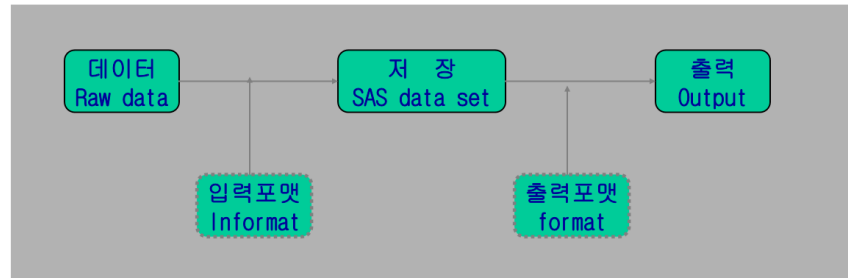


5-1강

[참고] 입력포맷과 출력포맷의 처리절차



자료	입력포맷	저장	출력포맷	출력
abcdefghij	\$10.	abcdefghij	\$5.	abcde
1.23456789	10.	1.23456789	4.2	1.23
1,100,100	COMMA9.	1100100	DOLLAR10.	\$1,100,100
01JAN1998	DATE9.	13880	YYMMDD8.	98-01-01

1.23456789에 대해 4.2로 출력포맷을 결정하면 1.23가 나오는데, 이는 소숫점을 포함한 것이다.

1. 데이터입력 구분자

- 구분자 지정
 - 구분자 : DLM
 - DELIMITER, DLMSTR=문자열
 - 여러 개의 문자를 지정
 - 연속된 구분자는 1개로 간주
 - 기본 구분자는 공백
 - 정밀한 구분 : DSD
 - 연속된 구분자는 결측값 포함
- 사용 위치
 - INFILE fileref DLM=' ab , ' dsd ;
 - FILE fileref DLM=' ab , ' dsd ;
- 확인 팁
 - delimiter 는 공백으로 변환한다
 - dsd 가 있는 경우 공백사이에 . 을 추가
 - 공백이 둘이면 공백 . 공백 으로
 - 공백이 셋이면 공백 . 공백 . 공백

- 문자열 “ ”로 표시
- 기본 구분자는 ‘,’

- Tab구분자(엑셀) : EXPANDTABS

예제: 구분자

```
DATA nums1;
  INFILE CARDS DLM='ab';
  INPUT x y z;
DATALINES;
1aa2ab3
4b5bab6
7a8b9
;
RUN;

PROC PRINT;
RUN;
```

```
DATA nums2;
  INFILE CARDS DSD DELIMITER='ab';
  INPUT x y z ;
CARDS;
1aa2ab3
4b5bab6
7a8b9
;
RUN;

PROC PRINT;
RUN;
```

구분자 : a b					
1	2	3	1	2	3
4	5	6	4	5	6
7	8	9	7	8	9

Obs	x	y	z
1	1	2	3
2	4	5	6
3	7	8	9

구분자 : a b 정밀한 구분									
1 2 3					1 . 2 . 3				
4 5 6					4 5 . . 6				
7 8 9					7 8 9				

OBS	x	y	z
1	1	.	2
2	4	5	.
3	7	8	9

부족한 자료

- MISCOVER : 데이터가 부족한 변수는 결측값으로 할당해서 다음 레코드를 읽는 것을 방지
- TRUNCOVER : 데이터값이 부족한 부분에 공백을 삽입하라

- FLOWOVER : 데이터값이 부족하면 다음레코드를 읽어서 INPUT Buffer에 넣어라
- STOPOVER : 데이터값이 부족하여 결측값이 생기면 데이터 생성을 중단한다.

※ 네 가지 중에서 한가지만 지정할 수 있으며, default는 FLOWOVER

예 5.12) MISSOVER 옵션의 사용

```
DATA miss;
INFILE CARDS MISSOVER;
INPUT a b c;
CARDS;

1 2 3
4 5
6 7 8
9 0 1 2
3 4 5
;

RUN;
```

[프로그램 내부에 데이터를 입력하는 경우
일반적으로 INFILE 명령문을 사용할 필요
가 없지만

INFILE 명령문에 있는 옵션을 사용하기 위
해서

INFILE CARDS 명령문을 사용할 수도 있
다.]

