

1985-7

SAS를 이용한 DB자료의 분석

—Interactive SAS를 중심으로—

1985. 12.

조사통계국

1985

알 림

당국에서는 그동안 개발해 온 주요통계 Data Base에 통계 분석 Package 인 SAS(Statistical Analysis System)를 연결하여 D/B내에 축적된 자료를 편리하게 분석 이용할 수 있도록 하였습니다.

이 책은 이용자가 이와같이 D/B에 연결된 SAS를 활용하여 각종 통계분석을 실시하는데 알아야 할 사용법을 소개하는 안내서입니다.

이 책을 사용하는 중에 의문이 있거나 보다 상세한 내용에 관심이 있으신 분은 자료관리과로 연락하시기 바랍니다.

자 료 관 리 과 장

연락처 720-2788, 2789

이 준 형

신 호 중

목 차

1. SAS Program의 기초	9
가. 작업순서	11
나. Program Edit 화면을 찾는 법	12
다. SAS Program 작성 요령	17
라. SAS Program 수행	21
마. SAS Program 에러 확인 및 수정	26
바. 작업종료 요령	30
사. 평균계산	34
아. 상관분석	41
자. 모든 기술 통계치 계산	44
차. dot수분포, PLOTTING, 정규분포성 검증	46
카. Histogram (막대그래프)	51
타. 추세선 작성	52
파. Table 및 Histogram 작성	53
2. DB를 이용한 SAS Program 작성	57
가. 작업요령	59
나. DB와 SAS 연결	60
다. Sample Dataset Allocate	61
라. Sample Program Copy 및 수행결과	62
(1) Source Program을 출력하는 Sample Program(EXEM 1).....	63
(2) DB자료를 출력해주는 Sample Program(EXEM2)	65

(3) DB자료를 읽어 새로운 Dataset 를 만드는 Sample Program (EXEM3)	69
(4) 회귀분석 Sample Program (EXEM4)	73
(5) 시계열자료 Plotting (EXEM5)	77
(6) Forecasting (EXEM6)	79
3. Batch 방식 SAS 이용법	87
가. 작업순서	89
나. Program Edit 화면을 찾는 법	90
다. SAS Program 작성	93
라. 작업수행	96
마. 작업종료	101
4. 분석기능별 SAS 명령어	105
가. 기술통계분석	107
(1) 평균 (MEANS)	107
(2) 상관분석 (CORR)	107
(3) 숫수분포표 (FREQ)	107
(4) Summary 통계치 (SUMMARY)	108
(5) 모든 기술 통계치 (UNIVARIATE)	108
(6) 막대도표, 블록도표, 원도표 및 별도표 (CHART)	109
(7) 변수간의 점도표 (PLOT)	109
(8) 시계열자료 PLOTTING (TIMEPLOT)	110
(9) 기술통계표 작성 (TABULATE)	110

(10) 순위 작성 (RANK)	110
(11) 표준화 점수 작성 (SAANDARD)	111
(12) 두 집단간 평균치 차 검증 (TTEST)	111
(13) BMDP를 이용한 통계분석 (BMDP)	111
나. 회귀분석기법	112
(1) 회귀분석 (REG)	112
(2) 모든 가능한 회귀요청에 대한 규명 (RSQUARE)	112
(3) 단계별 회귀분석 (STEPWISE)	113
(4) 고장시간의 회귀모형 규명 (LIFEREG)	113
(5) 라이프 테스트 (LIFETEST)	114
(6) 비 선형 회귀분석 (NLIN)	114
(7) 반응표면 회귀분석 (RSREG)	115
(8) Probit 모형 추정 (PROBIT)	115
다. 분산분석기법	116
(1) 분산분석 (ANOVA)	116
(2) 일반선형 모형 분석 (GLM)	116
(3) 실험설계 (PLAN)	117
(4) Nested 분산분석 (NESTED)	117
(5) 분산인자추정 (VARCOMP)	118
라. 요인분석기법	119
(1) 요인분석 (FACTOR)	119
(2) 주성분 분석 (PRINCOMP)	120
(3) 점수추출 (SCORE)	120

마. 정준분석 (CANCORR)	121
바. 판별분석기법	122
(1) 판별분석 (DISCRIM)	122
(2) 정준판별분석 (CANDISC)	122
(3) 단계별 판별분석 (STEPDISC)	123
(4) 최근방거리 판별분석 (NEIGHBOR)	123
사. 집락분석기법	124
(1) 집락분석 (CLUSTER)	124
(2) 간편 집락분석 (FASTCLUS)	124
(3) Tree Diagram 작성 (TREE)	125
(4) 공분산 집락분석 (VARCLUS)	125
(5) 집락내 합동 등공분산추정 (ACECLUS)	126
(6) 유목별 자료분석 (CATMOD)	126
아. 비 모수 통계분석 (NPAR1WAY)	127
자. 시계열분석 기법	128
(1) Box-Jenkins 시계열분석 (ARIMA)	128
(2) 자기회귀 시계열모형 추정 (AUTOREG)	128
(3) 시계열예측 (FORECAST)	129
(4) 시계열의 계절조정 (X11)	129
(5) 시차분포모형 추정 (PDLREG)	130
(6) 시계열의 Spectral 밀도함수 추정 (SPECTRA)	130
(7) 상태공간 벡터 추정 (STATESPACE)	130
차. 시스템 모델링 기법	131

(1) 모형정립 (MODEL)	131
(2) 선형연립방정식모형 추정 (SYSLIN)	131
(3) 선형연립방정식모형 Simulation (SIMLIN)	131
(4) 비선형 연립방정식모형 추정 (SYSNLIN)	132
(5) 비선형 연립방정식모형 Simulation (SIMNLIN)	132
카. 기타 통계분석	133
(1) Citi Bank Data Base (CITIBASE)	133
(2) Table 정리 및 계산 (COMPUTAB)	133
(3) 저당 및 이자율 계산 (MORTGAGE)	133
(4) 재무분석계산에 유용한 함수들	134
타. Dataset 처리	135
(1) SAS Dataset에 따른 SAS Dataset 추가 (APPEND).....	135
(2) SAS Dataset 내용 Read (BROWSE)	135
(3) 2개의 SAS Dataset 비교 (COMPARE)	135
(4) SAS file 내용 Print (CONTENTS)	135
(5) SAS Library Member Copy (COPY)	136
(6) SAS Library Member List 작성, 이름변경, 삭제 (DATASETS)	136
(7) SAS Dataset 조사변경 (EDITOR)	136
(8) SAS Dataset 내의 변수 및 변수값 Print (PRINT)	137
(9) SAS Dataset Sort (SORT)	137
(10) SAS Dataset 의 변수와 변수값의 위치변경 (TRANSPPOSE)	137
(11) 출력양식정의 (FORMAT)	137

