

통계계산소프트웨어

SAS DATA STEP

2018. 9.

SAS프로그래밍의 구조

1. DATA STEP

- SAS 데이터셋에 데이터 입력하기
- 새로운 변수값 계산
- 데이터의 오류 확인 및 수정
- 기존 데이터셋의 서브셋, 병합, 업데이트 등으로 새로운 SAS 데이터셋 만들기

2. PROC STEP

- 리포트 출력
- 기술 통계 생성
- 테이블 리포트 작성
- 도표 및 차트 생성

Data Step

1. 시작

DATA <SAS data set names> <options>

2. data step에는 일반적으로 다음 중 하나의 문장이 있다

INPUT, SET, MERGE, UPDATE

3. 기본적인 data step 문장

```
DATA one;  
INPUT id gender $ weight;  
CARDS;  
1234 M 49.5  
1537 F 40.0  
1745 M 70.2  
1955 F 42.3  
;RUN;
```

→ data step의 시작
→ 자료를 읽음
→ 자료의 시작

→ 자료의 끝

VIEWTABLE: Work.One			
	id	gender	weight
1	1234	M	49.5
2	1537	F	40
3	1745	M	70.2
4	1955	F	42.3

자료의 입력 (자유 입력/List Input)

Input문과 Cards(또는 datalines)문 사용

Input – \$는 그 관측값이 문자일 때

Cards – 빈칸으로 관측값을 구분한다

Input 자료의 개수만큼만 입력 받는다.

```
DATA one;  
INPUT id gender $ weight;  
CARDS;  
1234 M 49.5  
1537 F 40.0  
1745 M 70.2  
1955 F 42.3  
;RUN;
```



```
DATA data이름;  
INPUT 자료의 이름;
```

★ 자료의 출력

```
PROC PRINT DATA=one;  
RUN;
```

RUN 문 : 각 DATA 단계 또는 PROC 절차의 입력이
완료되었음을 SAS 시스템에 알리는 역할

자료의 입력 예

```
data one;  
  input x y z;  
  cards;  
1 2 3  
4 5 6  
;  
run;
```

OBS	x	y	z
1	1	2	3
2	4	5	6

```
data two;  
  input x y z;  
  cards;  
1 2 3 4 5 6  
7 8 9 8 4 2  
;  
run;
```

OBS	x	y	z
1	1	2	3
2	7	8	9

```
data three;  
  input x y z;  
  cards;  
1 2  
3 4 5  
6 7 8  
;  
run;
```

OBS	x	y	z
1	1	2	3
2	6	7	8

```
data four;  
  input x y z;  
  cards;  
1 2 3 4  
5 6  
7 8  
9 0 1  
;  
run;
```

OBS	x	y	z
1	1	2	3
2	5	6	7
3	9	0	1

자료의 입력 (열 지정 입력/Column Input)

Input문과 Cards(또는 datalines)문 사용

Input - \$는 그 관측값이 문자일 때

“Column input은 자료가 빈칸으로 구분되어 있지 않거나
몇 개의 자료를 건너 띄고 필요한 자료만을 읽을 때 쓰인다”

DATA one;

INPUT id 1-4 gender \$ 5-6 weight 7-11;

CARDS;

1234 M 49.5

1537 F 40.0

1745 M 70.2

1955 F 42.3

;RUN;



data 데이터 이름;



Input 자료의 이름
숫자 -> 데이터 입력 공간

1	2	3	4	5	6	7	8	9	0	1	2
1	2	3	4		M		4	9	.	5	
1	5	3	7		F		4	0			
1	7	4	5		M		7	0	.	2	
1	9	5	5		F		4	2	.	3	

자료의 입력 (포맷 입력/Formatted Input)

Format : a.b ; a는 전체 자릿수, b는 소수점 밑자리수

a. = a.0

주어진 자리만큼 순서대로 읽어간다. 자료에서 소수점이 주어지면 Format에서의 소수점 밑자리수는 무의미해진다.

```
data three;
  input id 4. gender$ 3. weight 4.1;
  cards;
1234 M 49.5
1537 F 40.0
1745 M 70.2
1955 F 42.3
;
run;
```

1	2	3	4	5	6	7	8	9	0	1	2
1	2	3	4		M		4	9	.	5	
1	5	3	7		F		4	0			
1	7	4	5		M		7	0	.	2	
1	9	5	5		F		4	2	.	3	

	id	gender	weight
1	1234	M	49.5
2	1537	F	40
3	1745	M	70.2
4	1955	F	42.3

```
data three;
  input x 4.2 a$ 4.;
  cards;
1234A
12.3 A
77 A B
199 A B
199A B
;
run;
```

1	2	3	4	5	6	7	8
1	2	3	4	A			
1	2	.	3		A		
	7	7		A		B	
1	9	9			A		B
	1	9	9	A			B

VIEWTABLE: Work.Three		
	x	a
1	12.34	A
2	12.3	A
3	0.77	A B
4	1.99	A B
5	1.99	A B

자료 읽기 : @@

Input 문에서 cards 의 값을 연속적으로 읽을 수 있게 해주는 옵션

Input 변수만큼 읽으면 더 이상 읽지 않음, @@를 이용 한 줄을 다 읽는다

```
data three;  
  input x y;  
  cards;  
  1 2 3 4 5 6  
  7 8 9 7 2 6  
;  
run;
```

VIEWTABLE: Work.Three		
	x	y
1	1	2
2	7	8

```
data four;  
  input x y @@;  
  cards;  
  1 2 3 4 5 6  
  7 8 9 7 2 6  
;  
run;
```

VIEWTABLE: Work.Four		
	x	y
1	1	2
2	3	4
3	5	6
4	7	8
5	9	7
6	2	6

하나의 개체가 여러 줄로 입력된 자료

- ✓ 하나의 관찰개체의 자료가 여러 줄에 걸쳐서 입력될 때 사용
- ✓ / : 포인터의 위치를 다음 줄의 첫 열로
- ✓ #n : 줄 포인터 - 포인터의 위치를 n번째 줄의 첫 열로
- ✓ @n : 열 포인터 - n번째 열로 자료의 입력시점을 이동

```
DATA club;  
  INPUT  indo name $ 6-19  
        / team $6.  
        #3 strtwght endwght;
```

```
CARDS;  
1023 David Show  
Red  
189 165  
1049 Amelia Serrano  
Yellow  
189 165  
;  
RUN;
```

VIEWTABLE: Work.Club					
	indo	name	team	strtwght	endwght
1	1023	David Show	Red	189	165
2	1049	Amelia Serrano	Yellow	189	165

자료의 입력_Infile 문 : 외부 파일로 부터 데이터셋의 생성

자료가 외부 파일에 저장 되어있을 경우 Cards문을 사용하여 불러 오지 않고 직접 불러 들일 때 사용

Input – 자료들의 이름을 지정

```
data test;  
  infile 'C:\Users\JHM\Desktop\sas교육용_1.txt';  
  input x y z;  
run;
```

data 변수이름 지정

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

데이터 읽기

■ 기존 SAS Data Set 읽기(편집기 사용)

✓ DATA 문장

- Data Step 시작 문장
- DATA 키워드 옆에 생성될 SAS Data Set 이름을 적음

✓ SET 문장

- 기존 SAS Data Set 을 읽을 때 사용하는 문장
- SAS Data Set 외의 Raw data 파일이나 기타 데이터 파일을 읽어올 수 없음
- 기본으로 입력 SAS Data Set의 모든 변수와 모든 관측치를 읽어 옴

데이터 읽기

■ 기존 SAS Data Set 읽기(편집기 사용)

```
LIBNAME 라이브러리이름 '경로';
```

```
DATA 출력-SAS-data-set;  
  SET 입력-SAS-data-set;  
  <기타 SAS 문장들>
```

```
RUN;
```

```
LIBNAME korea 'c:\wsas';
```

```
Data work.subset1;  
  SET korea.sales;
```

```
RUN;
```

[korea 라이브러리에 있는
sales 데이터를 불러들여
work라이브러리에 있는
subset1이라는 데이터명으로
저장]

데이터 읽기

■ 라이브러리 호출

```
LIBNAME korea 'c:\sas';
```

```
LIBNAME new 'c:\temp';
```

```
Data new.subset1;
```

```
    SET korea.sales;
```

```
    itemmean=(item1+item2+item3)/3;
```

```
RUN;
```

데이터 읽기 : 외부 파일로부터

DATA company;

INFILE 'E:\data\기업이미지.txt';

INPUT id 1-2 age 3 sex \$ 4 item1 5 item2 6 item3 7;

LABEL id='고객번호' age='나이' sex='성별'

item1='좋은 제품을 만들기 위해 노력한다'

item2='소비자를 중요하게 여긴다'

item3='신뢰할만한 기업이다';

RUN;

PROC PRINT DATA=company LABEL; RUN;

```
1 1M111
2 2M333
3 4F331
4 4M332
5 4M111
6 5F299
7 3M111
8 5F111
9 5M113
10 2F232
```

OBS	고객번호	나이	성별	좋은 제품을 만들기 위해 노력한다	소비자를 중요하게 여긴다	신뢰할만한 기업이다
1	1	1	M	1	1	1
2	2	2	M	3	3	3
3	3	4	F	3	3	1
4	4	4	M	3	3	2
5	5	4	M	1	1	1
6	6	5	F	2	9	9
7	7	3	M	1	1	1
8	8	5	F	1	1	1
9	9	5	M	1	1	3
10	10	2	F	2	3	2

데이터 읽기

SAS - [VIEWTABLE: Mysas]

파일(F) 편집(E) 보기(V) 도구(T) 데이터(D) 솔루션(S) 창(W) 도움말(H)

