DROP과 KEEP 문장은 변수를 원하는 변수를 제거하거나 그 변수들만 저장하고자 할 때 사용한다. DELETE는 관측치를 제외할 때 사용하는 것임을 기억하기 바란다.

```
data one;
    input x1 x2 x3 x4;
data two;
    input x1 x2 x5;
```

▶

```
data three;
    set one two;
    keep x3 x4 x5;
run;
```

동일

```
data three;
    set one two;
    drop x1 x2;
run;
```