

Chapter 5. 데이터 단계에서 사용되는 명령문

할당문과 선언문

❖ 할당문

- 자료값을 바꾸거나 새로운 변수를 생성

Variable = Expression;

- 등호 왼쪽: 기존의 변수 또는 새로운 변수 이름을 사용
- 등호 오른쪽: 사칙연산을 포함한 여러 가지 형태의 표현식을 사용

다양한 할당문의 사용

```
DATA mysas.exam;  
INPUT id $ name $ mid final;  
CARDS;  
001 김철수 10 40  
002 이영희 15 10  
001 강민호 50 15  
001 박지수 20 .  
;  
RUN;  
DATA mysas.exam1;  
SET mysas.exam;  
x=30;  
major="통계학"; major_id="STA"||id; mean=(mid+final)/2; y=mid**3;  
LABEL name="이름" mid="중간성적" final="기말성적";  
RUN;
```

할당문과 선언문

- 문자열
 - 인용부호(“”)로 문자열의 좌우를 닫아주어 변수 이름과 구별
- 연결연산자(||)
 - 여러 개의 문자열과 문자변수들을 연결하는 연산자
 - 문자열 또는 문자변수에 포함되어 있는 공백문자는 제거되지 않음
 - 공백문자를 제거하기 위해서는 TRIM 함수를 사용

함수의 이용

- SAS내 내장된 함수를 이용하면 복잡한 형태의 연산을 쉽게 수행가능

몇 가지 SAS 함수의 기능

유형	함수 및 사용형식	함수의 기능
숫자 함수	INT(arg)	arg의 값을 정수화
	LOG(arg)	arg의 자연대수를 구함
	LOG10(arg)	Arg의 사용대수를 구함
	MAX(arg1,arg2,...)	결측값이 아닌 arg1, arg2, ... 중에서 최대값을 구함
	MEAN(arg1,arg2,...)	결측값이 아닌 arg1, arg2, ... 들의 평균을 구함
	SUM(arg1,arg2,...)	결측값이 아닌 arg1, arg2, ... 들의 합을 구함
	SQRT(arg)	arg의 제곱근을 구함
날짜 함수	DAY(date)	date의 일을 구함(1~31)
	INTNX('interval',from,n)	from으로부터 n interval 이후의 날짜를 계산
	INTCX('interval',from,to)	from으로부터 to사이의 interval을 계산
	MDY(month,day,year)	년, 월, 일에 해당하는 SAS 날짜 값 계산
	MONTH(date)	date의 월을 구함
	QTR(date)	date의 분기를 구함
	WEEKDAY(date)	date의 요일값을 구함(1~7)



함수의 이용

몇 가지 SAS 함수의 기능		
유형	함수 및 사용형식	함수의 기능
문자 함수	COMPRESS(arg, characters)	Arg 내용 중 characters에 주어진 문자제거 문자가 주어지지 않으면 공백문자 모두 제거
	LEFT(arg)	Arg 앞쪽에 있는 공백문자 모두 제거 문자열을 왼쪽으로 정렬
	LENGTH(arg)	문자열 arg의 길이를 계산 결측값에 대해서는 1의 값을 가짐
	SUBSTR(arg, position, n)	Arg의 내용 중 주어진 위치로부터 시작하여 n번째까지의 문자열을 값으로 가짐
	TRANSLATE(source, to-1, from-1, ...)	'from'에 주어진 문자를 'to'에 주어진 문자로 바꿈
	TRIM(arg)	Arg의 오른쪽 공백문자를 모두 제거
	UPCASE(arg)	Arg의 모든 내용을 대문자로 바꿈

함수의 이용

다양한 함수의 사용

```
DATA mysas.finexam;  
INPUT name $ 1-8 mid final enterm entered;  
mid=INT(mid);  
max_mf=MAX(mid,final,30);  
name=COMPRESS(name);  
f_name=SUBSTR(name,1,2);  
ent_day=MDY(enterm,entered,1999);  
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;
```

함수의 이용

- MEAN/SUM 함수

MEAN(var1,var2,...) 또는 MEAN(OF var1 var2 var3 ...)