탐색기

'Mysas'의 내용

- Club
- Combine
- Company
- Concat

	고객번호	나이	성별	좋은 제품을 만들기 위해 노력한다	소비자를 중요하게 여긴다	신뢰할만한 기업이다
1	1	1	M	1	1	1
2	2	2	M	3	3	3
3	3	4	F	3	3	1
4	4	4	M	3	3	2
5	5	4	M	1	1	1
6	6	5	F	2	.	.
7	7	3	M	1	1	1
8	8	5	F	1	1	1
9	9	5	M	1	1	3
10	10	2	F	2	3	2

데이터 읽기

- 기존 SAS Data Set 읽기(탐색기 화면에서 읽기)
 - ✓ SAS 탐색기에서 읽고자 하는 SAS 파일을 선택, 2번 클릭하여 읽기
 - ✓ VIEWTABLE 이 열리며 데이터 읽음

The screenshot shows the SAS Explorer window. On the left, the '탐색기' (Explorer) pane displays the contents of 'E:\WSAS', listing several SAS files including kreish01.sa..., kreish02.sa..., kreish03.sa..., kreishw.sa..., kreisjob.sa..., kreisp01.sa..., kreisp02.sa..., kreisp03.sa..., kreispw.sa..., and kreissd.sas... The '탐색기' label is circled in red. On the right, the 'VIEWTABLE: _EXPO_kreissd' window displays a data table with the following structure:

	(가구) 가구고유번호	(가구) w1 가구번호	(가구) w2 가구번호	(가구) w3 가구번호	(개인) 개인고유번호	(가구) 원표본여부	(개인) 성별
1	1	1	1	1	10001011	1	
2	2	2	2	.	10002011	1	
3	2	2	2	.	10002012	1	
4	2	2	2	.	10002231	1	
5	3	3	.	.	10003011	1	
6	3	3	.	.	10003012	1	
7	4	4	.	.	10004011	1	
8	4	4	.	.	10004012	1	
9	5	5	5	5	10005011	1	
10	6	6	.	.	10006011	1	
11	6	6	.	.	10006012	1	
12	6	6	.	.	10006231	1	
13	7	7	7	7	10007011	1	

데이터 읽기

■ RAW Data File 읽기

✓ DATA 문장

- Data Step 시작 문장
- DATA 키워드 옆에 생성될 SAS Data Set 이름을 적음

✓ INFILE 문장

- 읽어올 외부 파일을 지정하는 문장
- INFILE 키워드 옆의 읽어 올 외부 파일의 경로 및 파일명 따옴표 안에 적음
예) infile 'c:\sas\sales.csv';
- **firstobs** : 자료를 불러들이기 시작하는 obs를 지정
- **expandtabs** : 자료의 사이가 tab으로 떨어져 있는 경우 사용

```
DATA output-SAS-data-set;  
    INFILE 'raw-data-file-name';  
    firstobs=2 expandtabs;  
    INPUT specification;  
RUN;
```

데이터 읽기

■ RAW Data File 읽기

✓ INPUT 문장

- Raw Data File을 어떻게 읽어올 것인가를 지정
- Raw Data File의 데이터 값을 어떻게 읽어서, 어떤 변수에 저장할 것인지를 지정함
- INPUT 문장 작성 방법에 따라 Column input, Formatted input, List input 등의 방법이 있음

Input 방식	방식 결정 요소		구문
	파일형식	비표준 데이터 처리	
Column	고정너비	처리 불가능	Input 변수명 <\$>시작위치-끝위치 ... ;
Formatted		처리 가능	Input <@start> 변수명 입력 형식 ... ;
List	구분자로 구분	처리 불가능	Input 변수명 <\$> ... ;
		처리 가능	Input 변수명 입력형식 ... ;

입력형식

구분	INFORMAT	내용	데이터값	입력포맷	저장
숫자	w.	w 자릿수의 정수로 표현	123	5.	123
	w.d	정수부분 w + 소수부분 d	123	5.1	12.3
	COMMAw.d	콤마와 \$포함, ()는 음수	(\$1,100)	COMMA10.	-1100
	PERCENTw.d	%부호 포함, ()는 음수	(20%)	PERCENT5.	-0.20
문자	\$w.	문자이전 공백 삭제	__Min Ho	\$8.	Min Ho
	\$CHARw.	문자이전 공백 포함	__Min Ho	\$CHAR8.	__Min Ho
날짜	MMDDYYw.	MM-DD-YY의 형태	01-01-1961	MMDDYY10.	366
	TIMEw.	HH:MM:SS의 형태	10:30	TIME5.	37800
	DATEw.	DD-MON-YY	01JAN1961	DATE9.	366

- ✓ YYMMDD8. : 67-08-13 YYMMDD10. : 1967-08-13
- ✓ MON : JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
- ✓ SAS의 날짜/시간 기준 : 1960년 1월 1일(=0) 00:00:01(=1)

원시데이터 형태

고정 (Fixed-format)

자유(free-format)

표준 데이터 유형

문자, 숫자(.포함)

Raw Data File Exercise

1	---	+	---	10	---	+	---	20
2810				61				MOD F
2804				38				HIGH F
2807				42				LOW M
2816				26				HIGH M
2833				32				MOD F
2823				29				HIGH M

1	---	+	---	10	---	+	---	20
BARNES				NORTH				360.98
FARLSON				WEST				243.94
LAWRENCE				NORTH				195.04
NELSON				EAST				169.30
STEWART				SOUTH				238.45
TAYLOR				WEST				318.87

Raw Data File Staff

1	---	+	---	10	---	+	---	20	---	+	---
EVANS				DONNY				112			29,996.63
HELMS				LISA				105			18,567.23
HIGGINS				JOHN				111			25,309.00
LARSON				AMY				113			32,696.78
MOORE				MARY				112			28,945.89

비표준 데이터 유형

1	1	Male	1,1,16%
2	2	Man	3,1,39\$
3	4	Female	3,3,19%
4	4	Man	3,3,24%
5	4	M	1,1,101%
6	5	Female	2,.,.
7	3	MR	1,1,1105%
8	5	Female	1,1,130%
9	5	Man	1,1,32%
10	2	Female	2,3,26%

고정 포맷 & 표준데이터 유형 (COLUMN INPUT)

데이터 유형	011M111 022M333 034F331 044M332 054M111 065F2.. 073M111 085F111 095M113 102F232	01,1,M,1,1,1 02,2,M,3,3,3 03,4,F,3,3,1 04,4,M,3,3,2 05,4,M,1,1,1 06,5,F,2,... 07,3,M,1,1,1 08,5,F,1,1,1 09,5,M,1,1,3 10,2,F,2,3,2	01 1 Male 1 1 1 02 2 Man 3 3 3 03 4 Female 3 3 1 04 4 Man 3 3 2 05 4 M 1 1 1 06 5 Female 2 . . 07 3 MR 1 1 1 08 5 Famme 1 1 1 09 5 Man 1 1 3 10 2 Female 2 3 2
특징	각 변수의 값을 읽는 위치가 모든 레코드에서 동일함		
문법	입력방법 : 변수명 변수유형 시작위치-끝위치 ex) age 1-2 , gen \$ 6-18, gen \$ 3-3 = gen \$ 3		

비표준 데이터 유형 (Formatted INPUT)

데이터 유형	고정 (Fixed-format)	자유(free-format)
	<pre> 1---+-----10---+-----20---+----- EVANS DONNY 112 29,996.63 HELMS LISA 105 18,567.23 HIGGINS JOHN 111 25,309.00 LARSON AMY 113 32,696.78 MOORE MARY 112 28,945.89 </pre>	<pre> 1 1 Male 1,1,16% 2 2 Man 3,1,39\$ 3 4 Female 3,3,19% 4 4 Man 3,3,24% 5 4 M 1,1,101% </pre>
	<pre> Martin, Virginia 09aug80 34800 Singleton, MaryAnn 24apr85 27900 Leighton, Morice 16dec83 32600 Freeler, Carl 15feb88 29900 Cage, Merce 19oct82 39800 </pre>	<pre> Martin Virginia 09aug80 34,800 Singleton MaryAnn 24apr85 27,900 Leighton Morice 16dec83 32,600 Freeler Carl 15feb88 29,900 Cage Merce 19oct82 39,800 </pre>
	<p>각 레코드의 변수별 데이터값 시작위치가 동일</p> <p>사전에 정의된 유형</p> <ul style="list-style-type: none"> - 00,000,000 : 천단위 숫자 구분 - ddMONyy : 날짜 표시 	<p>변수LIST 순서에 따라 구분자로 분리됨</p> <p>사전에 정의된 유형</p> <ul style="list-style-type: none"> - 00,000,000 : 천단위 숫자 구분 - ddMONyy : 날짜 표시
문법	<p>@입력시작위치 변수명 입력포맷</p> <p>ex) @1 id 2. , @6 gen \$12.</p>	<p>변수명 입력끝포인터 입력포맷</p> <p>ex) id & 2. , gen : \$12.</p>

포맷수정자 : & 와 :

INPUT 변수명 & (\$자릿수.);

- &(ampersand)는 2개 이상의 공백 다음에 오는 값 전까지 인식하는 것으로 &다음에 자릿수를 지정하면 공백까지 값을 읽은 후에 자릿수만큼 유효 자리로 인정한다.
- 관찰값 보다 자릿수가 작은 경우에는 해당 자릿수만큼 관찰값으로 읽는다.

