

7.

어디서나 사용하는 SAS 문장들

7장에서 소개하는 문장들은 SAS 프로그램에서 DATA 단계와 PROC 단계에 구애받지 않고 어디서나 사용 가능하다.

7.1 주석(comment) 문장

주석(comment) 문장은 SAS 프로그램 내 어디서도 사용 가능하며 프로그램의 목적이라든가 각 단계에서 수행하는 작업에 대한 **설명 등을 프로그램 사이에 덧붙여** 사용자가 프로그램의 흐름을 쉽게 파악할 수 있도록 하는 것으로, 프로그램의 실행과는 전혀 상관없는 **비실행 문장**이다.

- | |
|------------------------------|
| ① <code>* message ;</code> |
| ② <code>/* message */</code> |

message 는 사용자가 작성하는 설명문으로 길이 제한 없음

② 에서의 *message* 에는 무슨 문자라도 사용 가능하나,

① 에서의 *message* 에는 세미콜론(:) 사용 불가

(즉, * 과 ; 사이에 또 다른 ; 사용 불가)

(예) `* This step computes SSE ;`

(예) `*=====*`
`| Programmed by Nae K. Sung |`
`| 1/1/2005 |`
`*-----*`;

(예) 문장 내에도 포함되는 주석 문장은 ②

`PROC PRINT;`

`VAR wt ht /* wt=weight, ht=height */ ;`

`RUN;`

(예)

```
DATA class;  
    Set class1;  
RUN;
```

```
PROC PRINT;  
RUN; /*
```

```
DATA credit;  
    SET class;  
    IF score < 60 THEN credit='F';  
RUN;
```

```
PROC PRINT;  
RUN;    */
```

7.2 OPTIONS 문장

프로그램에서 시스템 선택사항을 변경 지정하는 목적으로 사용한다.

OPTIONS *option* ... ;

한 프로그램 내에 몇 번이라도 사용 가능

option 으로는 다음과 같은 것들

NONUMBER (출력창에 페이지 출력 안됨)

NODATE (출력창에 날짜/시간 등 출력 안됨)

LS (출력 시 Line Size 지정)

NOCENTER (출력 시 왼쪽 정렬)

FORMDLIM (페이지와 페이지 구분을 위한 옵션)

NONOTES (로그창에 error 및 warning을 제외한 notes 출력 안함)

<http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a000309878.htm>

<http://support.sas.com/documentation/cdl/en/lesysoptsref/68023/HTML/default/viewer.htm#p1tmgku1vq7pwqn1iqioeflxgec1.htm>

(예) OPTIONS NONUMBER NODATE LS=80 FORMDLIM="-";

/* FORMDLIM 은 한글 아닌 1개 문자만 지정 가능*/

(예) OPTIONS NUMBER DATE LS=132 FORMDLIM="";

※ 굵은 색 밑줄은 디폴트

CENTER | NOCENTER

실행 결과가 output 창으로 출력될 때 사용할 정렬 방식 지정.

CENTER 는 가운데 정렬, NOCENTER는 왼쪽 정렬이지만, PUT 문장을 사용한 출력은 이 선택사항의 영향을 받지 않는다.

DATE | NODATE

현재 날짜를 output 창의 모든 출력 페이지마다 상단에 인쇄할지 지정.

날짜를 인쇄하고 싶지 않으면 NODATE을 선택한다.

ERRORS=*n*

자료를 읽어들이는 때 지정된 입력 방식과 맞지 않는 관측이 존재하면 오류 메시지가 Log 창으로 출력되는데, 최대 몇 개 관측들에 대하여 오류

메시지를 출력할지 지정한다. **디폴트는 20개**이다. 예를 들어, 총 관측 수가 500개인 자료를 읽는데 오류가 있는 관측의 수가 50개라면 최초 20개 관측에 대해서는 오류 메시지가 출력되나 나머지 30개에 대해서는 오류 메시지 출력없이 그대로 작업이 수행된다.