예) Data가 부족한 경우의 처리

```
DATA one
INFILE CARDS 옵션;
INPUT a b c;
CARDS;

1 2 3
4 5
6 7 8
9 0 1 2
3 4 5
;

RUN;
```

옵션: FLOWOVER

	a	b	c
1	1	2	3
2	4	5	6
3	9	0	1
4	3	4	5

옵션: MISSOVER

	a	b	c
1	1	2	3
2	4	5	.
3	6	7	8
4	9	0	1
5	3	4	5

옵션: STOPOVER

	a	b	c
1	1	2	3

예 5.13) DLM 또는 DSD 옵션의 사용

```
DATA score;
INFILE CARDS DLM=',';
INPUT t1 t2 t3;

CARDS;
91,87,95
97,,92 /*사이 공백 비교*/
1,1,1
;
RUN;
```

	t1	t2	t3
1	91	87	95
2	97	92	1

DLM은 연속된 구분자는 하나로 처리한다.

```
DATA score;
INFILE CARDS DLM=',';
INPUT t1 t2 t3;

CARDS;
91,87,95
97, ,92 /*사이 공백 비교*/
1,1,1
;
RUN;
```

VIEWTABLE: Work.Score			
	t1	t2	t3
1	91	87	95
2	97	.	92
3	1	1	1

DLM에 대하여 공백을 .으로 처리한 것이다.

```
DATA score;
INFILE CARDS DSD;
INPUT t1 t2 t3;

CARDS;
91,87,95
97,,92
1,1,1
;
RUN;
```

VIEWTABLE: Work.Score			
	t1	t2	t3
1	91	87	95
2	97	.	92
3	1	1	1

- DSD 옵션

1. 콤마(,)를 구분자로 사용한다. 콤마 이외 구분자는 DLM옵션과 같이 사용
2. 연속된 구분자들 사이는 결측값으로 처리
3. 인용부호안에 있는 문자는 구분자로 인식 안함. 인용부호는 자릿값으로 받아들이지 않음

구분자; DLM 옵션의 사용

```
DATA case1;
INFILE CARDS dlm='&$*, ' ;
INPUT a b c ;
CARDS;
11&$12, 13
2$$$22* 23
3* 32,33
;
RUN;
```

	a	b	c
1	11	12	13
2	2	22	23
3	3	32	33

- 지정된 문자들의 조합으로 이루어진 모든 문자열은 하나의 구분자가 된다.

2. 할당문과 선언문

- 할당문

사용자는 다음과 같이 새로운 변수 혹은 기존 변수를 생성하거나 값을 바꿀 수 있다.

```
Variable = Expression
```

- Example

```
major = "통계학"          /*문자형 변수의 값을 할당하는 경우*/
mean = (test1 + test2)    /*기존 변수와의 연산을 통해 값을 할당하는 경우*/
x = log(y)                /*함수를 이용하여 값을 할당하는 경우*/
```

- 선언문

- 지정된 변수에 대해 특정한 기능을 표시한다.
- LENGTH, FORMAT, LABEL, DROP, KEEP, RENAME 등이 있음.

예 6.1) 다양한 할당문의 사용

```
DATA exam;
INPUT id $ name $ mid final; x=30;          /* Numeric Constant*/
major='통계학';                             /* Character Constant */
major_id='STA'||id;                          /* Concatenation */
```

```

mean=(mid+final)/2;                                /* Arithmetic Expression */
y=mid**3;                                           /* Exponentiation */
LABEL name='이름' mid='중간성적' final='기말성적';

CARDS;
001 김철수 10 40
002 이영희 15 10
001 강민호 50 15
001 박지수 20 .
;
RUN;

```

VIEWTABLE: Work.Exam									
	id	이름	중간성적	기말성적	x	major	major_id	mean	y
1	001	김철수	10	40	30	통계학	STA001	25	1000
2	002	이영희	15	10	30	통계학	STA002	12,5	3375
3	001	강민호	50	15	30	통계학	STA001	32,5	125000
4	001	박지수	20	.	30	통계학	STA001	.	8000

- [연결연산자(II)]는 여러 개의 문자열과 문자변수들을 연결하는 연산자로 공백문자는 제거되지 않는다]

3. 함수의 이용: 숫자함수

- 여러 가지 기능을 갖는 함수를 이용하여 복잡한 형태의 연산이나 필요로 하는 변수 변환을 손쉽게 수행할 수 있다

```
Variable = function_name( [ [ arg1 ] [,arg2, ...] ] )
```

- 숫자 함수

```

y = sum(OF x1-x10)          /*변수 x1에서 x10까지의 합을 구하여 변수 y에 할 당한다..*/
y = mean(x1, x2, x3)        /*변수 x1, x2, x3의 평균을 계산하여 변수 y에 할 당한다..*/