INPUT 변수명 : (\$자릿수.);

- :(colon)은 해당 변수를 쓰고 그 다음에 :을 한 후에 자릿수를 지정하면 공백까지 관찰값을 읽은 후에 자릿수만큼 유효자리로 인정한다.
즉, 관찰값의 길이를 지정하는데 사용한다.
- 중간에 공백을 포함하지 않은 비표준데이터 값과 길이 8이상의 문자값을 읽는다

포맷수정자 : &

DATA STL;

INPUT SEASON \$ WIN_LOSE \$ WINNING_RATE \$ DISTRICT \$ PO \$ BEST_bWAR &\$16.;

CARDS;

2011 90-72 0.556 2위 우승 푸홀스 (5.3)

2012 88-74 0.543 2위 NLCS 몰리나 (6.9)

2011 97-65 0.599 1위 WS 웨인라이트 (6.4)

2014 90-72 0.556 1위 NLCS 웨인라이트 (6.4)

2015 100-62 0.617 1위 NLDS 헤이워드 (6.5)

;

RUN;

PROC PRINT; RUN;

OBS	SEASON	WIN_LOSE	WINNING_RATE	DISTRICT	PO	BEST_bWAR
1	2011	90-72	0.556	2위	우승	푸홀스 (5.3)
2	2012	88-74	0.543	2위	NLCS	몰리나 (6.9)
3	2011	97-65	0.599	1위	WS	웨인라이트 (6.4)
4	2014	90-72	0.556	1위	NLCS	웨인라이트 (6.4)
5	2015	100-62	0.617	1위	NLDS	헤이워드 (6.5)

포맷수정자 : & 를 이용하여 관찰값 읽기

```
DATA club2;  
  INPUT  indo name & $18. team $ stwgt endwgt;  
  CARDS;  
1023 David Shaw red 189 165  
1049 Amelia Serrano yellow 145 124  
;RUN;  
PROC PRINT DATA=club2;  
RUN;
```

공백 2칸

SAS 시스템

OBS	indo	name	team	stwgt	endwgt
1	1023	David Shaw	red	189	165
2	1049	Amelia Serrano	yellow	145	124

공백이 한칸일 경우

SAS 시스템

OBS	indo	name	team	stwgt	endwgt
1	1023	David Shaw red 189	1049	.	.

포맷수정자 : :

DATA topten2;

INPUT rank city \$&12. pop86 : comma.;

CARDS;

1 NEW YORK 7,262,700

2 LOS ANGELES 3,259,340

3 CHICAGO 3,009,530

4 HOUSTON 1,728,910

5 PHILADELPHIA 1,642,900

6 DETROIT 1,086,220

7 SAN DIEGO 1,015,190

8 DALLAS 1,003,520

9 SAN ANTONIO 914,350

10 PHOENIX 894,070

;

RUN;

PROC PRINT; **RUN**;

: 수정자를 사용하면 comma9.라고 써주지 않아도 된다.

자유 포맷 & 표준데이터 유형 (LIST INPUT)

데이터 유형

```
01 1 Male    1 1 1
02 2 Man     3 3 3
03 4 Female  3 3 1
04 4 Man     3 3 2
05 4 M       1 1 1
06 5 Female  2 . .
07 3 MR      1 1 1
08 5 Famme   1 1 1
09 5 Man     1 1 3
10 2 Female  2 3 2
```

```
1 1 Male 1 1 1
2 2 Man 3 3 3
3 4 Female 3 3 1
4 4 Man 3 3 2
5 4 M 1 1 1
6 5 Female 2 . .
7 3 MR 1 1 1
8 5 Famme 1 1 1
9 5 Man 1 1 3
10 2 Female 2 3 2
```

```
1,1,M,1,1,1
2,2,M,3,3,3
3,4,F,3,3,1
4,4,M,3,3,2
5,4,M,1,1,1
6,5,F,2,...
7,3,M,1,1,1
8,5,F,1,1,1
9,5,M,1,1,3
10,2,F,2,3,2
```

```
1,1,Male,1,1,1
2,2,Man,3,3,3
3,4,Female,3,3,1
4,4,Man,3,3,2
5,4,M,1,1,1
6,5,Female,2,...
7,3,MR,1,1,1
8,5,Famme,1,1,1
9,5,Man,1,1,3
10,2,Female,2,3,2
```

특징

구분자(공백)로 분리됨

구분자(COMMA)로 분리됨

기타 특징

- 결측값은 . 으로 표시되어 있다.
- 문자열은 공백을 포함하지 않고, 8자 이하이다.

문법

- 입력방법은 변수명 변수유형 ex) age gen \$
- 구분자 지정은 ex) INFILE fileref DLM=' , ' ;
(SAS SYSTEM Default Delimiter : 공백)
- 8자를 초과하는 문자데이터가 있는 변수에 대해서는 LENGTH 문장으로 미리 선언/지정 : LENGTH name \$ 10 ;

데이터 읽기_실습

■ Column Input

- ✓ 열 번호 지정
- ✓ 자료값이 고정된 열을 가지고 있어야 함

DATA scores;

INPUT name \$ 1-18 score1 25-27 score2 30-32
score3 35-37;

CARDS;

Joseph 11 32 76

Mitchel 13 29 82

Sue Ellen 14 27 74

;RUN;

PROC PRINT;

RUN;



OBS	name	score1	score2	score3
1	Joseph	11	32	76
2	Mitchel	13	29	82
3	Sue Ellen	14	27	74

데이터 읽기_실습

- Formatted Input → 파일 : offers.txt

```
DATA discounts;  
  infile 'd:\data\offers.txt';  
  INPUT @1 Cust_type 4.  
        @5 Offe_dt mmddyy8.  
        @14 Item_gp $8.  
        @22 Discount percent3.;  
RUN;  
PROC PRINT data=discounts;  
RUN;
```

SAS 시스템

OBS	Cust_type	Offe_dt	Item_gp	Discount
1	1014	17502	Outdoors	0.15
2	2020	17446	Golf	0.07
3	1030	17431	Shoes	0.10
4	1030	17431	Clothes	0.10
5	2020	17355	Clothes	0.15

Description	Column
Customer Type	1-4
Offer Date(월일년순)	5-12
Item Group	14-21
Discount	22-24



데이터 읽기_실습

■ List Input

- ✓ 데이터가 자유 포맷, 즉 하나 이상의 공백문자로 구분되었을 때 사용

```
DATA scores;  
  LENGTH name $ 12;  
  INPUT name $ score1 score2;  
CARDS;  
Riley 1132 1187  
Henderson 1015 1102  
;RUN;  
PROC PRINT data=scores;  
RUN;
```



SAS 시스템			
OBS	name	score1	score2
1	Riley	1132	1187
2	Henderson	1015	1102

데이터 읽기

■ 구분자

- ✓ 기본 구분자는 공백임
- ✓ 즉, 구분자를 기술하지 않을 경우 공백이 구분자로 인식됨
- ✓ INFILE 문장의 dlm=옵션으로 구분자를 정의할 수 있음

```
INFILE 'raw-data-file-name' dlm='구분자';
```

```
Data subset3;  
Infile 'c:\wsas\sales.csv' dlm=', ';
```

데이터 읽기_실습

- List Input : 표준데이터 처리 → 파일 : sales.txt

```
DATA subset3;  
infile 'd:\data\sales.txt' dlm=';';  
INPUT Employee_ID  
       First_Name $ Last_name $ Gender $  
       Salary      Job_Title $ Country $  
;RUN;  
PROC PRINT data=subset3;  
RUN;
```



문자형 변수의 길이가 8byte

SAS 시스템

OBS	Employee_ID	First_Name	Last_name	Gender	Salary	Job_Title	Country
1	120102	Tom	Zhou	M	108255	Sales Ma	AU
2	120103	Wilson	Dawes	M	87975	Sales Ma	AU
3	120121	Irenie	Elvish	F	26600	Sales Re	AU
4	120122	Christin	Ngan	F	27475	Sales Re	AU
5	120123	Kimiko	Hotstone	F	26190	Sales Re	AU
6	120124	Lucian	Daymond	M	26480	Sales Re	AU

데이터 읽기_실습

```
DATA subset;  
infile 'F:\data\sales.txt' dlm=',';LENGTH First_name $ 10;  
INPUT Employee_ID  
       First_Name $ Last_name $ Gender $  
       Salary      Job_Title $ Country $  
;RUN;  
data a;retain employee_ID First_name ;set subset;run;  
PROC PRINT data=a;  
RUN;
```

OBS	First_name	Employee_ID	Last_name	Gender	Salary	Job_Title	Country
1	Tom	120102	Zhou	M	108255	Sales Ma	AU
2	Wilson	120103	Dawes	M	87975	Sales Ma	AU
3	Irenie	120121	Elvish	F	26600	Sales Re	AU
4	Christina	120122	Ngan	F	27475	Sales Re	AU
5	Kimiko	120123	Hotstone	F	26190	Sales Re	AU
6	Lucian	120124	Daymond	M	26480	Sales Re	AU

OBS	employee_ID	First_name	Last_name	Gender	Salary	Job_Title	Country
1	120102	Tom	Zhou	M	108255	Sales Ma	AU
2	120103	Wilson	Dawes	M	87975	Sales Ma	AU
3	120121	Irenie	Elvish	F	26600	Sales Re	AU
4	120122	Christina	Ngan	F	27475	Sales Re	AU
5	120123	Kimiko	Hotstone	F	26190	Sales Re	AU
6	120124	Lucian	Daymond	M	26480	Sales Re	AU

데이터 읽기_실습

- List Input : 비표준데이터 처리 → 파일 : sales.txt

```
DATA subset3;  
infile 'd:\data\sales.txt' dlm=';';  
INPUT Employee_ID  
       First_Name $ Last_name $ Gender $  
       Salary      Job_Title $ Country $  
       Birth_Date :date. Hire_Date :mmdyy.  
;RUN;  
PROC PRINT data=subset3;  
RUN;
```