(12) 정형적인 출력양식 Print(FORMS)	137
(13) 월별 일정표 작성(CALENDAR)	138
(14) SAS OPTION LIST(OPTIONS)	139
(15) PDS Dataset 의 Member List 작성, 이름변경, 삭제(PDS)	
(16) PDS Dataset Copy(PDSCOPY)	138
(17) Disk Dataset Space Release(RELEASE)	138
(18) Backup, Member List 및 Directory 작성(SOURCE)	139
(19) Tape Copy(TAPECOPY)	139
(20) Tape Label Print (TAPELABEL)	139
(21) 상이한 OS하의 SAS Dataset 변환(XCOPY)	139
(22) BMDP, DATATEXT, OSIRIS, SPSS 및 Version 72 SAS 등의 Dataset 을 현재의 SAS와 호환성 있는 Dataset으로 변환(CONVERT)	140
(23) SAS 처리결과를 특정 File에 출력(PRINTTO)	140

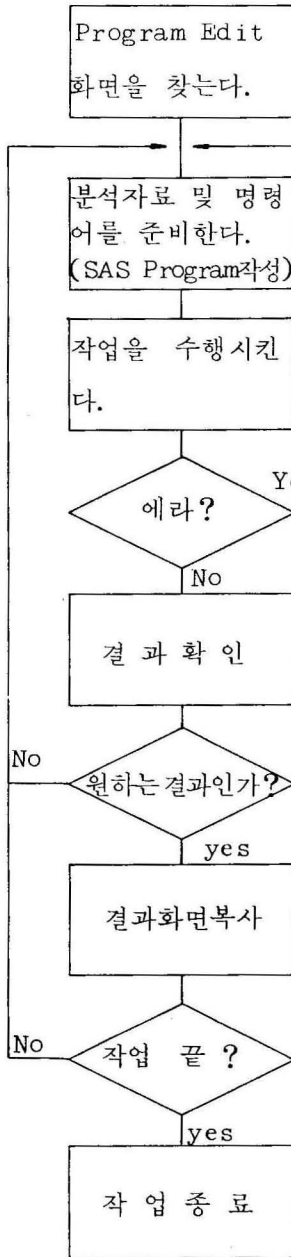
부 록

(1) SAS 명령어 사용규약	143
(2) Data Step에서 사용하는 명령어	144
(3) Proc Step에서 사용하는 명령어	150
(4) Data 및 Proc Step 모두에 사용하는 명령어	151
(5) SAS Data set 정의문	153
(6) SAS Option List	154
(7) 변수 표현방법	155
(8) Pointer 제어기호	155

(9) Line Hold 에 사용되는 기호	155
(10) INPUT format 지시기호	155
(11) 연산자의 처리 순서	156
(12) SAS Data 입력 형식	156
(13) SAS Data 출력 형식	156
(14) SAS에서 사용가능 함수	157
(15) Funtion Key	159
(16) Command	160
(17) Macro어 문장	162
(18) Macro어 함수	162
(19) Macro어와 함께 Data Step에서 사용되는 함수	162
(20) 자동 Macro 변수들	163

1. SAS Program의 기초

가. 작업순서



SAS를 이용하여 통계분석을 하려면 먼저 Program Editor라는 화면을 찾는다.

Program Editor 화면에 분석대상자료 및 분석명령어를 SAS 규칙에 따라 작성한다.

SAS Program이 작성되면 이를 컴퓨터에 보내어 작업을 수행시킨다.

에라 메시지를 확인한다.

SAS Program에 에라가 있으면 결과가 나오지 않고 System Log 화면에 에라 메시지가 나온다. 이를 확인한후 SAS Program을 다시 부른다.

결과가 화면에 나오면 바라는 결과인지 여부를 확인하고 원하는 결과가 아니면 SAS Program을 다시 불러낸다.

결과표가 필요할때는 복사명령을 사용하여 복사한다.

SAS 작업을 끝내고자 할때는 작업종료요령에 따라 작업을 끝내고 계속하려면 Program Edit 화면으로 되돌아간다.

나. Program Edit 화면을 찾는 법

SAS 작업을 위해서는 Terminal의 최초화면 [나-1]에 'TSO' 를 Key in 하고, Key 를 누르면

[나-1]

```

#####
00
00  =====  =====  =====  00
000  =====  =====  =====  000
000  ==  ==  ==  ==  ==  ==  ==  ==  000
000  ==  ==  ==  ==  ==  ==  ==  ==  000
000  =====  =====  =====  000
000  =====  =====  =====  000
000  =====  =====  =====  000
00  E.P.B COMPUTER CENTER ==) ACF/VIA) V.2.1  00
00
#####
IXI) MVS EXPRESS *B403 IXI)
+-----+
| *** PLEASE ENTER TSO,CICS,CICS1 | 01
| OR YOUR LOGON COMMAND *** | 00
| ==) TSO | 00
+-----+ 00
#####

```

화면 [나-2]가 나온다.

[나-2]

```

IXJ56700A ENTER USERID
290

```

화면 [나-2]에 USER ID를 Key in 하고 Key 를 누르면 화면 [나-3]이 나온다.

[나-3]

```
----- VS2 REL 3.8 TIME SHARING OPTION -----
PF1/PF13 ==> Help   PF3/PF15 ==> Logoff  PA1 ==> Attention  PA2 ==> Reshow
You may request specific HELP information by entering a '?' in any entry field.
ENTER LOGON PARAMETERS BELOW:                RACF LOGON PARAMETERS:

  USERID   ==> Z90
  PASSWORD  ==>
  NEW PASSWORD ==>
  PROCEDURE ==> SASPROC                       GROUP IDENT ==>
  ACCT NMBR ==> 90
  SIZE      ==> 4096
  PERFORM   ==>
  COMMAND   ==>

  ENTER AN 'S' BEFORE EACH OPTION DESIRED BELOW:
          -NOMAIL          -NONOTICE          -RECONNECT          -OIDCARD
```

화면 [나-3]의 'PASSWORD ==>'란에 이용자의 PASSWORD를 Key in하고 Key 를 누르면 화면 [나-4]가 나온다.

[나-4]

```
ICH700011 Z90          LAST ACCESS AT 12:05:11 ON TUESDAY, NOVEMBER 19, 1985
IKJ564551 Z90 LOGON IN PROGRESS AT 12:36:35 ON NOVEMBER 19, 1985
IKJ569511 NO BROADCAST MESSAGES
***
```

화면 [나-4]가 나올 때까지 기다렸다가 화면 [나-4]의 맨 마지막 줄에 '***'가 나온 후, Key 를 누르면 화면 [나-5]가 나온다.

[나-5]

```
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ==> 6
                                USERID - Z90
0 ISPF PARMS - Specify terminal and user parameters    TIME - 12:38
1 BROWSE     - Display source data or output listings  TERMINAL - 3278
2 EDIT      - Create or change source data           PF KEYS - 24
3 UTILITIES - Perform utility functions
4 FOREGROUND - Invoke language processors in foreground
5 BATCH     - Submit job for language processing
6 COMMAND - Enter TSO command or CLIST
7 DIALOG TEST - Perform dialog testing
8 SDSF      - Spool display and search facility
C CHANGES  - Display summary of changes for this release
T TUTORIAL  - Display information about ISPF/PDF
X EXIT      - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF:
```

화면 [나-5]의 'OPTION ==>' 란에 '6'을 Key in 하고, ENTER Key를 누르면 화면 [나-6]이 나온다.

