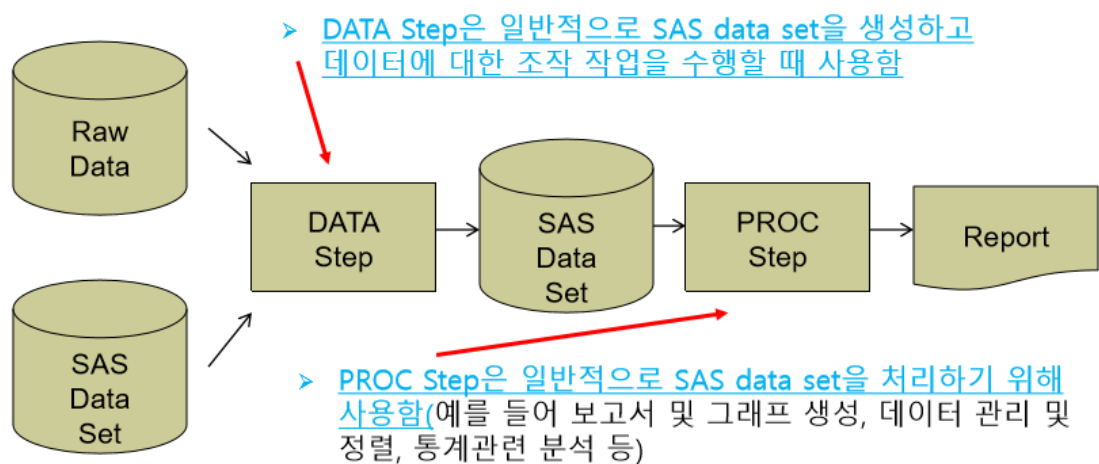


2-2강

SAS 프로그램

- SAS 프로그램은 사용자가 실행시키는 일련의 스텝 (DATA step 및 PROC step)으로 구성됨



DATA Step은 데이터 전처리를 의미한다.

PROC Step에서 원하는 통계계산을 하는 것이다.

SAS 문장

- SAS프로그램은 SAS문장으로 구성되며, 하나의 SAS문장은 SAS 키워드, SAS 이름 (name), 연산자(operator), 특수문자로 이루어지며, 데이터를 제외한 모든 문장은 세미콜론(;)으로 끝난다.

Aa SAS문장 구성요소	≡ 특징	≡ 예시
<u>SAS 키워드</u>	이름으로 사용될 수 없는 고정된 어휘	DATA, INPUT 등
<u>SAS 이름</u>	사용자가 임의로 작성이 가능	name, number, score등
<u>연산자</u>	사용자가 원하는 연산을 지정하는 기호	/, =, + 등
<u>특수문자</u>		\$ 등

자주 사용하는 SAS 중심어(keyword)

- DATA : 데이터를 읽어 들여 SAS 데이터를 만들거나 데이터를 수정하는 DATA 단계
- CARDS (또는 datalines) : 바로 다음 줄부터 자료가 뒤따름
- INPUT : 데이터의 입력순서 및 입력 데이터값이 숫자인지 문자인지를 명시함
- RUN : DATA 단계나 PROC 단계가 끝났음을 의미함
- PROC : SAS데이터 분석을 하거나 함수를 사용하여 그 결과를 확인하는 PROC 단계

SAS의 프로그램 구조

1. DATA Step

- DATA라는 키워드와 데이터이름으로 시작
- SAS Data Set의 생성 및 자료 값의 계산 및 변형 등과 관련된 작업을 처리

✓ 신규 생성

```
DATA tmp;  
INFILE 'E:\data\sample.txt'; /* 외부파일에서 데이터 가져옴 */  
INPUT gender $ age score;  
RUN;
```

✓ 데이터셋 활용

```
DATA tmp;  
SET sample; /* 기존 데이터셋 복사 */  
RUN;
```

✓ 프로시저 활용

```
PROC SORT DATA=sample OUT=tmp;  
BY gender age;  
RUN;
```

INFILE은 외부 데이터를 가져오는 키워드이다.

gender \$ → 문자,, age, score은 숫자.

RUN은 DATA랑 짝을 맞춰야 한다.

SET의 제1용법은 복사하는 것이다.

SORT는 정렬이다. → 오름차순이 기본이다.

BY는 gender로 먼저 정렬하고, 나중에 age로 정렬하라는 의미이다.

OUT은 tmp이다. OUT을 안쓰면 sample이 SORT되어서 바뀌어버린다.

2. PROC Step

- PROC이라는 키워드와 고유명칭이 부여된 기능을 갖는 Procedure의 이름으로 시작
- Procedure별로 고유기능을 가지며, 자신만의 명령문 문법을 갖는다.