** : 포맷 문자형 자료를 읽을 때
지정길이에 관계없이 처음 공백이
나올 때까지 읽음



OBS	Employee_ID	First_Name	Last_name	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
1	120102	Tom	Zhou	M	108255	Sales Ma	AU	3510	10744
2	120103	Wilson	Dawes	M	87975	Sales Ma	AU	-3996	5114
3	120121	Irenie	Elvish	F	26600	Sales Re	AU	-5630	5114
4	120122	Christin	Ngan	F	27475	Sales Re	AU	-1984	6756
5	120123	Kimiko	Hotstone	F	26190	Sales Re	AU	1732	9405
6	120124	Lucian	Daymond	M	26480	Sales Re	AU	-233	6999

데이터 읽기_실습

■ 구분자 사례(&)

→ 파일 : special.txt

```
Region&State&Month&Expenses&Revenue  
Southern&GA&JAN2001&2000&8000  
Southern&GA&FEB2001&1200&6000  
Southern&FL&FEB2001&8500&11000  
Northern&NY&FEB2001&3000&4000  
Northern&NY&MAR2001&6000&5000  
Southern&FL&MAR2001&9800&13500  
Northern&MA&MAR2001&1500&1000
```

[Program]

```
PROC import datafile= "d:\data\special.txt"  
  out=mydata  dbms=dlm  replace;  
  delimiter='&';  
  getnames=yes;  
run;
```

→ 폴더 지정

```
options nodate ps=60 ls=80;  
proc print data=mydata;  
run;
```

데이터 읽기

■ 구분자 사례(&) 결과

OBS	Region	State	Month	Expenses	Revenue
1	Southern	GA	JAN2001	2000	8000
2	Southern	GA	FEB2001	1200	6000
3	Southern	FL	FEB2001	8500	11000
4	Northern	NY	FEB2001	3000	4000
5	Northern	NY	MAR2001	6000	5000
6	Southern	FL	MAR2001	9800	13500
7	Northern	MA	MAR2001	1500	1000

데이터 읽기

■ LENGTH 문장

- ✓ 변수 길이를 지정 (자료값의 손실을 방지)
- ✓ 숫자형은 2~8 바이트, 문자형은 1~32,767 까지 지정

```
LENGTH 변수명 $ length
```

```
DATA newlength;  
SET mylib.internationaltours;  
    LENGTH Remarks $ 30;  
    if Vendor = 'Hispania' then Remarks = 'Bonus for 10+ people';  
    else if Vendor = 'Mundial' then Remarks = 'Bonus points';  
    else if Vendor = 'Major' then Remarks = 'Discount for 30+ people';  
  
RUN;
```

데이터 읽기

■ 외부 텍스트 데이터 읽기

- ✓ 원자료가 보조기억장치에 있어서 자료를 SAS 프로그램과 함께 입력하지 않고, 따로 작성한 경우에는 INFILE문을 사용하여 외부데이터를 읽을 수 있음
- ✓ 한가지 주의해야 할 것은 INFILE은 반드시 INPUT 문 앞에 와야 한다

■ 예제

- ✓ 데이터 : 20명의 통계학과 수강생들에 대한 기초조사 결과
- ✓ 연령(세), 성별(1 남자, 2 여자), 키 (cm), 체중(kg),
- ✓ 즐기는 음식(1 육류, 2 생선류, 3 채소류)을 조사한 결과

Data " d:\data\sample1.txt "

데이터 읽기 예제

■ 프로그램

→ 파일 : sample1.txt

```
DATA sample1;  
infile " d:\data\sample1.txt ";  
INPUT index age gender height weight food;  
PROC PRINT DATA=sample1;  
RUN;
```

■ 결과 (일부)

OBS	index	age	gender	height	weight	food
1	1	30	1	183	82	1
2	2	28	2	160	62	3
3	3	27	1	178	77	2
4	4	23	1	172	70	2
5	5	25	1	168	72	3
6	6	27	1	179	77	1
7	7	26	1	169	71	1

데이터 읽기

■ INFILE 문에 사용되는 옵션

- ✓ 외부파일명 : 인용부호(' 이나 ") 내에 파일이름 지정
(경로까지 모두 지정)
- ✓ FIRSTOBS= 라인수 : 외부파일을 읽기 할 시작 위치(행번호)를 지정
- ✓ OBS= 라인수 : 외부파일을 읽기 할 끝 위치(행번호)를 지정
- ✓ LRECL=N : 읽고자 하는 레코드 폭을 지정(기본값은 132 열)
- ✓ PAD : 가변 길이 레코드를 읽을 때 , LRECL과 함께 사용
- ✓ MISSOVER : 자릿값을 읽지 못하는 변수에 대해서는 결측값으로 처리하도록 지정
- ✓ STOPOVER : 읽고자 하는 레코드에 결측값이 있으면 데이터생산 중단

데이터의 길이가 매우 긴 경우 (136 컬럼을 넘어가는 경우)에는 LRECL의 값을 크게 줌 [예를 들면, "LRECL=30000 PAD" 옵션을 줌]

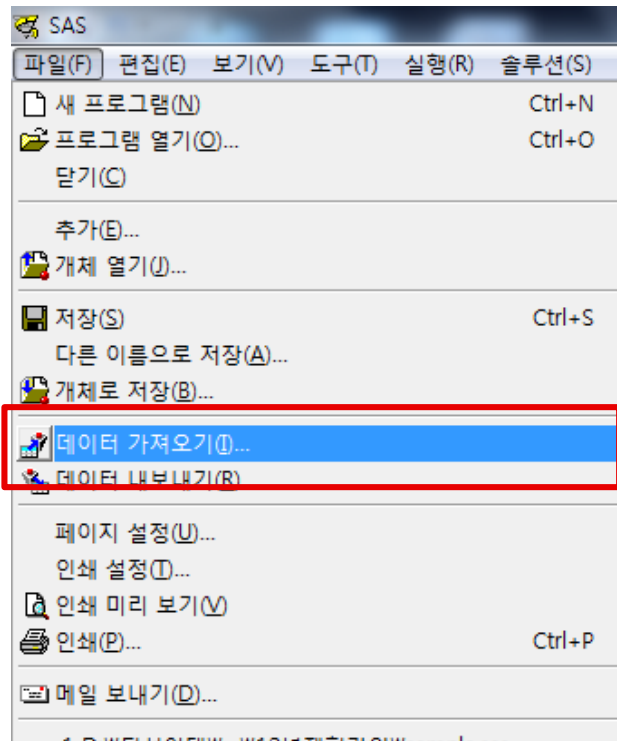
데이터 읽기

■ IMPORT WIZARD를 활용하여 읽기

- ✓ 공백이 구분자인 텍스트 파일
- ✓ [Data]d:\data\sample1.txt

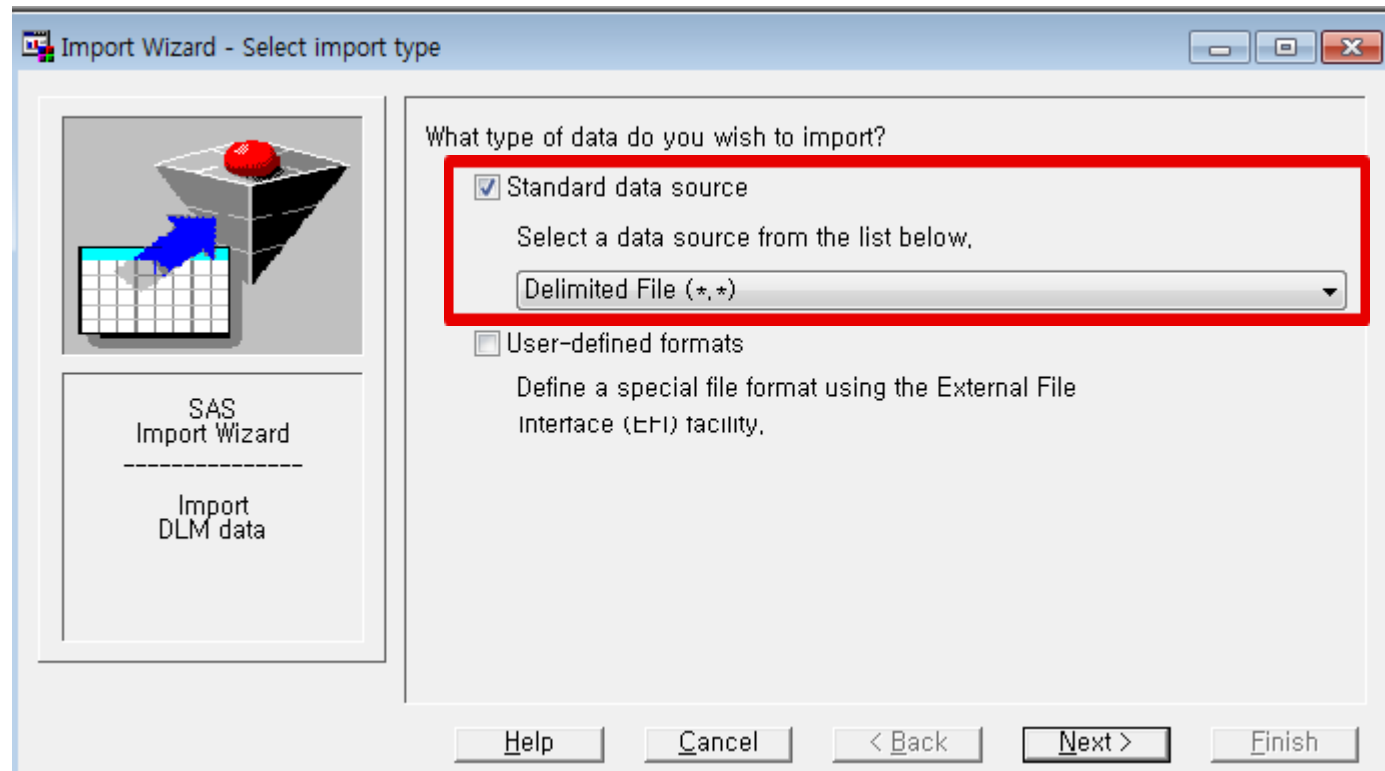
→ 파일 : sample1.txt

- ✓ [파일] > [데이터 가져오기]