```

함수의 이용: 문자함수와 날짜함수

형 식	기 능
<code>compress(arg, chars)</code>	arg의 내용 중 chars로 주어진 문자를 제거
<code>left(arg)</code>	주어진 arg의 앞 공백을 모두 제거
<code>length(arg)</code>	주어진 arg의 문자열 길이를 계산
<code>substr(arg, p, n)</code>	arg의 p번째 문자로부터 n개의 문자열을 선택
<code>translate(arg, to, from)</code>	주어진 arg의 문자열에서 from부분을 to로 변환
<code>trim(arg)</code>	주어진 arg의 오른쪽 공백문자를 모두 제거
<code>DAY(date)</code>	Date의 값을 구한다.(1~31)
<code>MDY(month, day, year)</code>	년, 월, 일에 해당하는 SAS 날짜값을 계산한다.
<code>MONTH(date)</code>	Date의 월을 구한다.(1~12)
<code>YEAR(date)</code>	Date의 년도를 구한다.
<code>WEEKDAY(date)</code>	Date의 요일값을 구한다.(1~7)
<code>TODAY()</code>	현재 날짜에 해당하는 SAS날짜값을 구한다.
<code>INTNX('interval',from,n)</code>	from부터 n interval이후 날짜를 계산한다.

SAS 날짜 값을 만드는 함수

- MDY (month, day, year);

세 개의 숫자변수 (도는 숫자를 사용하여 SAS날짜변수를 만듦)

```
DATA mdyex;
  INPUT mon day year;
  newdate = MDY(mon, day, year);
  FORMAT newdate YYMMDD10.;
CARDS;
12 11 1997
03 18 1998
;
RUN;
```

- TODAY();

컴퓨터 시스템 상의 현재 날짜를 변환

```
DATA todayex;
  SET mdyex;
  today = TODAY();
  diff = today - newdate;
  FORMAT today newdate YYMMDD10.;
RUN;
```

- YYQ(year, quarter);

연도와 분기를 갖는 두 개의 숫자변수를 이용하여 날짜변수를 만듦

자주 사용하는 공백제거함수

- COMPRESS

COMPRESS 함수는 앞쪽에 있는 문자변수 또는 문자열 중에서 뒤쪽의 문자변수 또는 문자열에 포함되어 있는 모든 문자를 제거

예) COMPRESS('K.Y.M # 1991'#,;') → KYM1991

- TRIM

TRIM 함수는 문자열 뒤의 공백을 제거

- CATX

CATX 함수는 결합대상 string 들의 앞 뒤의 공백을 제거하고 구분자를 넣어 문자열을 결합

- STRIP

STRIP 함수는 string 앞과 뒤의 공백을 제거

예6.2) 다양한 함수의 이용

```
DATA exam1;
INPUT name $ 1-8 mid final enterm enterd;
    mid=INT(mid);                                /* Numeric Function */
    max_mf=MAX(mid,final,30);
    name=COMPRESS(name);                          /* Character Function */
    f_name=SUBSTR(name,1,2);
    ent_day=MDY(enterm,enterd,1999);              /* Date Function */
    ent_week=WEEKDAY(ent_day);
    FORMAT ent_day yymmdd6.;

CARDS;
김 철 수 10.1 40 11 22
이 영희 15.7 10 11 29
강민호 50.3 15 12 05
박지 수 20.8 . 12 15
;
RUN;
```

VIEWTABLE: Work.Exam1									
	name	mid	final	enterm	enterd	max_mf	f_name	ent_day	ent_week
1	김철수	10	40	11	22	40	김	991122	2
2	이영희	15	10	11	29	30	이	991129	2
3	강민호	50	15	12	5	50	강	991205	1
4	박지수	20	.	12	15	30	박	991215	4

예6.3) MEAN과 SUM 함수의 사용

```
DATA score;
INPUT name $ (x1-x3) (1.) y1-y3;
total = SUM(OF x1-x3 y1-y3);
average = MEAN(OF x1-x3 y1-y3);
logx = LOG(x1); sqrtx = SQRT(x1);
intmean = INT(average);

CARDS;
김철수 551 2 1 3
최민지 .31 4 5 1
이영희 153 2 . 2
오인수 412 4 . .
;
RUN;

PROC PRINT DATA=score;
RUN;
```

OBS	name	x1	x2	x3	y1	y2	y3	total	average	logx	sqrtx	intmean
1	김철수	5	5	1	2	1	3	17	2.83333	1.60944	2.23607	2
2	최민지	.	3	1	4	5	1	14	2.80000	.	.	2
3	이영희	1	5	3	2	.	2	13	2.60000	0.00000	1.00000	2
4	오인수	4	1	2	4	.	.	11	2.75000	1.38629	2.00000	2