[나-6]

```
----- TSO COMMAND PROCESSOR -----
ENTER TSO COMMAND OR CLIST BELOW:
==> GSAS
```

화면 [나-6]의 '==>'란에 'GSAS'를 Key in 하고 ENTER Key를 누르면 화면 [나-7]이 나온다.

[나-7]

```
COMMAND ==> SAS(R) LOG 15.52
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE-TRIAL SITE (02122001).
NOTE: CPUID VERSION = 00 SERIAL = 090381 MODEL = 4361

-----
COMMAND ==> PROGRAM EDITOR
00001 -
00002 -
00003 -
00004 -
00005 -
00006 -
00007 -
00008 -
```

사용자는 화면 [나-7]에서 SAS Programming 을 할 수가 있으나 실제 SAS Program 작성 및 작업수행요령 등은 행을 바꾸어 설명하고 여기서는 상기화면의 내용을 설명하기로 한다.

화면 [나-7]의 점선을 중심으로 상단을 SAS Log 화면, 하단을 Program Edit 화면이라 칭한다.

① 각 화면의 기능

- Program Edit 화면

- 분석대상자료 및 분석명령어를 SAS Program 작성규칙에 따라 작성하고 SAS 프로그램 수행명령을 발할 수 있는 화면

- SAS Log 화면

- 이용자가 작성하여 수행시킨 SAS 프로그램을 해독하여 수행과정상 발생하는 정보 및 Error Message 를 이용자에게 전달해주는 화면

② 화면의 확장

- Program Edit 화면 확장

SAS 프로그램을 작성하거나 수정할 때는 Program Edit 화면을 넓

게 사용하는 것이 유리하다. 이때는 CURSOR를 SAS Log 화면 상단의 적당한 곳에 옮겨 놓고 **PF2** Key 를 누르면 Program Edit 화면은 CURSOR의 위치까지 확장된다.

— SAS LOG 화면의 확장

SAS Program 을 수행시키면 수행과정상의 여러가지 정보 및 SAS Program작성상의 에라 등이 화면에 나오게 되는데 특히 에라 메시지를 확인하기 위해서는 SAS LOG 화면을 넓히는 것이 유리하다. 이때는 CURSOR를 Program Edit 화면의 하단부로 옮기고 **PF2** Key 를 누르면 SAS LOG 화면이 CURSOR의 위치까지 확장된다.

다. SAS Program (대상자료 및 분석 명령어) 작성 요령

(1) SAS program의 구조

자 료 설 명 부 분	자료명, 변수명, 각 변수들의 크기 등이 명시된 부분
자 료 부 분	각 변수들의 실제자료를 기입하는 부분 (단 외부화일에 있는 자료를 이용할 때 는 화일명을 기입)
자 료 처 리 부 분	분석처리 명령어를 기입하는 부분

(2) 일반적인 문제구조

3-1 반 성 적

성 명	국 어	수 학	과 학
김 복 남	86.7	98.4	87.6
남 철 수	78.5	89.6	98.6
이 숙 자	98.4	76.7	87.4
김 정 구	89.5	95.4	96.4
유 민 회	85.3	79.5	92.1
황 용 구	92.4	98.3	91.0

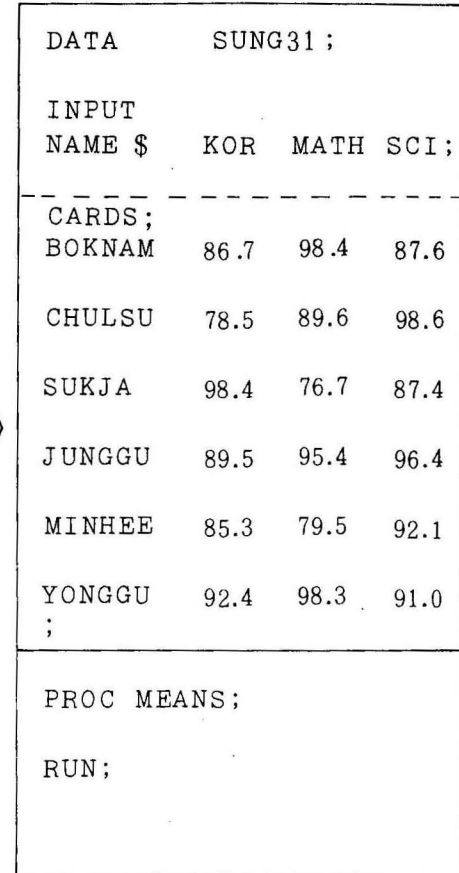
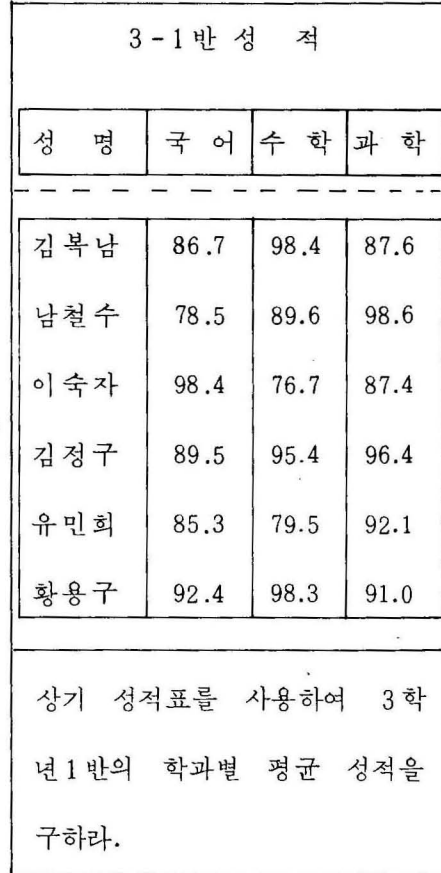
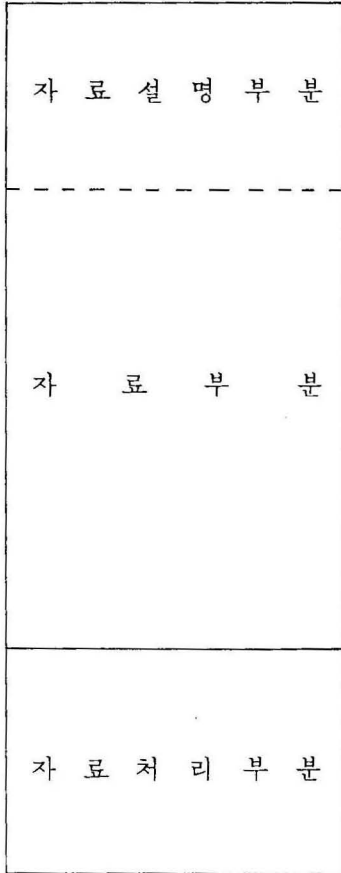
상기 성적표를 사용하여 3학년 1반의 학과별 평균성적을 구하라.