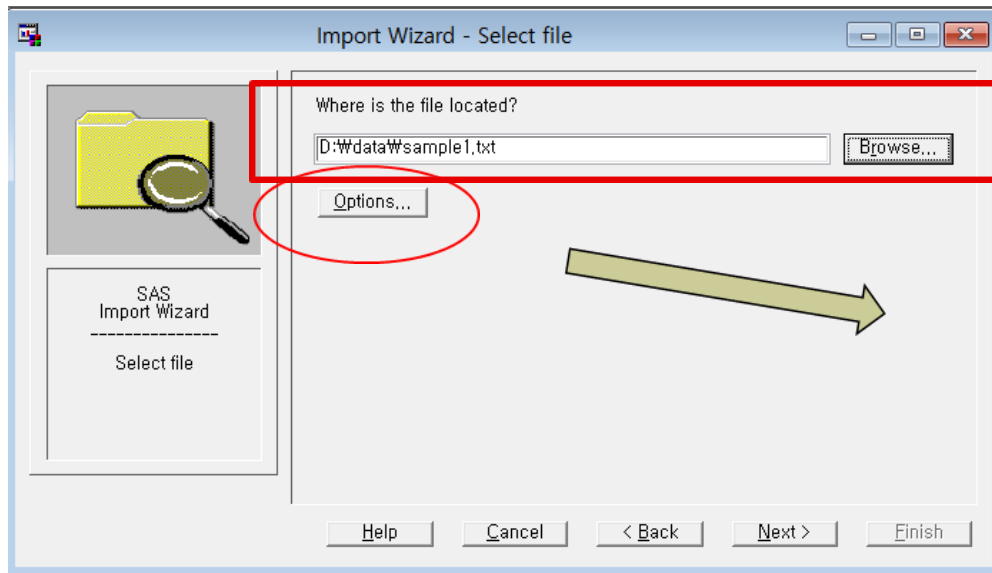
데이터 읽기

- ✓ [Standard data source]에 체크
- ✓ 불러올 파일 종류를 선택 : 공백이 구분자인 파일이므로 Delimited File 선택

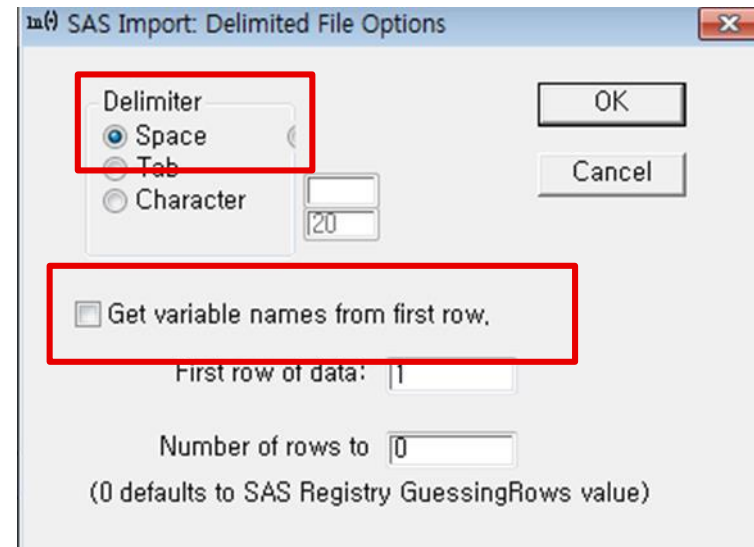


데이터 읽기

- ✓ 파일 선택 후 [Options] 단추를 클릭
- ✓ Delimiter 에 Space 체크
- ✓ 첫 행부터 데이터이므로 [get variable names from first row] 항에 체크를 하지 않음

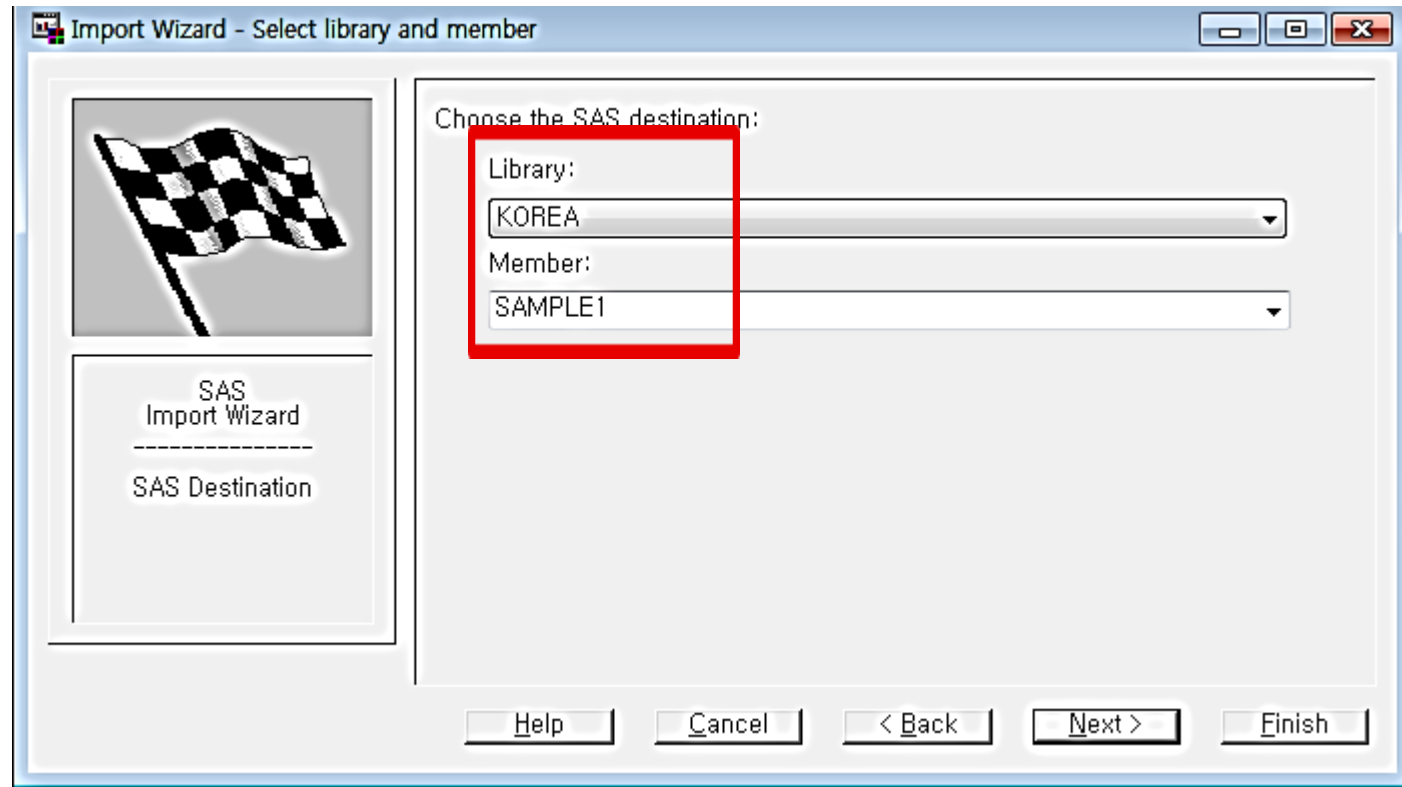


- ✓ 만약 읽을 파일이 Tab이 구분자인 경우 [Options]에서 Delimiter 에 Tab 체크




데이터 읽기

- ✓ 저장하고자 하는 폴더를 라이브러리에서 선택
- ✓ 파일 이름도 설정



데이터 읽기

- ✓ 라이브러리에서 저장 확인
- ✓ VIEWTABLE 확인

탐색기		VIEWTABLE: Korea					
'Korea'의 내용		VAR1	VAR2	VAR3	VAR4	VAR5	VAR6
 Sample1	1	1	30	1	183	82	1
	2	2	28	2	160	62	3
	3	3	27	1	178	77	2
	4	4	23	1	172	70	2
	5	5	25	1	168	72	3
	6	6	27	1	179	77	1
	7	7	26	1	169	71	1
	8	8	29	1	171	75	3
	9	9	34	2	158	60	2
	10	10	31	1	183	77	3
	11	11	26	2	162	59	1
	12	12	26	1	173	70	2
	13	13	35	1	173	68	3
	14	14	24	1	176	66	3
	15	15	29	2	170	70	2
	16	16	33	1	177	72	2
	17	17	38	2	159	55	1
	18	18	26	1	166	69	3
	19	19	26	1	169	66	2
	20	20	28	2	159	60	2