$n = 0, 1, \dots$ 가능

`LINESIZE=width` 또는 `LS=width`

줄 크기 즉, 출력에서 한 줄에 포함되는 문자들의 최대 개수를 지정한다.
지정 가능한 *width* 범위는 64~256.

`MISSING='character'`

분실(결측) 숫자변수 값을 어떻게 출력할지 지정.

디폴트는 마침표(.)

분실값을 공백으로 나타내고 싶으면 `MISSING=''`

분실값을 *로 나타내고 싶으면 `MISSING='*'` (단, 한 개 문자만 사용 가능)

NOTES | NONOTES

프로그램 실행 시 SAS 시스템의 동작 상태 및 오류 메시지 등이 Log 창에 출력되는데, NONOTES를 지정하면 아주 축약된 형태의 오류 메시지만 출력된다. NONOTES를 지정하면 작업 소요 시간이 단축된다.

NUMBER | NONUMBER

Output 창으로 출력되는 SAS 절차들의 결과를 페이지 단위로 번호 붙여 출력할지 지정. 디폴트 NUMBER는 매 페이지 상단에 일련 번호를 인쇄한다.

`PAGENO=n`

Output 창으로 실행 결과를 출력할 때 출력 페이지의 시작 번호를 지정.

`PAGESIZE=n` 또는 `PS=n`

페이지 크기 즉, output 창으로 실행 결과를 출력할 때 한 페이지에 포함된 최대 줄의 개수를 지정. 지정 범위는 15~32767.

글꼴의 크기에 따라 다르지만 10급을 사용할 경우 66줄이 한 페이지를 이루므로 `PS=60` 과 같이 지정하는 것이 좋다. 위 아래 3 줄씩은 보통 사용하지 않는다.

[참고] PROC OPTIONS; RUN;

: 현재 시스템 선택사항 설정이 로그창에 출력

7.3 RUN 문장

DATA 단계나 PROC 단계가 끝이 났음을 SAS 시스템에 알려주는 역할을 한다.

RUN;

[참고] SAS 시스템은 항상 단계(step)를 단위로 프로그램을 실행/처리한다.

일괄 처리할 SAS 프로그램이라면 DATA 단계나 PROC 단계 사이에 RUN 문장을 항상 삽입하지 않아도 프로그램 실행에는 상관이 없으나, 단계의 끝에는 반드시 RUN 문장을 사용하는 것이 좋다. RUN 문장을 사용하지 않으면 SAS 시스템은 언제나 새로운 DATA 문장이나 PROC 문장을 만난 후에야 비로소 한 단계가 끝이 났다고 인식하고 그때까지 읽어들이는 단계를 실행한다.

(예) DATA first;

SET old;

PROC PRINT;

⇒ 처음 DATA 단계는 제대로 수행되나 출력은 없다. 왜냐하면, 그 다음의 PRINT 절차의 끝이 명시되지 않았기 때문에 PRINT 절차의 부속 문장이 더 입력될 것으로 인식을 하고 실행이 보류된다.

(예) 비실행문(TITLE 문 또는 FOOTNOTE 문 등) 사용 시 문제점

TITLE1 '첫 번째 타이틀'; /* TITLE 문장은 비실행문 */

PROC PRINT;

*RUN;

TITLE1 '새로운 타이틀';

PROC PRINT;

*RUN;

PROC PLOT;

PLOT x*y;

*RUN;

*QUIT;

⇒ 첫 줄 TITLE 문장 읽고, 두 번째와 세 번째 문장을 읽고, 단계가 끝났다는 RUN 문장이 없으므로 네 번째 줄에 있는 TITLE 문장을 읽고, 다섯 번째 줄에 있는 PROC 문장을 만나므로 앞서 읽은 네 문장을 실행한다. 따라서 첫 번째 REG 절차의 출력 표제로 첫 번째 TITLE 문장의 표제가 아닌, 두 번째 TITLE 문장의 표제가 나오게 되어 의도된 바와 같은 결과를 얻지 못한다.