(3) SAS program의 작성

SAS Program 구조

SAS 프로그램 및 문제의 결합

SAS Program



① DATA SUNG31;

SAS Program에서 3-1반 성적으로 SAS DATASET를 만들어 SUNG31이란 이름으로 사용하겠다는 명령이다. DATA는 언제나 SAS Program의 첫머리에 나와야 하며 SUNG31은 3-1반 성적에 대하여 사용자가 임의로 정해서 사용하는 이름으로 DATASET NAME이라고도 칭한다. Dataset Name 다음에는 반드시 ;이 나와야 한다.

② INPUT NAME \$ KOR MATH SCI;

SAS DATASET을 만들기 위하여 SAS의 분석대상이 되는 자료를 성명(NAME \$), 국어점수(KOR), 수학점수(MATH), 과학점수(SCI) 순으로 읽어 들이라는 명령으로 자료는 반드시 여기서 정해진 배열순으로 입력되어야 한다.

INPUT은 자료를 읽으라는 명령어로 주어진대로 사용해야 하며 NAME, KOR, MATH, SCI 등은 사용자가 성명, 국어점수, 수학점수, 과학점수 등에 대하여 임의로 정해진 변수명이다.

변수명은 8자 이내의 영문 또는 숫자를 사용하되 첫글자는 반드시 영문자를 사용해야 한다.

NAME 다음의 \$는 NAME이라는 변수가 문자로 된 자료임을 알려주는 부호이다. ;는 INPUT문의 끝임을 알려준다.

③ CARDS ;

SAS에서 INPUT 명령으로 읽어들이는 자료가 CARD나 터미널상으로 입력됨을 알려주는 명령어이다. Tape나 Disc 입력은 INFILE이란 명령을 사용하나 여기서는 설명을 생략한다.

④

BOKNAM	86.7	98.4	87.6
:	:	:	:
YONGGU	92.4	98.3	91.0
:			

INPUT 명령으로 읽어들이는 자료가 성명 (NAME), 국어점수 (KOR), 수학점수 (MATH), 과학점수 (SCI) 순으로 반복되어 나타나 있음을 알 수 있다. 맨 마지막 줄의 ;은 자료의 끝임을 알려주는 부호로써 반드시 표시해 주어야 한다.

⑤ PROC MEANS;

평균을 구하라는 명령어이다. 이 PROC문은 사용자의 필요에 따라 몇 번이고 반복사용이 가능하다.

⑥ RUN;

상기 명령어들을 실제로 처리하라는 명령어로서 반드시 나와야 한다.

라. SAS Program 수행

아래 화면 [라-1]은 SAS Program 작성요령에 따라 작성된 SAS Program이다.

[라-1]

```

COMMAND ==>                                     SAS(R) LOG 13 41
-----
COMMAND ==>                                     PROGRAM EDITOR

00001 DATA SUNG31 ;
00002 INPUT NAME $ KOR MATH SCI ;
00003 CARDS ;
00004 BOKNAM 86.7 98.4 87.6
00005 CHULCU 78.5 89.6 98.6
00006 SUKJA 98.4 76.7 87.4
00007 JUNGGU 89.5 95.4 96.4
00008 MINHEE 85.3 79.5 92.1
00009 YONGGU 92.4 98.3 91.0
00010
00011 PROC MEANS
00012 RUN
00013
00014
00015
00016
00017
00018

```

SAS Program을 수행시키려면 [PF3] (or [PF15]) Key를 누르거나 CURSOR를 Program Edit 화면의 'COMMAND==>'란으로 옮겨 'SUBMIT'를 Typing하고, [ENTER] Key를 누르면 작업이 수행되고 그 결과가 화면 [라-2]와 같이 나온다.

[라-2]

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	STD ERROR OF MEAN
KOR	6	88.4666667	6.74821434	78.5000000	98.4000000	2.75425513
MATH	6	89.6500000	9.54122634	76.7000000	98.4000000	3.89518921
SCI	6	92.1833333	4.56614352	87.4000000	98.6000000	1.86412029

결과를 Print하고자 할 때는 [Print Key] Key를 누르면 화면의 내용이 프린터에 인쇄되어 나온다.

결과표를 확인하고 다음 작업을 위하여 Program Edit 화면으로 되돌아가고자 할 때는 **PF3** (or PF15) Key를 누르면 화면 [라-3]이 나오는데 이때 만약 아직 화면에 나오지 아니한 작업결과가 남아있다면 남은 결과가 화면에 나오므로 결과가 다 나올 때까지 **PF3** Key를 눌러야 한다.

[라-3]

```

COMMAND ==> SAS(R) LOG 13:44
-----
COMMAND ==> PROGRAM EDITOR

00001
00002
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
00016
00017
00018
    
```

화면 [라-3]에서 방금 수행한 SAS Program을 다시 불러오고자 할 때는 **PF4** (or **PF16**) Key를 누르면 화면 [라-4]와 같이 SAS Program이 다시 나온다.

[라-4]

```

COMMAND ==> SAS(R) LOG 13:44
-----
COMMAND ==> PROGRAM EDITOR

00001 DATA SUNG31 ;
00002 INPUT NAME $ KOR MATH SCI ;
00003 CARDS ;
00004 BOKNAM 86.7 90.4 87.6
00005 CHULSU 78.5 89.6 98.6
00006 SUKJA 98.4 76.7 87.4
00007 JUNGJU 89.5 95.4 96.4
00008 MINHEE 85.3 79.5 92.1
00009 YONGGU 92.4 90.3 91.0
00010
00011 PROC MEANS ;
00012 RUN ;
00013
00014
00015
00016
00017
00018
    
```

화면 [라-4]에서는 사용자의 의사에 따라 내용을 수정하거나 새로운 프로그램을 작성할 수도 있다.

<참고> SAS Program Save요령

화면 [라-4]의 SAS Program을 나중에 다시 이용하려면 보관해 둘 필요가 있다. 이때는 사용자의 Library를 SAS와 연결해야 하는데 Library의 연결작업은 SAS Program을 시작할 때 한번만 해주면 된다.

[S - 1]

```

COMMAND ==>                                SAS(R) LOG 10 28
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C., 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE TRIAL SITE (08839001).
NOTE: CPUID   VERSION = 00   SERIAL = 080381  MODEL = 4381 .
-----
COMMAND ==>                                PROGRAM EDITOR

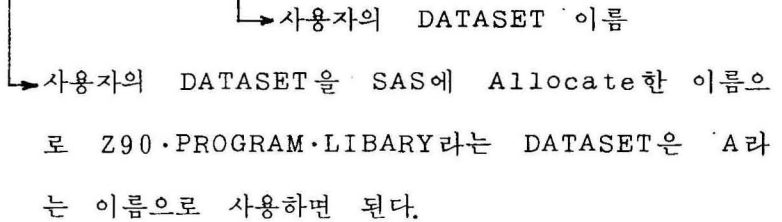
00001 TSO ALLOC F(A) DA(Z90.PROGRAM.LIBRARY) SHR
00002
00003
00004
00005
00006
00007
00008
    
```

화면 [S - 1]은 사용자의 Dataset인 ' Z90·PROGRAM·LIBRAY' 를 A 라는 이름으로 Allocate 하라는 뜻이다.