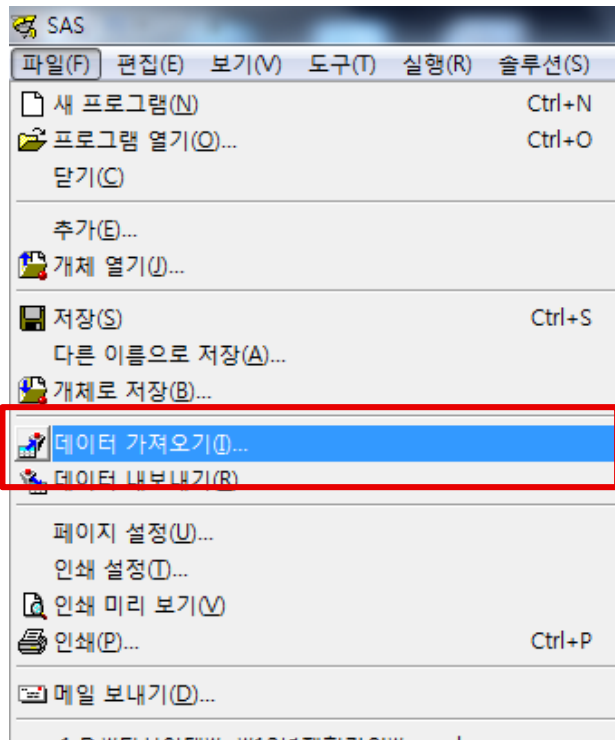
데이터 읽기

■ IMPORT WIZARD를 활용하여 읽기

- ✓ 공백이 구분자(&)인 텍스트 파일
- ✓ [Data] **d:\data\special.txt**

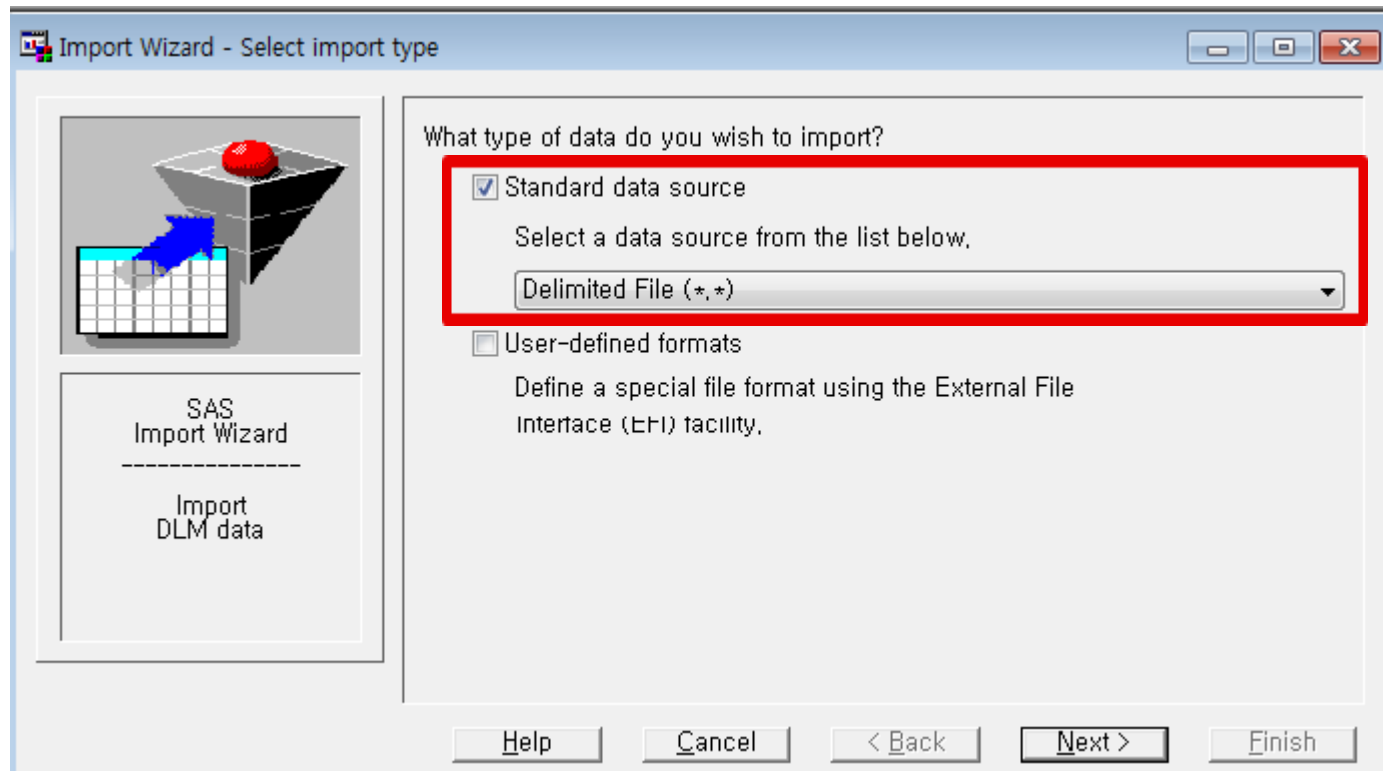
→ 파일 : special.txt

- ✓ [파일] > [데이터 가져오기]



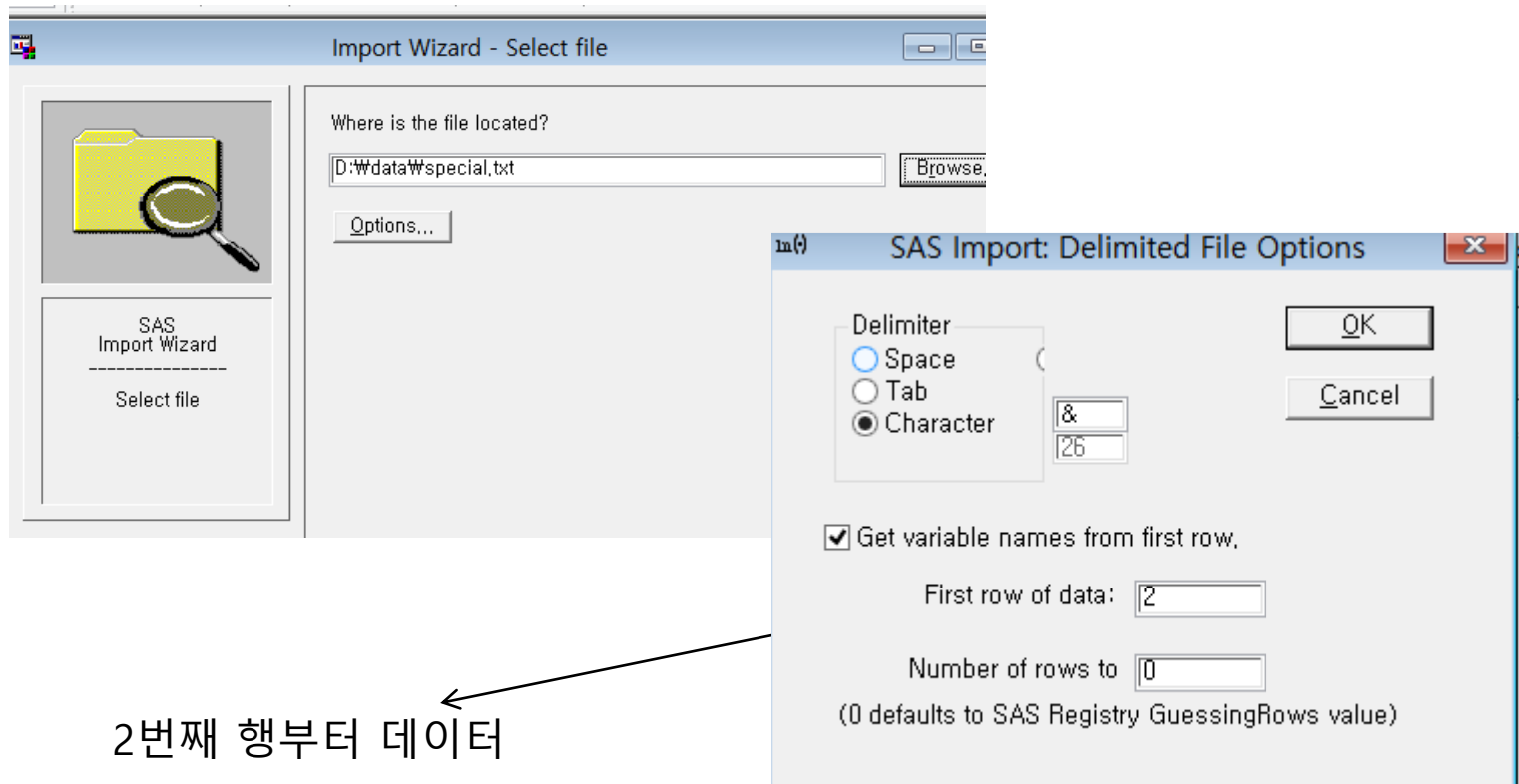
데이터 읽기

- ✓ [Standard data source]에 체크
- ✓ 불러올 파일 종류를 선택 : 공백이 구분자인 파일이므로 Delimited File 선택