- 평균/합계를 계산하는 함수
- OF 명령어: 변수 이름을 생략하여 사용하고자 하는 경우 사용
- 결측값이 아닌 자료값만을 가지고 연산을 수행

MEAN과 SUM함수의 사용

```
DATA mysas.avescore;  
INPUT name $ (x1-x3)(1.) y1-y3;  
total=SUM(OF x1-x3 y1-y3);  
average=MEAN(OF x1-x3 y1-y3);  
CARDS;  
김철수 551 2 1 3  
최민지 .31 4 5 1  
이영희 153 2 . 2  
오인수 412 4 . .  
;  
RUN;
```

조건문

❖ IF-THEN 명령문

- 어떤 할당문을 모든 개체에 적용하는 대신 특정 조건에 부합하는 개체에 대해서만 사용하고자 하는 경우 사용

IF condition THEN action;

- condition: 조건 논리식
- action: 조건식이 참인 경우에만 수행되는 할당문을 표현
- 조건식에는 대부분 비교연산자가 사용
 - 3장의 기호 또는 약어를 사용
 - Ex) IF final EQ . THEN final=10;

IF-THEN 명령문의 사용

```
DATA mysas.exam2;  
SET mysas.exam;  
IF final=. THEN final=10;  
IF (mid+final)>=50 THEN score1="P";  
IF mid>=30 or final>=30 THEN score2="P";  
IF score1=" " THEN  
DO;  
score1="F";  
score2="F";  
END; RUN;
```


조건문

- IN 명령어
 - IF 조건문에 여러 개의 자료값을 나열하라 때 매우 유용하게 사용
 - 콤마 또는 하나 이상의 공백으로 구분하여 자료값을 나열
 - 문자변수에 대해서는 각 문자열을 인용부호(" ")로 닫아주어야 함

IN 명령어의 사용

```
DATA mysas.exam3;  
SET mysas.exam;  
IF mid IN (10 15) THEN score1="C";  
IF mid IN (20, 15) THEN score1="C";  
IF name IN ("이영희", "박지수") THEN sex="F";  
IF name IN ("김철수", "강민호") THEN sex="M";  
RUN;
```

조건문

❖ IF-THEN-ELSE 명령문

- ELSE 명령문: 바로 위에 있는 조건식이 참이 아닌 경우에만 수행

IF-THEN-ELSE 명령문의 사용

```
DATA mysas.exam4;  
SET mysas.exam;  
IF mid>=30 THEN score1="A";  
ELSE IF mid>=20 THEN score1="B";  
ELSE score1="C";  
IF final>=30 THEN score2="P";  
ELSE score2="F";  
RUN;
```

조건문

❖ 관찰개체의 삭제: DELETE 명령문

- 데이터를 삭제하는 명령문
- IF 명령문과 DELETE 명령문을 함께 사용하면 특정한 조건에 맞는 데이터만 제거하는 작업을 쉽게 수행

DELETE 명령문의 사용

```
DATA mysas.person1;  
SET mysas.person;  
IF dept='math' OR dept='econ' THEN DELETE; (또는 IF dept='stat';)  
RUN;
```

날짜 변수

- 날짜형 입력포맷
 - 1960년 1월 1일을 기준으로 그 때부터의 날수로 바꾸어서 데이터 셋에 저장

날짜형 입력포맷, 함수, 출력포맷

```
DATA mysas.hire;  
INPUT name $6. +1 bdate DATE7. +1 hired MMDDYY8.;  
Hiredate=hired+(365.25*3);  
Hireqtr=QTR(hiredate);  
IF hired>'01jan94'D THEN new="YES";  
FORMAT bdate MMDDYY8. hired YYMMDD8. hiredate WEEKDATE17.;  
CARDS;  
김철수 01jan60 09-15-90  
최민지 05oct49 01-24-92  
이영희 18mar88 10-10-93  
오인수 29feb80 05-29-94  
;  
RUN;
```

- DATE7.과 MMDDYY8.이라는 입력포맷으로 두 변수 bdate와 hired를 읽어들이
- 산술연산식에 의해 변수 hired에 3년을 더하여 변수 hiredate를 생성
- QTR을 이용하여 변수 hireqtr을 생성
- 날짜상수를 이용하여 비교연산식을 사용
 - 날짜상수는 'ddmmmyy'D 형태로 지정

문자변수와 LENGTH 명령문

- LENGTH 명령문 : SAS 데이터셋에 변수가 저장될 때 사용되는 바이트 수를 지정
 - 주의 : 변수의 길이를 지정하는 것이 아님
 - 문자변수 : 영문 한 글자는 1바이트, 한글 한 글자는 2 바이트