7.4 TITLE 문장

Output창(출력창)으로 출력되는 매 페이지마다 표제를 붙일 때 사용하는 문장으로, 모두 10줄까지 사용할 수 있다.

TITLE n 'text' ;

n 은 표제가 인쇄될 줄 번호 즉, 출력하고자 하는 line (1~10 까지 가능)
한 번 적용하면 다음 TITLE 문 나오기 전까지 똑같은 표제 적용
표제 지정 없을 시 디폴트는 'The SAS System'(영문)/'SAS 시스템'(한글)
 $text$ 는 표제로 하고 싶은 문자열로, 사용 가능한 문자열의 범위는 시스템
옵션 LINESIZE 지정에 좌우되며, 디폴트는 최대 132 문자열

TITLE; 또는 **TITLE1;**

앞서 지정된 표제가 다음 각 출력 페이지에도 계속 인쇄되는데, 앞서
지정된 표제가 더 이상 인쇄되지 않도록 하는 **null title**
(앞서 지정된 모든 표제 해제)

(예) **TITLE 'Project - Robust ANOVA';**

⇒ TITLE 다음에 번호가 없으면 자동적으로 1 이 부여되며 SAS 시스템의
디폴트 표제 대신 이 표제가 출력됨

(예) **TITLE1 'Project - Core Institute of Statistics';**

TITLE3 'March 1, 2009';

⇒ TITLE2 가 없으므로 출력되는 3 줄의 표제 중 2 번째 줄은 빈 줄 즉, 한
줄 띄우는 것

(예) **TITLE1 'Project - Core Institute of Statistics';**

TITLE10 'March 1, 2009';

⇒ 두 표제 사이에 8 줄의 빈 줄이 생김

(예) **TITLE1 'Project - Core Institute of Statistics';**

TITLE3 'March 1, 2009';

.

. (SAS 문장들)

TITLE2;

⇒ TITLE2를 만나면 TITLE2부터 TITLE10까지 모든 과거 지정 해제됨

(즉, TITLE1 문장은 아직 유효함)

TITLE1;

⇒ 지금까지의 모든 표제 지정 해제됨

7.5 FOOTNOTE 문장

용법은 TITLE 문장과 동일하다.

```
FOOTNOTE $n$  'text' ;
```

n 은 각주가 인쇄될 줄 번호 즉, 출력하고자 하는 line (1~10 까지 가능)
한 번 적용하면 다음 FOOTNOTE 문 나오기 전까지 똑같은 각주 적용
 $text$ 는 각주로 하고 싶은 문자열로, 사용 가능한 문자열의 범위는 시스템
옵션 LINESIZE 지정에 좌우되며, 디폴트는 최대 132자

```
FOOTNOTE;
```

또는

```
FOOTNOTE1;
```

앞서 지정된 각주가 다음 각 출력 페이지에도 계속 인쇄되는데, 앞서
지정된 각주가 더 이상 인쇄되지 않도록 하는 **null footnote**
(앞서 지정된 모든 각주 해제)

(예) FOOTNOTE1 'Project – Core Institute of Statistics';
FOOTNOTE3 'March 1, 2009';
PROC PRINT;
RUN;

7.6 ENDSAS 문장

이 문장을 현재 편집 중인 SAS 프로그램의 끝에 삽입하고 실행시키면 프로그램의 실행이 끝남과 동시에 SAS 시스템이 자동적으로 종료된다.

ENDSAS;

7.7 LIBNAME 문장

폴더/경로를 지정할 때 사용한다. (영구 SAS 자료를 처리하기 위해 사용)

자료 보관 형태에 따라 2 종류의 SAS 자료

- 일시/임시 자료 (temporary data set)

: 프로그램이 실행되면 SAS 시스템의 디폴트 폴더(library; 라이브러리)인 **work**에 만들어졌다가 SAS 시스템 종료와 함께 삭제

- 영구 자료 (permanent data set)

: 지정된 디스크 상의 폴더(library; 라이브러리)에, 지정된 파일 이름으로 영구 저장되며 나중에 몇 번이고 호출하여 재사용 가능

[참고] SAS 데이터 라이브러리는,

SAS 자료가 저장되는 디스크 상의 공간으로 이는 곧 **폴더**이다.
특정 폴더를 받는 라이브러리 이름을 *libref*(library reference)로 표기하며, *libref* 는 라이브러리로 배정된 폴더를 SAS 시스템에서 지칭하는 대명사다.