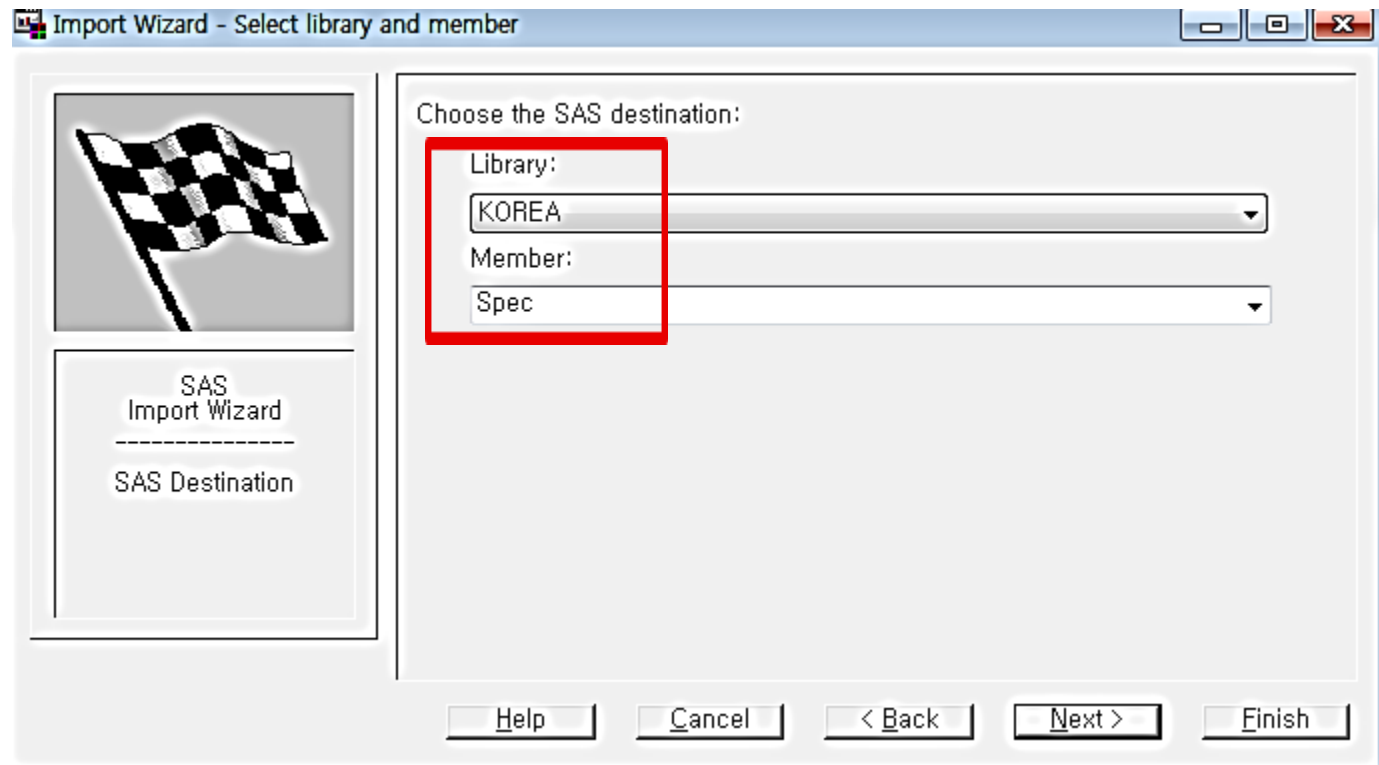
데이터 읽기

- ✓ 파일 선택 후 [Options] 단추를 클릭
- ✓ Delimiter 에 character 체크, "&" 입력
- ✓ 첫 행이 변수명이므로 [get variable names from first row] 항에 체크



데이터 읽기

- ✓ 저장하고자 하는 폴더를 라이브러리에서 선택
- ✓ 파일 이름도 설정



데이터 읽기

- ✓ 라이브러리에서 저장 확인
- ✓ VIEWTABLE 확인

탐색기

'Korea'의 내용

Sample1 Sample3 Spec

로그 - (제목없음)

VIEWTABLE: Korea.Spec

	Region	State	Month	Expenses	Revenue
1	Southern	GA	JAN2001	2000	8000
2	Southern	GA	FEB2001	1200	6000
3	Southern	FL	FEB2001	8500	11000
4	Northern	NY	FEB2001	3000	4000
5	Northern	NY	*****	6000	5000
6	Southern	FL	*****	9800	13500
7	Northern	MA	*****	1500	1000

VIEWTABLE: Korea.Spec

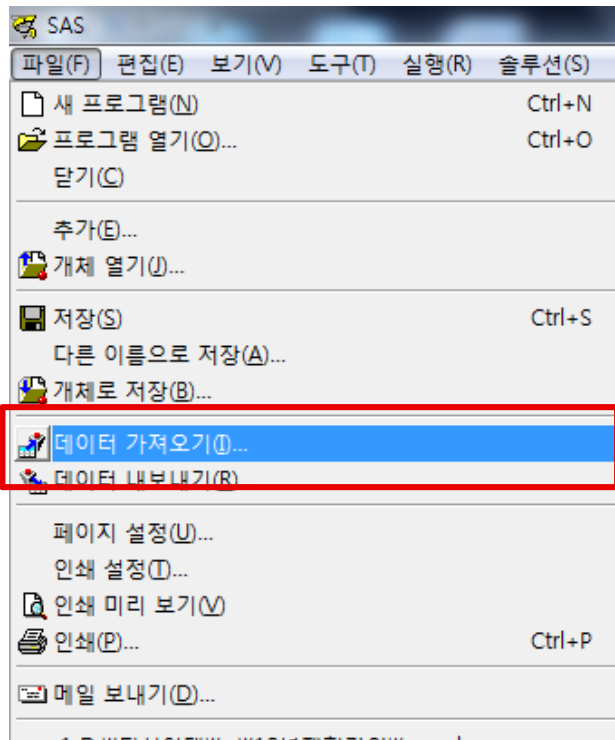
	Region	State	Month	Expenses	Revenue
1	Southern	GA	JAN2001	2000	8000
2	Southern	GA	FEB2001	1200	6000
3	Southern	FL	FEB2001	8500	11000
4	Northern	NY	FEB2001	3000	4000
5	Northern	NY	MAR2001	6000	5000
6	Southern	FL	MAR2001	9800	13500
7	Northern	MA	MAR2001	1500	1000

IMPORT WIZARD를 활용하여 EXCEL 파일 읽기

- ✓ 엑셀 파일

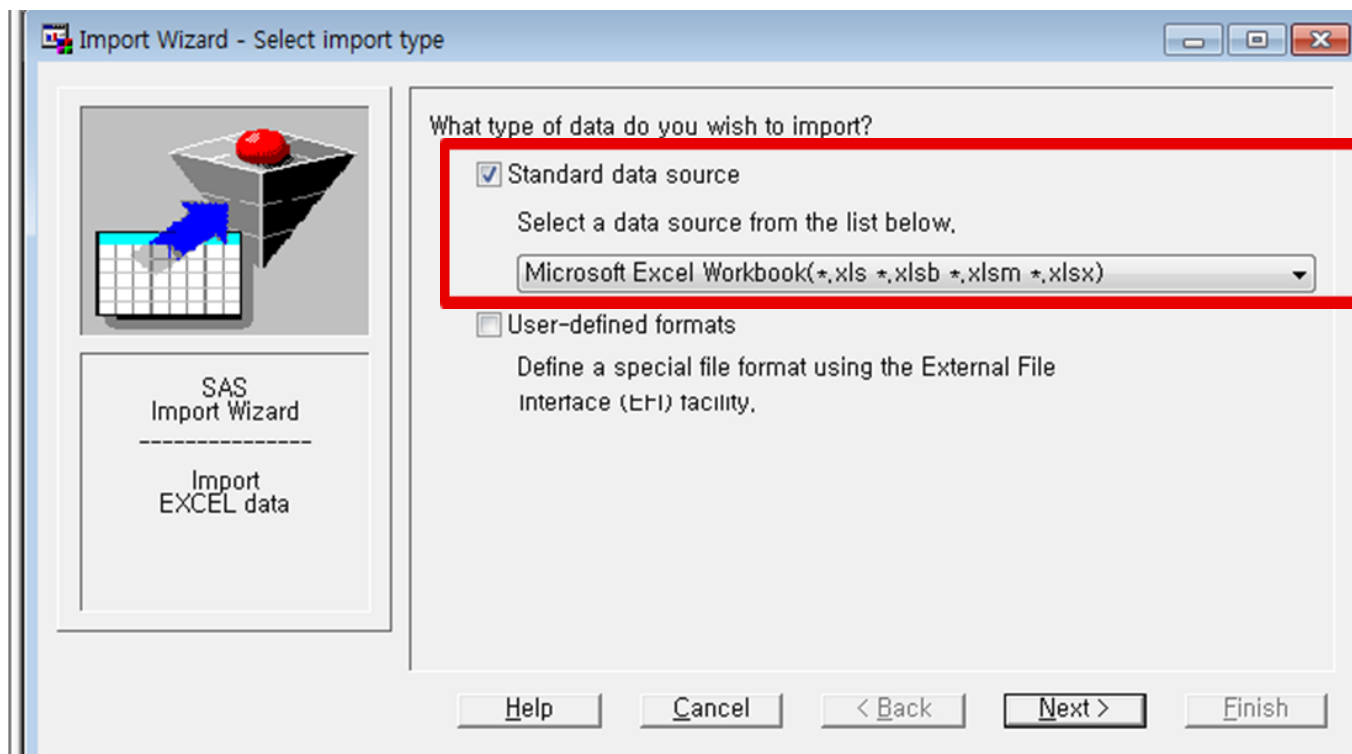
[Data] "d:\data\sample1.xls"

- ✓ [파일] > [데이터 가져오기]