✓ 프로시저 활용

```
PROC UNIVARIATE DATA=new NORMAL;  
OUTPUT OUT  
TestsForNormality = normtest;  
RUN;
```

✓ 솔루션 활용 : 데이터분석 , 시계열분석 등의 솔루션을 활용한 분석 수행

✓ 매크로 활용 : 반복적인 자료 처리 및 분석 처리

직접 PROC을 활용하는 방법이 있다.

→ UNIVARIATE NORMAL 주어진 자료가 정규분포인지 아닌지를 정확하게 판별 가능하다.

솔루션, 매크로를 사용하는 방법도 있다.

SAS 프로그램 예

```
DATA ex1_1;  
INPUT id $ name $ gender $ weight height;  
IF weight >= 70 THEN weight1="GE70";  
ELSE weight1="LT70";  
CARDS;  
101 김철수 M 74 170
```

```

102 이영희 F 68 166
103 안수지 F 55 155
104 박민호 M 72 167
;
RUN;

/* SAS 데이터 인쇄하기*/
PROC PRINT DATA=ex1_1;
RUN;

/* 통계량의 계산 */
PROC MEANS DATA=ex1_1;
VAR weight height;
RUN;

```

PROC PRINT에서 DATA를 안쓰면 바로 위에서 쓴 DATA를 가져온다.
그치만 습관적으로 써주는 것이 좋다.

```

PROC MEANS DATA=ex1_1;
VAR

```

에서 VAR에 계산할 변수들의 이름을 지정해주는 것이다.

프로그래밍 기본 개념

SAS 구문(Syntax) 규칙

- SAS 프로그램은 비교적 문법이 단순
- 주로 영문 또는 그 약어로 구성
- 규칙
 - SAS 문장은 일반적으로 키워드로 시작하고 항상 세미콜론(;)으로 끝냄
 - 프로그램 작성시 가장 기본적인 문법이며 가장 많은 오류가 발생하는 규칙.
 - 세미콜론으로 문장을 구분하게 되므로 SAS 문장이 길어지게 되면 여러 줄에 입력해도 되며 1줄에 여러 개의 문장을 복수로 작성해도 됨
 - 사용자가 직접 정의하는 이름들은 영문자(또는 “_”)로 시작해야 하며 숫자 이외의 다른 기호의 조합은 사용할 수 없음
 - 숫자로 연결된 변수명 : VAR1 - VAR4 (VAR1, VAR2 , VAR3 , VAR4 를 통칭)

- 영문으로 연결된 경우 AGE1 -- JOB6 (이 범위의 모든 이름을 통칭)
- 기본적으로 SAS 키워드는 사용자 문장으로 사용할 수 없음
- 인용부호는 홑(홀) 따옴표(' ')와 겹 따옴표(" ")가 사용
 - 인용부호 내에 따옴표가 입력될 수 있으므로 가급적 겹 따옴표(" ")를 사용하는 것이 좋음
- 별표로 (*)시작하고 세미콜론(;)으로 끝나는 문장은 주석 처리
 - 실행할 필요가 없는 문장은 문장의 맨 앞에 별표(*)를 붙임
 - 여러 문장에 대해 주석 처리할 때는 시작(/*)과 끝(*/)으로 표시

☞ 주석(Comments)

- 주석(Comments)은 SAS가 프로그램 처리시에 처리하지 않고 무시하게 되는 글자들을 말함.
- 프로그램 내의 어느 곳이든 사용 가능
- /* comment*/ 혹은 * comment ;
- 따옴표(' ' 또는 " ") 내의 문자값(character value)를 제외하고는 프로그램에서 대소문자를 구별하지 않음
- 확장편집기 내의 어떠한 칼럼에서 시작해도 관계 없으나, 프로그램을 쉽게 읽기 위해 들여쓰기를 사용하여 작성함

주석(Comment) 사례

1. * message ; format:

```
* This code finds the number in the BY group ;
```

2. * message ; format:

```
* ----- *
|This uses one comment statement      |
| to draw a box.                      |
* ----- *;
```

3. /* message */ format:

```
input @1 name $20. /* last name */
      @200 test 8. /* score test */
      @50 age 3.; /* customer age */
```

4. /* message */ format:

```
/* For example 1 use: x=abc;
for example 2 use: y=ghi; */
```

데이터 입력: 프로그램 내부에서 입력

Cards (Datalines) 명령문 이용

- 명령문 형식

```
CARDS;           [Data 입력 선언]
...              [Data 입력]
;                [Data 입력 종료]
```

- Example

```
DATA personal;
INPUT name $ dept $ salary age;
CARDS;
김재훈 stat 96 38
김영옥 math 89 42
;
RUN;
```

INFILE 명령문을 이용

- 명령문 형식

```
INFILE '[Data File 의 경로 및 이름 지정]'
```

- Example

```
DATA example;
INFILE 'c:\work\person.dat';
```

```
INPUT name $ dept $ salary age;  
RUN;
```