`LIBNAME libref 'folder' ;`

libref 이름은 최대 8자의 한 단어

용도가 유보된 몇 어휘는 사용 불가 (8장 참고)

folder 는 SAS에서 사용하기 전에 존재하는 폴더(경로)이어야 함

[참고] SAS 자료 이름은 마침표(.)로 구분된 두 단어를 사용하여
마침표 앞 단어는 *libref*, 마침표 뒤 단어는 파일명
(e.g.) work.one comstat.two ...

(예) `LIBNAME child 'a:\family\children'; /* child 라는 라이브러리 생성 */`

`LIBNAME preadult 'c:\teenage'; /* preadult 라는 라이브러리 생성 */`

`DATA preadult.teens;`

`SET child.eldest;`

`IF age < 13 THEN DELETE;`

`RUN;`

⇒ A: 드라이브의 경로 \family\children 에 저장되어 있는 영구 SAS 자료 파일 eldest를 읽어들이 age 라는 변수가 13 이상인 관측들만 추려서 경로 C:\teenage 에 teens 라는 이름의 영구 SAS 자료를 생성하는 프로그램

[참고] SAS 시스템에서는 SAS 자료 이름의 확장자는 표기하지 않아도 됨

(예) DATA test; /* ⇔ DATA work.test */

INPUT x y;

DATALINES;

2 3

3 4

RUN;

(예) LIBNAME mylib “.”; /* 마침표(.)는 현재 폴더를 의미 (현재 폴더를 mylib
이라는 라이브러리로 정의) */

/* 현재 폴더는 SAS 창 우측 하단에 표시되어 있음 */

/* 현재 폴더를 변경하려면 현재 폴더 이름을

더블클릭하여 새로운 폴더 지정 */

7.8 FILENAME 문장

폴더를 포함한 완전한 파일 이름을 받는 대명사를 지정할 때 사용한다. (외부 아스키 파일(txt 파일) 등을 읽어들이는 때 사용)

FILENAME *fileref* 'filename' ;

fileref(file reference 또는 file shortcut)는 외부 파일 이름을 받는 대명사. 외부 아스키파일 읽어들이는 경우 INFILE 문장과 함께 사용.

fileref 은 library 이름 만드는 규칙과 동일

(영문자, 숫자, _ 조합 최대 8자 (시작은 영문자 또는 _))

(예) FILENAME child 'a:\Wfamily\Wchildren\Weldest.txt';

/* 경로 포함 파일명(확장자명 포함)까지 지정 */

DATA teens;

INFILE child;

INPUT sex age ...;

RUN;

⇒ A:\Wfamily\Wchildren 에 저장되어 있는 아스키 파일 eldest.txt를 읽어들이어 teens 라는 데이터 파일명으로 저장

⇔ FILENAME child 'a:\Wfamily\Wchildren'; /* 경로만 지정 */

/* LIBNAME 문장으로 대체 불가능 */

DATA teens;

INFILE child(eldest.txt); /* 경로명 (파일명 (확장자명 포함)) */

INPUT sex age ...;

RUN;

⇔ DATA teens;

INFILE 'a:\Wfamily\Wchildren\Weldest.txt';

INPUT sex age ...;

RUN;

(예) FILENAME child 'a:\Wfamily\Wchildren\Weldest.txt';

LIBNAME myfolder 'c:\Wteenage';

DATA myfolder.teens;

INFILE child;

INPUT sex age ... ;

RUN;