LENGTH variable <\$> length ...;

- \$: 변수가 문자형임을 지정하는 옵션
- length : 바이트 수로 숫자형에 대해서는 2~8, 문자형에서는 1~32767까지 지정
- 프로그램에서 LENGTH 명령문 다음에 나타나는 변수에만 영향을 줌

LENGTH 명령문의 사용

```
DATA mysas.length;  
LENGTH name $ 9 grade $ 6;  
INPUT name$ score;  
IF 1<=score <= 3 THEN grade='LOW';  
ELSE IF 4<=score<=7 THEN grade='MIDDLE';  
ELSE IF 8<=score<=10 THEN grade='HIGH';  
CARDS;  
HyunCheol 7  
YongChan 10  
MinHee 3  
;  
RUN;
```

- LENGTH 명령문에 의해 변수의 길이가 미리 지정되어 있지 않으면 INPUT 명령문에서 지정된 변수 name에 대해서는 디폴트인 8이 길이가 되고, grade에 대해서는 이 변수의 값으로 처음나타나는 LOW에 의해 길이가 3이 됨.

RETAIN 명령문

❖ RETAIN 명령문

- 어떤 변수의 값을 바로 전 단계에서의 값을 유지시키고자 할 때 사용
- 개체 간의 자료값 비교나 연산에서 유용하게 사용

RETAIN variable-1 [initial-value-1];

- 특별한 지정이 없으면 RETAIN 명령문에 지정된 변수는 숫자변수
- 문자변수로 지정할 때는 적절한 길이의 문자형 초기값을 지정
 - 변수의 초기값을 지정하지 않으면 초기값을 결측값이 됨

RETAIN 명령문의 사용

```
DATA mysas.case;  
INPUT name$ x @@;  
RETAIN max_x 0 y 0 oldname "***";  
max_x=MAX(max_x,x);  
Y=y+x*x;  
IF name=oldname THEN case="Old";  
ELSE case="New";  
oldname=name;  
DROP oldname;  
CARDS;  
AAA 1 BBB 4 BBB 8 BBB 3  
CCC 2 CCC 5 DDD 6  
;  
RUN;
```

ARRAY 명령문

❖ ARRAY 명령문

- 일련의 변수들을 배열의 원소들로 지정하는 것
- 여러 개의 변수들에 대해서 동일한 작업을 반복할 필요가 있을 때 ARRAY 명령문과 DO-END 명령문을 함께 이용하여 효율적으로 작업을 수행

ARRAY name {#} <\$> variable-list;

ARRAY 명령문의 사용

```
DATA mysas.tscore;  
INPUT name$ math stat eng kor art;  
ARRAY tscore [5] math stat eng kor art;  
DO i=1 TO 5;  
IF tscore(i)=9 THEN tscore(i)=.;  
END;  
CARDS;  
김철수 5 5 1 2 1  
최민지 9 3 1 4 5  
이영희 1 5 3 2 9  
오인수 4 1 2 4 9  
;  
RUN;
```

- 다섯 개의 변수 math, ..., art는 ARRAY 명령문에 의해 tscore라는 배열의 원소로 지정
- 이 배열에 포함된 어느 변수의 자료값이 9가 되면 이를 결측값으로 바꿈
- 즉, tscore(1) = math, tscore(2) = stat, ..., tscore(5) = art를 의미

기타 명령문들

❖ 논리문의 이용

- 논리문: 표현식이 참이면 1, 거짓이면 0을 가지게 되는 일종의 조건식

논리문의 사용

```
DATA logical;  
INPUT age path;  
group=(age > 0)+(age<=35);  
IF path THEN dept='STAT';  
CARDS;  
...
```

❖ LIST 명령문

- 현재 처리되고 있는 개체에 대한 입력 데이터를 LOG 윈도우에 출력

LIST 명령문의 사용

```
DATA list;  
INPUT trt crop herb;  
IF trt=. THEN LIST;  
CARDS;  
...
```


기타 명령문들

❖ RETURN 명령문

- 현재 처리되고 있는 개체에 대해 그 이후의 작업을 수행하지 않고 데이터-단계의 처음으로 돌아가서 다음 개체에 대한 작업을 수행하고자 할 때 사용