- ALLOCATE문 작성요령

```

TSO ALLOC F(A) DA(Z90·PROGRAM·LIBRARY) SHR;
    
```



'----'의 Underline에 그어진 부분은 주어진대로 Coding 한다.

PF3 (or PF15) Key를 누르면 화면 [S - 2]가 나온다.

[S - 2]

```
COMMAND ==> SAS(R) LOG 10:
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE TRIAL SITE (08839001).
NOTE: CPUID VERSION = 00 SERIAL = 080381 MODEL = 4381 .
      1 TSO ALLOC F(A) DA(Z90.PROGRAM.LIBRARY) SHR ;
-----
COMMAND ==> PROGRAM EDITOR
00001
00002
00003
00004
00005
00006
00007
00008
```

화면 [S - 2]에서 PF4 (or **PF16**) Key를 누르면 방금 사용한 SAS Program이 화면 [S - 3]과 같이 나온다.

[S - 3]

```
COMMAND ==> SAS(R) LOG 13 44
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE TRIAL SITE (08839001).
NOTE: CPUID VERSION = 00 SERIAL = 080381 MODEL = 4381 .
      1 TSO ALLOC F(A) DA(Z90.PROGRAM.LIBRARY) SHR ;
-----
COMMAND ==> SAVE A(MEMBER1) PROGRAM EDITOR
00001 DATA A;
00002 INPUT ENGLISH MATH SCIENCE NAME $ SEX $ ;
00003 CARDS;
00004 86.7 98.4 87.6 BOKMAM MALE
00005 78.5 89.6 98.6 CHULSU MALE
00006 98.4 76.7 87.4 YONHEE FEMALE
00007 89.5 95.4 96.4 SOOKJA FEMALE
00008 85.3 79.5 92.1 JUNGU MALE
```

화면 [S - 3]의 하단 Program Edit 화면 상단에 있는 'COMMAND ==>'란에 'SAVE A(MEMBER1)'을 Typing하고 **ENTER** Key를 누르면 A Library에 방금 사용한 SAS Program이 MEMBER1이라는 이름으로 수록되면서 화면 [S - 4]가 나온다.

• SAVE A(MEMBER1)

- └─ 수록할 SAS Program의 이름으로 사용자가 임의로 정해줌.
- └─ Program이 수록될 Dataset의 이름
- └─ SAS Program을 Dataset에 수록하라는 명령어

[S - 4]

```
COMMAND ==> SAS(R) LOG 13:44
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE-TRIAL SITE (08839001).
NOTE: CPUID VERSION = 00 SERIAL = 080381 MODEL = 4381 .
      1 TSD ALLOC F(A) DA(Z90.PROGRAM.LIBRARY) SHR ;
-----
COMMAND ==> PROGRAM EDITOR
NOTE: MEMBER 'MEMBER1' SAVED.
00001 DATA A;
00002 INPUT ENGLISH MATH SCIENCE NAME $ SEX $ ;
00003 CARDS;
00004 86.7 98.4 87.6 ROKMAM MALE
00005 78.5 89.6 98.6 CHULSU MALE
00006 98.4 76.7 87.4 YONHEE FEMALE
00007 89.5 95.4 96.4 SODKJA FEMALE
00008 85.3 79.5 92.1 JUNGU MALE
```

화면 [S - 4]에서는 작업을 종료하거나 혹은 새로운 SAS Program 을 시작할 수 있다.

SAVE한 Program 을 불러내려면 Program Edit 화면의 'COMMAND ==>' 란에 'COPY A (MEMBER1)' 을 Typing 하고 ENTER Key 를 누르면 MEMBER1의 내용이 나온다.

마. SAS Program 에러 확인 및 수정

화면 [마-1]에는 SAS Program이 작성되어 있다. 이 프로그램에는 평균을 구하라는 명령어 MEANS가 MEAN으로 잘못 작성되어 있으나 사용자는 이를 모른채 [PF3] (or [PF15]) Key 를 눌러 작업을 수행케 한 바 화면 [마-2]가 나왔다.

[마-1]

```

COMMAND ==>
-----
COMMAND ==>
PROGRAM EDITOR

00001 DATA SUNG31 ;
00002 INPUT NAME $ KOR MATH SCI ;
00003 CARDS ;
00004     BOKNAM 86.7 98.4 87.6
00005     CHULSU 78.5 89.6 98.6
00006     SUKJA 98.4 76.7 87.4
00007     JUNGGU 89.5 95.4 96.4
00008     MINHEE 85.3 79.5 92.1
00009     YONGGU 92.4 98.3 91.0
00010
00011 PROC MEAN
00012     RUN
00013
00014
00015
    
```

화면 [마-2]의 상단에 “에러로 인하여 작업이 중단되었다”는 Message가 나와 있는바 이는 SAS Program에 어떤 에러가 있음을 알려주고 있다.

[마-2]

```

COMMAND ==>
-----
NOTE: SAS STOPPED PROCESSING THIS STEP BECAUSE OF ERRORS.
COMMAND ==>
PROGRAM EDITOR

00001
00002
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
    
```

SAS Program의 에러는 SAS LOG 화면에서 보아야 하는데 화면 [마-2]는 너무 좁아서 Message 확인이 곤란하므로 우선 SAS LOG 화면을 확장할 필요가 있다. 이때는 CURSOR를 Program Edit 화면의 하단부로 옮기고 [PF2] (or [PF14]) Key 를 누르면 화면 [마-3]과 같이 SAS LOG 화면이 확장된다.

[마-3]

```

COMMAND ==>                                SAS(0) LOG 13:50
NOTE: SAS STOPPED PROCESSING THIS STEP BECAUSE OF ERRORS.

-----
COMMAND ==>                                PROGRAM EDITOR
00001
00002
00003
00004
00005
    
```

화면 [마-3]에는 SAS LOG의 맨마지막 Message 만 나와 있다. 따라서 Log의 내용을 거꾸로 거슬러 올라가면서 에러 Message를 찾아야 한다. Backward Key는 **PF7** (or **PF19**) Key이므로 [마-3]화면에서 **PF7** (or **PF 19**) Key를 누르면 SAS LOG Message가 화면 [마-4]와 같이 나온다.