7.9 %INCLUDE 문장

외부 파일의 내용을 그대로 SAS 프로그램에 삽입하는 목적으로 사용한다.

```
%INCLUDE source ;
```

(예) DATA scatter;

```
INPUT x y @@;
```

```
DATALINES;
```

```
-2 -8 -1 -1 0 0 1 1 2 8
```

```
RUN;
```

```
PROC PLOT;
```

```
PLOT y*x;
```

```
QUIT;
```

⇔ c:\work 에 다음과 같은 외부 SAS 프로그램 파일 **plot.sas** 존재한다면,

```
PROC PLOT;
```

```
PLOT y*x;
```

```
QUIT;
```

```
DATA scatter;
```

```
INPUT x y @@;
```

```
DATALINES;
```

```
-2 -8 -1 -1 0 0 1 1 2 8
```

```
RUN;
```

/* plot.txt 도 가능 */

```
%INCLUDE "c:\work\plot.sas";
```

/* 확장자명까지 정확히 기재 */

⇔ **FILENAME** myfolder "c:\work";

/* LIBNAME 문장으로 대체 불가능 */

```
DATA scatter;
```

```
INPUT x y @@;
```

```
DATALINES;
```

```
-2 -8 -1 -1 0 0 1 1 2 8
```

```
RUN;
```

```
%INCLUDE myfolder(plot);
```

/* 확장자명 생략하면 plot.sas 로 인식 */

⇔ **FILENAME** myfolder "c:\work\plot.sas";

/* 확장자명도 정확히 기재 */

```
DATA scatter;
```

```
INPUT x y @@;
```

```
DATALINES;
```

-2 -8 -1 -1 0 0 1 1 2 8

RUN;

%INCLUDE myfolder;

(예) 잘못 사용된 예제

“c:\work” 에 data.txt 이름으로 다음과 같이 자료가 저장되어 있다면,

-2	-8
-1	-1
0	0
1	1
2	8

DATA scatter;

INPUT x y;

DATALINES;

%INCLUDE “c:\work\data.txt”; /* DATALINES 문장 다음에 나타나는
문자열은 무조건 자료값으로만 간주하므로 잘못된 표현 */

RUN;

(예) c:\work 에 data.txt 이름으로 자료가 저장되어 있다면,

원시 자료에 DATALINES 문장을 다음과 같이 데이터 앞에 첨부하여야 한다.

(이 파일을 carddata.sas 라고 하자.)

DATALINES;

-2	-8
-1	-1
0	0
1	1
2	8

DATA scatter;

INPUT x y;

%INCLUDE “c:\work\carddata.sas”;

RUN;

7.10 출력 전달 시스템 (ODS: Output Delivery System)

SAS 출력 형태를 지정할 수 있는 문장으로, 출력창으로 나오는 출력물을 html, pdf 등의 출력물 형태로 변경할 수 있는 문장이므로 사실상 SAS 프로그램 어느 위치에서나 사용 가능하다.

다양한 포맷의 출력을 제공하기 위한 도구로 다음과 같은 포맷의 문서 출력을 지원한다.

- LISTING - (Output 창으로 나가는 텍스트 출력) 디폴트 출력, 고정폭 글꼴 사용
- PRINTER - (인쇄용 출력) Postscript 출력, 변동폭 글꼴 사용
- PDF - (Adobe 포맷) PDF(Postable Document Format) 출력
- HTML - (Results Viewer 창으로 나가는 출력) 웹 페이지 게시용 HTML 출력
- RTF - (텍스트 포맷에 색상 등 삽입한 포맷) 문서 편집기 삽입용

ODS를 이용하려면 항상 출력하고 싶은 절차 전후에서 ODS를 열고 닫고, 열 때는 출력 파일이 저장될 경로를 지정한다.

```
(예) DATA;  
      INPUT x $;  
      DATALINES;  
           Sung  
           Moon  
  
      RUN;  
      ODS PDF FILE="c:\W전산통계\Wtest.pdf";  
      ODS HTML FILE="c:\W전산통계\Wtest.htm";  
      PROC PRINT;  
      RUN;  
      ODS PDF CLOSE;  
      ODS HTML CLOSE;
```