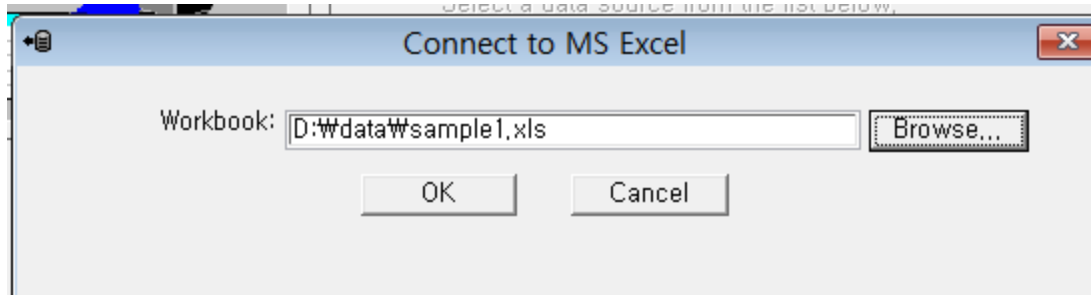
데이터 읽기

- ✓ [Standard data source]에 체크
- ✓ 불러올 파일 종류를 선택 : Microsoft Excel Workbook 선택

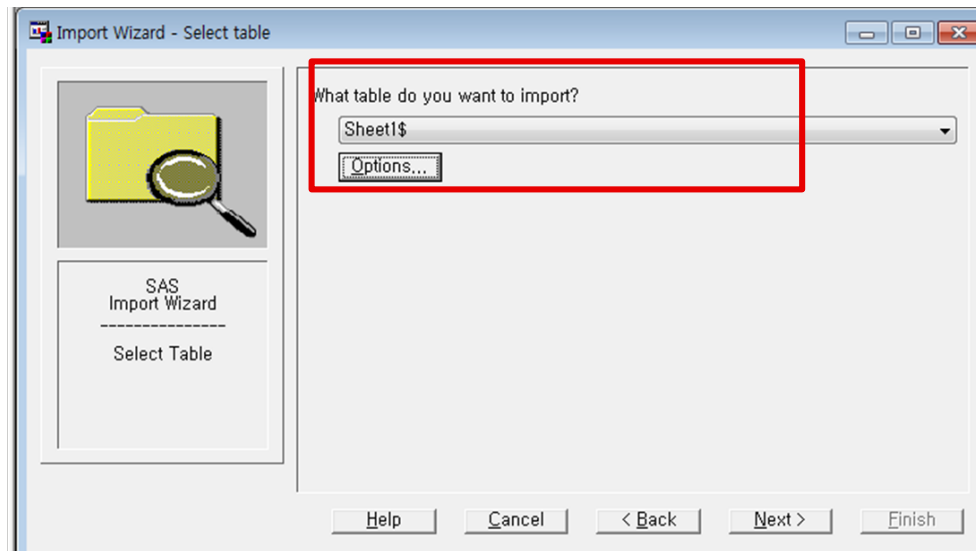


데이터 읽기

- ✓ 파일 선택 후 [OK] 단추를 클릭

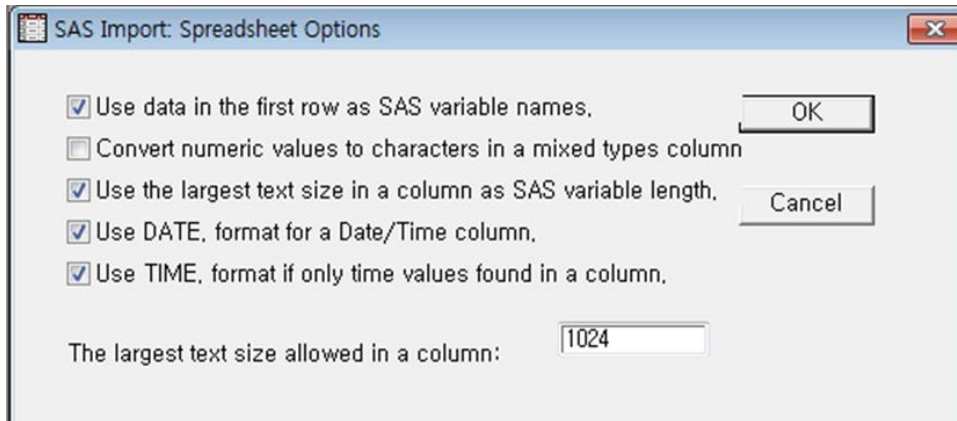


- ✓ 불러오기 원하는 시트 선택(sheet1)

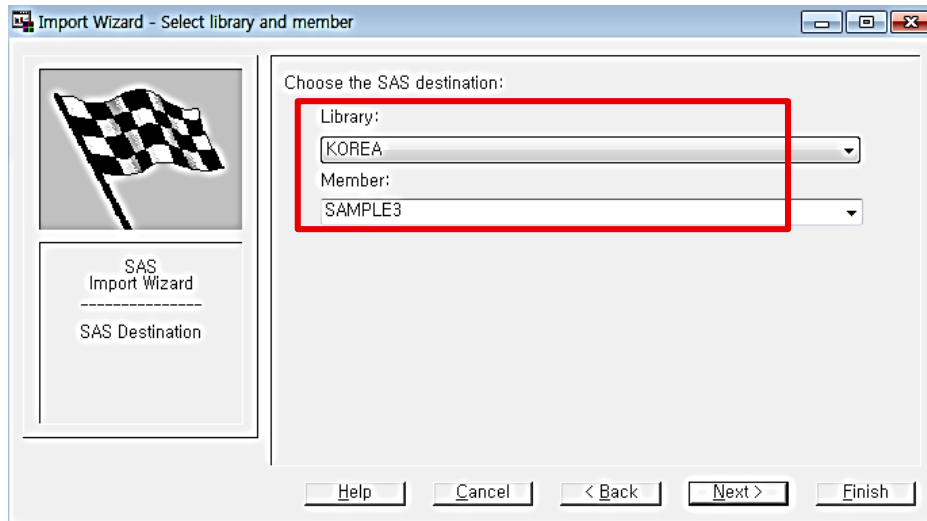


데이터 읽기

- ✓ 불러오고자 하는 시트에 대한 옵션 선택

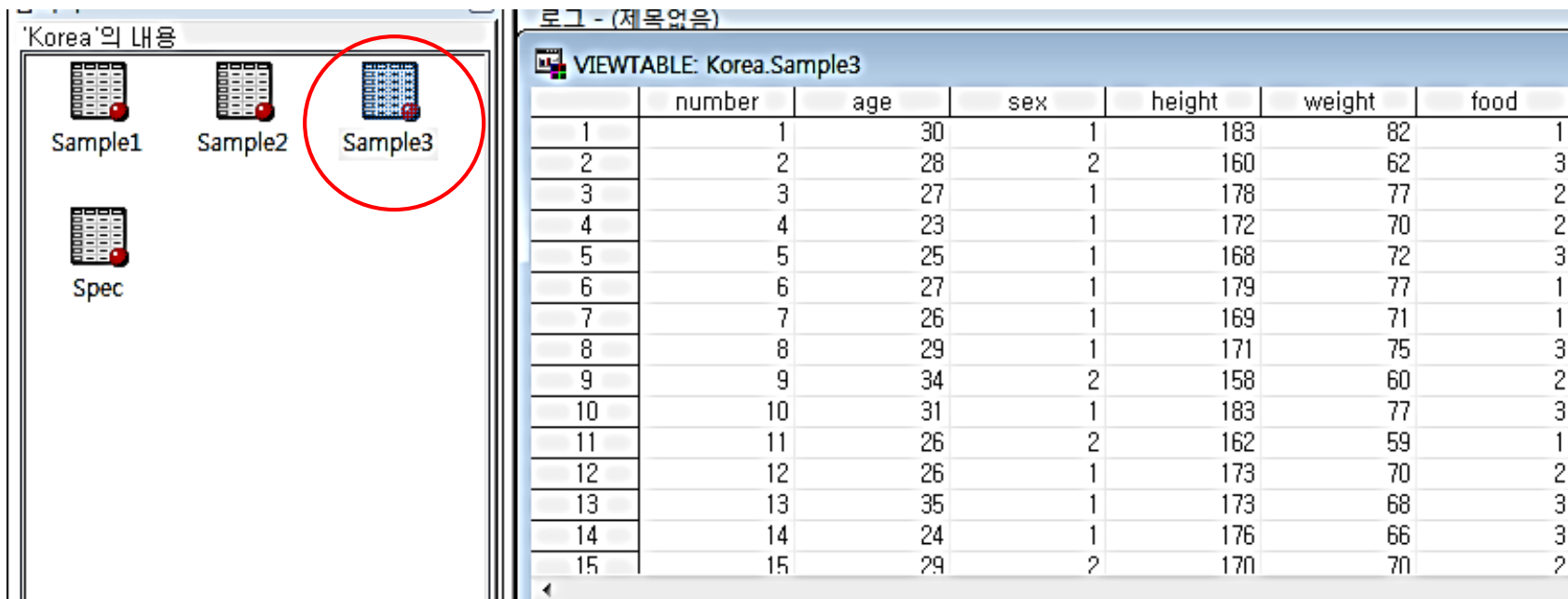


- ✓ 저장하고자 하는 라이브러리 및 파일명 선정



데이터 읽기

- ✓ 라이브러리에서 저장 확인
- ✓ VIEWTABLE 확인



로고 - (제목없음)

VIEWTABLE: Korea.Sample3

	number	age	sex	height	weight	food
1	1	30	1	183	82	1
2	2	28	2	160	62	3
3	3	27	1	178	77	2
4	4	23	1	172	70	2
5	5	25	1	168	72	3
6	6	27	1	179	77	1
7	7	26	1	169	71	1
8	8	29	1	171	75	3
9	9	34	2	158	60	2
10	10	31	1	183	77	3
11	11	26	2	162	59	1
12	12	26	1	173	70	2
13	13	35	1	173	68	3
14	14	24	1	176	66	3
15	15	29	2	170	70	2