논리문의 사용

```
DATA return;  
INPUT x y z;  
IF x=y THEN RETURN;  
x=y+z;  
xsquare=x*x;  
CARDS;  
...
```

❖ LINK 명령문

- 데이터 단계를 종료하고 현재의 작업내용만으로 데이터셋을 생성하고자 할 때 사용

LIST 명령문의 사용

```
DATA stop;  
INPUT x y z;  
IF _N_>250 THEN STOP;  
CARDS;  
...
```

기타 명령문들

❖ GOTO 명령문

- 프로그램의 수행 도중 특정한 부분으로 건너 띄어 작업을 계속 수행하고자 할 때 사용

```
GOTO label;  
...  
label : SAS 프로그램; ...
```

- label
 - GOTO 명령문의 수행에 의해 작업의 수행이 이동되는 목적지를 식별해 주기 위해 사용자가 정의한 이름
 - 현재 사용하고 있는 데이터-단계 내에 위치해야 함

GOTO 명령문의 사용

```
DATA goto;  
INPUT year weight damage plant;  
IF year=78 THEN GOTO sub;  
total = damage/plant;  
RETURN;  
sub: weight = weight*1.3;  
plant = plant*2;  
CARDS;  
...
```

기타 명령문들

❖ LINK 명령문

- 데이터 단계의 특정한 부분을 하나의 블록으로 설정하고 해당 블록과 연결하여 작업을 수행하고자 할 때 사용
- 하나의 블록은 label:으로 시작하여 RETURN;으로 끝나야 함

논리문의 사용

```
DATA link;  
INPUT year weight damage plant;  
IF year=78 THEN LINK sub;  
total = damage/plant;  
RETURN;  
sub: weight = weight*1.3;  
plant = plant*2;  
CARDS;  
...
```

기타 명령문들

❖ ABORT 명령문

- 데이터 단계를 종료하고 데이터 셋을 생성하거나 대체하지 않도록 하고자 할 때 사용

ABORT 명령문의 사용

```
DATA abort;  
INPUT id score;  
IF _ERROR_ THEN ABORT;  
CARDS;  
111 50  
abc 70  
333 80  
;  
RUN;
```

연습문제

❖ 3명의 학생에 대해 6번의 퀴즈를 본 점수를 기록한 다음 데이터로부터 다음과 같은 새로운 변수들을 포함하는 SAS 데이터 셋을 생성하여라.

- Q1 ~ Q6의 최소값을 가지는 변수: small
- Q1 ~ Q6의 최대값을 가지는 변수: big
- Q1 ~ Q6의 총합을 가지는 변수: total
- Q1 ~ Q6의 평균값을 가지는 변수: mean

id	Q1	Q2	Q3	Q4	Q5	Q6
001	3	2	5	4	4	3
002	.	2	1	2	1	2
003	5	3	4	4	.	5

❖ 변수 Q6의 값이 Q6의 평균값보다 크거나 같으면 변수 group에 'gtmean', 그렇지 않으면 변수 group에 'ltmean'을 저장시키는 SAS 프로그램을 작성하여라.

연습문제

- ❖ 다음 프로그램을 실행시키면 exam이 생성된다. 이 데이터셋에서 각 변수에 대한 결측값이 변수 x1~x5에 대해서는 9, 변수 a, b, c, d에 대해서는 99, 나머지 변수들에 대해서는 999로 입력되어 있다고 할 때, 이들 자료값을 결측값으로 처리하는 프로그램을 작성하여라. 단, 결측값은 SAS에서의 디폴트인 마침표(.)로 하여라.

GOTO 명령문의 사용

```
DATA exam;  
INPUT x1-x5 a b c d y1-y5 z1-z3;  
CARDS;  
1 0 1 0 1 2 2 2 2 1 2 3 4 5 3 3  
9 0 0 0 9 99 99 99 7 999 999 4 5 6 999 999 999  
;  
RUN;
```

- ❖ 다음 데이터를 이용하여 시작일자(startdate)부터 종료일자(enddate)까지 걸린 일 수를 출력하는 SAS 프로그램을 작성하여라.

Project_ID	StartDate	EndDate
398	17mar2007	02nov2007
942	22jan2008	11jul2008
167	15aug2009	15feb2010
250	04jan2011	11dec2011