[마-4]

```

COMMAND ==>                                SAS(0) LOG 13:56
 89      CARDS
NOTE: DATA SET WORK.SUNG31 HAS 6 OBSERVATIONS AND 4 VARIABLES. 529 OBS/TRK.
 96
 97 PROC MEAN
      183
ERROR 183: THE PROCEDURE NAME IS NOT KNOWN TO THE SYSTEM.
 98      RUN
NOTE: SAS STOPPED PROCESSING THIS STEP BECAUSE OF ERRORS.
-----
COMMAND ==>                                PROGRAM EDITOR
00001
    
```

화면 [마-4]에는 MEAN에 Underline이 그어져 있고 Underline 밑에 183이라는 번호가 나와 있는데 이것은 'ERROR 183'에 에러의 내용이 설명되어 있다는 뜻으로 183 밑에 바로 해당 에러 Message가 나와 있음을 알 수 있다.

만약 [마-4]에 관련 에러 Message 가 나와 있지 않다면 [PF7] (or [PF19]) Key를 눌러서 화면 [마-5]와 같이 SAS LOG Message를 거슬러 올라가야 한다.

[마-5]

```

COMMAND ===>                                SAS LOG 03.01
      85 PROC MEAN
      ---
      103
ERROR 183. THE PROCEDURE NAME IS NOT KNOWN TO THE SYSTEM.
      86 RUN
NOTE: SAS STOPPED PROCESSING THIS STEP BECAUSE OF ERRORS.
      87 DATA SURG31
      88 INPUT NAME $ KGR MATH SCI
      89 CARDS
NOTE: DATA SET WORK.SURG31 HAS 6 OBSERVATIONS AND 4 VARIABLES. 600 BYTES.
      96
      97 PROC MEAN
      ---
      103
ERROR 183. THE PROCEDURE NAME IS NOT KNOWN TO THE SYSTEM.
      98 RUN
-----
COMMAND ===>                                PROGRAM EDIT
00001

```

에러를 확인하여 SAS Program의 에러원인을 알았으면 이제 다시 SAS Program을 불러 수정해야 한다. 이를 위해 [PF3] (or [PF15]) Key를 누르면 Program Edit 화면이 [마-6]과 같이 나온다.

[마-6]

```

COMMAND ===>                                SAS LOG 03.01
NOTE: SAS STOPPED PROCESSING THIS STEP BECAUSE OF ERRORS.
-----
COMMAND ===>                                PROGRAM EDIT
00001
00002
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
00016
00017

```

화면 [마-6]에서 Recall Key 인 **PF4** (or **PF16**) Key 를 누르면 화면 [마-7]과 같이 방금 수행한 SAS Program 이 불러나온다.

[마-7]

```

COMMAND ---,                                SAS:R) LOG: 14 00
NOTE: SAS STOPPED PROCESSING THIS FILE BECAUSE OF LOGORS.
COMMAND ---,                                PROGRAM LOG FOR

00001 DATA WORK1 ;
00002   LABEL NAME * KMR BATH SET ;
00003   LABEL ;
00004   KORHAM 88.7 98.4 87.8
00005   CHULSU 78.5 69.8 98.8
00006   CHUKJA 98.4 76.7 87.4
00007   JUNGDU 89.5 98.4 98.4
00008   HINHEI 87.8 78.5 92.1
00009   YONGDU 82.4 98.3 91.9
00010 ;
00011 PROC MEAN ;
00012   RUN ;
00013 ;
00014 ;
00015 ;
00016 ;
00017 ;

```

화면 [마-7]에서 보면 평균을 구하라는 명령어 MEAN 이 잘못되어 있으므로 이를 MEANS 로 수정하고 다시 **PF3** (or **PF15**) Key 를 눌러 작업을 수행시키면 결과를 볼 수 있다.

바. 작업종료 요령

[바 - 1]

```
COMMAND ==> SAS(R) LOG 16:16
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE-TRIAL SITE (08839001).
NOTE: CPUID VERSION = 00 SERIAL = 080381 MODEL = 4381 .
-----
COMMAND ==> PROGRAM EDITOR
00001
00002
00003
00004
00005
00006
00007
00008
```

화면 [바 - 1]은 Program Edit 화면인데 작업을 종료하기 위해서는 아래 화면 [바 - 2]와 같이 'ENDSAS;' 라는 명령어를 Typing 한다.

[바 - 2]

```
COMMAND ==> SAS(R) LOG 16:16
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE-TRIAL SITE (08839001).
NOTE: CPUID VERSION = 00 SERIAL = 080381 MODEL = 4381 .
-----
COMMAND ==> PROGRAM EDITOR
00001 ENDSAS
00002
00003
00004
00005
00006
00007
00008
```

이때 작업을 종료하는데는 프로그램 Function Key (PF Key)를 사용하는 방법과 Command 를 사용하는 두가지 방법이 있다.

(방법 1) : **PF 3** (or **PF 15**) Key 를 누르면 화면 [바 - 3]이 나온다.

(방법 2) : CURSOR 를 Program Edit 화면의 'COMMAND ==>' 란으로 옮

겨 'SUBMIT'를 Typing 하고 **ENTER** Key를 누르면 화면
[바 - 3]이 나온다.

[바 - 3]

```
----- TSO COMMAND PROCESSOR -----  
ENTER TSO COMMAND OR CLIST BELOW:  
==> GSAS
```

화면 [바 - 3]의 GSAS 를 지우고 아래화면 [바 - 4]와 같이 '=X'를
Typing 하고, **ENTER** Key 를 누르면 화면 [바 - 5]가 나온다.

[바 - 4]

```
----- TSO COMMAND PROCESSOR -----  
ENTER TSO COMMAND OR CLIST BELOW:  
==> =X
```

[바 - 5]

```
SPECIFY DISPOSITION OF LOG DATA SET -----  
COMMAND ==>  
  
LOG OPTIONS FOR THIS SESSION          NO LIST DATA SET THIS SESSION  
-----  
PROCESS OPTION ==>  
SYSOUT CLASS ==> A  
LOCAL PRINTER ID ==>  
  
VALID PROCESS OPTIONS:  
  J - SUBMIT JOB TO PRINT (AND DELETE)      K - KEEP DATA SET (DO NOT PRINT)  
  L - ROUTE TO LOCAL PRINTER (AND DELETE)   D - DELETE DATA SET (DO NOT PRINT)  
  
INSTRUCTIONS:  
  PRESS ENTER KEY TO COMPLETE ISPF TERMINATION.  
  ENTER END COMMAND TO RETURN TO THE PRIMARY OPTION MENU.  
  
JOB STATEMENT INFORMATION: (IF OPTION 'J' SELECTED)  
==> //Z90A      JOB (ACCOUNT), 'NAME'  
==> // *  
==> // *  
==> // *
```

화면 [바 - 5]의 'PROCESS OPTION ==>' 란에 'D'를 Typing 하고,
ENTER Key를 누르면 화면 [바 - 6]이 나온다.

[바 - 6]

```
Z90.SPFLOG1.LIST HAS BEEN DELETED.  
READY
```

화면 [바 - 6]에 'LOGOFF'를 Typing 하고, **ENTER** Key를 누르면 화면
[바 - 7]이 나오면서 작업은 완전히 끝난다.

사. 평균계산

(1) 예 제

다음표는 어떤 학급의 국어·수학·과학점수이다.

A 반 성적표

성명	성별	국어	수학	과학
북 남	남	86.7	98.4	87.6
철 수	남	78.5	89.6	98.6
영 희	여	98.4	76.7	87.4
숙 자	여	89.5	95.4	96.4
정 구	남	85.3	79.5	92.1
민 희	여	85.4	92.1	93.7
용 구	남	92.4	98.3	91.0

A 반의 학과별 평균을 구하라.

(2) 학과별 평균치 계산

① SAS Program작성

A 반 성적표					DATA A ; INPUT NAME \$ SEX \$ KOR MATH SCI ; ----- CARDS ; BOKNAM MALE 86.7 98.4 87.6 CHULSU MALE 78.5 89.6 98.6 YUNGHLE FEMALE 98.4 76.7 87.4 SUKJA FEMALE 89.5 95.4 96.4 JUNGGU MALE 85.3 79.5 92.1 MINHEE FEMALE 85.4 92.1 93.7 YONGGU MALE 92.4 98.3 91.0 ;
성명	성별	국어	수학	과학	
북 남	남	86.7	98.4	87.6	PROC PRINT ; PROC MEANS ; RUN ;
철 수	남	78.5	89.6	98.6	
영 희	여	98.4	76.7	87.4	
숙 자	여	89.5	95.4	96.4	
정 구	남	85.3	79.5	92.1	
민 희	여	85.4	92.1	93.7	
용 구	남	92.4	98.3	91.0	
A 반의 학과별 평균을 구하라.					

- DATASET name → A
- 변수명 정의

변 수	변 수 명	비 고
성 명	NAME	문자 변수이므로 \$표시
성 별	SEX	문자 변수이므로 \$표시
국 어	KOR	숫자 변수
수 학	MATH	숫자 변수
과 학	SCI	숫자 변수

- 자료코딩

CARDS 는 자료가 Card 나 터미날로 입력됨을 의미하며 자료는 INPUT 문에서 정의된 순서에 따라 하나 이상의 빈칸을 두고 코딩 하되 한 Line에 한 Record 를 coding 할 것.

여기서 레코드란 한 사람에 대한 성명, 성별, 국어, 수학 및 과학점수를 의미한다.

- 처리부분 코딩

'PROC PRINT ;'는 입력된 자료의 내용을 그대로 인쇄하라는 명령이며 'PROC MEANS ;'는 평균을 구하라는 명령이다.

'RUN ;'은 상기 PROC 문을 실행하라는 명령이다.

② 작성결과 및 화면설명

평균을 구하기 위하여 작성한 SAS Program은 화면 [사 - 1] 과 같다.

[사 - 1]

```

-----
COMMAND ==>                                PROGRAM EDITOR
00001  DATA A;
00002  INPUT KOREAN MATH SCIENCE NAME $ SEX $
00003  CARDS;
00004  86.7  98.4  87.6  BOKMAM MALE
00005  78.5  89.6  98.6  CHULSU MALE
00006  98.4  76.7  87.4  YONHEE FEMALE
00007  89.5  95.4  96.4  SOKKJA FEMALE
00008  85.3  79.5  92.1  JUNGU MALE
00009  85.4  92.1  93.7  MINHEE FEMALE
00010  92.4  98.3  91.0  YONGGU MALE
00011  ;
00012  PROC PRINT ;
00013  PROC MEANS ;
00014  RUN ;
    
```

DATA 문은 앞의 SAS Program 작성요령에서 설명한 방법에 따라 작성하고 PROC PRINT ; 라는 명령문은 바로전의 DATA Set의 내용을 Print 하라는 명령문이다. PROC MEANS ; 라는 명령문은 바로전의 DATA 문에서 작성된 DATA Set의 모든 숫자 변수에 대해 평균을 계산하라는 명령문이다.

PROC 문에서는 PROC MEANS DATA=A ; 와 같이 특정한 DATA Set을 지정하여 작업을 할 수 있고 바로 직전에 작성된 DATA Set을 사용할 때는 DATA Set의 지정을 생략할 수도 있다.

또한 특정한 변수를 지정할 수도 있는데 PROC MEANS DATA=A ; VAR KOREAN MATH ; 와 같이 작성하면 DATA Set A에 있는 자료중 KOREAN 과 MATH 라는 변수에 대한 평균을 계산하라는 명령문이다.

③ 작업수행

화면 [사-1]과 같이 SAS Program을 작성한 후 **PF3** (or **PF15**) Key를 누르면 첫번째 PROC의 실행결과가 화면 [사-2]와 같이 나온다. 첫번째 PROC문은 PRINT 하라는 명령이었으므로 입력된 자료가 입력된 형태 그대로 나와 있다.

[사-2]

OBS	KOREAN	MATH	SCIENCE	NAME	SEX
1	86.7	98.4	87.6	BOKMAM	MALE
2	78.5	89.6	98.6	CHULSU	MALE
3	98.4	76.7	87.4	YUNHEE	FEMALE
4	89.5	95.4	96.4	SOOKJA	FEMALE
5	85.3	79.5	92.1	JUNGU	MALE
6	85.4	92.1	93.7	MINHEE	FEMALE
7	92.4	98.3	91.0	YONGGU	MALE

SAS Programming (또는 Coding)을 하는데 있어서 중요한 것은 분석에 필요한 DATA를 입력시킨뒤 필요한 PROC을 불러서 분석을 행하기 전에 입력시킨 DATA들이 과연 제대로 SAS에 전해졌는지를 반드시 확인해보아야 한다는 점이다. 이러한 문제는 PROC PRINT를 이용하여 해결할 수가 있다. 화면 [사-2]의 자료는 화면 [사-1]에서 입력한 자료와 같음을 확인할 수 있으므로, 다음 단계의 작업을 수행할 수 있다.

입력자료 확인이 끝나고 다음 작업을 진행시키려면 **PF3** (or **PF15**) Key를 눌러야 한다. 이때는 두번째 PROC의 결과인 평균이 화면 [사-3]과 같이 나온다.

[사 - 3]

① VARIABLE	② N	③ MEAN	④ STANDARD DEVIATION	⑤ MINIMUM VALUE	⑥ MAXIMUM VALUE	⑦ STD ERROR OF MEAN
KOREAN	7	88.02857143	6.26836350	78.5000000	98.4000000	2.36921871
MATH	7	90.00000000	8.75899538	76.7000000	98.4000000	3.31058907
SCIENCE	7	92.40000000	4.20753293	87.4000000	98.6000000	1.59029797

위 화면은 PROC MEANS 문장에 따른 Output 화면이다. 이때는 변수에 대한 평균값뿐만 아니라 ① 변수명 ② 관측치수 ③ 평균 ④ 표준편차 ⑤ 최소값 ⑥ 최대값 ⑦ 표준오차 등을 산출해 준다.

④ 결과표에 표제명 삽입요령

결과표에 Title 을 추가하여 앞으로 여러 PROC 들에 의해 산출되는 결과표와 구별이 용이하게 하려면 TITLE문장을 삽입해야 한다.

결과표에 'AVERAGE SCORE OF CLASS' 라는 표제명을 사용하고 싶을 때는 화면 [사 - 4] 와 같이 PROC 명령어 다음에 TITLE 이란 명령어 및 표제명을 인용부호와 함께 Coding 하면 된다.

[사 - 4]

```

-----
COMMAND ==>                                PROGRAM EDITOR

00001  DATA A;
00002  INPUT KOREAN MATH SCIENCE NAME $ SEX $
00003  CARDS;
00004  86.7  98.4  87.6  BOKNAM MALE
00005  78.5  89.6  98.6  CHULSU MALE
00006  98.4  76.7  87.4  YONHEE FEMALE
00007  89.5  95.4  96.4  SOOKJA FEMALE
00008  85.3  79.5  92.1  JUNGU MALE
00009  85.4  92.1  93.7  MINHEE FEMALE
00010  92.4  98.3  91.0  YONGGU MALE
00011
00013  PROC MEANS ;
00014  TITLE ' AVERAGE SCORE OF CLASS '
00015  RUN ;

```

PF 3 (or PF 15) Key 를 눌러 이 프로그램을 수행시키면 화면 [사-5] 와 같은 결과가 나온다.

[사- 5]

AVERAGE SCORE OF CLASS						
VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	STD ERROR OF MEAN
KOREAN	7	88.02857143	6.26836350	78.5000000	98.4000000	2.36921871
MATH	7	90.00000000	8.75899538	76.7000000	98.4000000	3.31058907
SCIENCE	7	92.40000000	4.20753293	87.4000000	98.6000000	1.59029797

화면 [사- 5] 에는 표제명이 AVERAGE SCORE OF CLASS 란 이름으로 중앙에 나와있다. 따라서 여러 PROC 명령어를 사용하여 결과표가 많이 나온다 해도 구별이 용이함을 알 수 있다.

(3) 남녀별 과목별 평균치 계산

이번에는 남·여별 각과목에 대한 평균점수를 알아보자. 화면 [사- 6] 은 이를 위해 작성한 SAS Program이다.

[사- 6]

```

-----
COMMAND ==>                                PROGRAM EDITOR
00001  DATA A;
00002  INPUT KOREAN MATH SCIENCE NAME $ SEX $ ;
00003  CARDS;
00004  86.7  98.4  87.6  BOKNAM MALE
00005  78.5  89.6  98.6  CHULSU MALE
00006  98.4  76.7  87.4  YONHEE FEMALE
00007  89.5  95.4  96.4  SOOKJA FEMALE
00008  85.3  79.5  92.1  JONGU MALE
00009  85.4  92.1  93.7  MINHEE FEMALE
00010  92.4  98.3  91.0  YONGGU MALE
00011  ;
00012  PROC SORT ; BY SEX ;
00013  PROC PRINT ;
00014  PROC MEANS ;
00015  BY SEX ;
00016  RUN ;

```

남·여 성별에 따른 평균치의 산출을 위해서는 DATA Set 를 성별에 따라 Sort 하여야 하는데 이를 위하여 PROC SORT; BY SEX; 라는 명령어를 사용하였으며 이러한 결과를 보기 위해 PROC PRINT 문을 SORT 문장 뒤에 삽입시켰다.

그리고 PROC MEANS 문장에서 BY SEX ; 라는 문장을 뒀으로써 성별로 산출된 숫자변수(과목)들의 평균값을 얻을 수 있다.

화면 [사 - 6]에서 **PF 3** (or **PF 15**) Key 를 누르면 남녀별로 SORT 된 자료가 화면 [사 - 8]과 같이 나온다.

[사 - 8]

OBS	KOREAN	MATH	SCIENCE	NAME	SEX
1	98.4	76.7	87.4	YONHEE	FEMALE
2	89.5	95.4	96.4	SOOKJA	FEMALE
3	85.4	92.1	93.7	MINHEE	FEMALE
4	86.7	98.4	87.6	ROKHAM	MALE
5	78.5	89.6	98.6	CHULSU	MALE
6	85.3	79.5	92.1	JUNGU	MALE
7	92.4	98.3	91.0	YONGGU	MALE

SEX 란을 보면 입력자료가 남녀별로 SORT 되어 있음을 알 수 있다.

다시 **PF 3** (or **PF 15**) Key 를 누르면 남녀별 과목별 평균이 화면 [사 - 9]와 같이 나온다.

[사 - 9]

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM Value	MAXIMUM Value	Std. Deviation of Values
-----SEX=FEMALE-----						
KOREAN	3	91.10000000	6.64605146	85.4000000	98.4000000	3.03709960
MATH	3	88.06666667	9.98114890	76.7000000	95.4000000	5.76261500
SCIENCE	3	92.50000000	4.61844130	87.4000000	96.4000000	2.66645833
-----SEX=MALE-----						
KOREAN	4	85.72500000	5.71219456	78.5000000	92.4000000	2.85609728
MATH	4	91.45000000	8.97125038	79.5000000	98.4000000	4.48562519
SCIENCE	4	92.32500000	4.60099627	87.6000000	98.6000000	2.30049813

상단에는 여자에 대한 PROC MEANS 의 결과가 그 아래에는 남자에 대한 PROC MEANS 의 결과가 나와있다.

아. 상관분석

(1) 예 제

다음 통계표는 전도시 전가구의 수입, 음식비 지출, 주거비 지출에 관한 자료이다.

도시가계의 수입 및 지출

연 도	수 입	지 출	
		음 식 비	주 거 비
75	78.9	28.4	1.6
76	111.2	34.9	2.3
77	143.2	40.1	3.1
78	198.4	51.3	4.2

연 도	수 입	지 출	
		음 식 비	주 거 비
79	281.2	63.4	6.5
80	344.4	77.4	8.1
81	408.1	92.9	8.9
82	460.8	101.6	10.8
83	528.3	106.8	13.6

상기 자료를 이용하여 수입, 음식비, 주거비 상호간의 상관관계를 계산하라.

(2) SAS Program 작성

상관관계를 알아보기 위하여 작성된 프로그램은 화면 [아-1] 과 같다. 상관관계를 구하라는 명령어는 CORR 이다.

[아-1]

```

-----
COMMAND ==>                                PROGRAM EDITOR

00001  DATA A,
00002  INPUT SUIP FOOD HOUSING
00003  CARDS;
00004  78.9 28.4 1.6
00005  111.2 34.9 2.3
00006  143.2 40.1 3.1
00007  198.4 51.3 4.2
00008  281.2 63.4 6.5
00009  344.4 77.4 8.1
00010  408.1 92.9 8.9
00011  460.8 101.6 10.8
00012  528.3 106.8 13.6
00013  ;
00014  PROC CORR ;
00015  TITLE 'CORRELATION OF HOUSEHOLD DATA'
00016  RUN ;

```

PF 3 (or PF 15) Key 를 눌러 작업을 수행시키면 화면 [아-2] 와 같은 결과가 나온다.

CORRELATION OF HOUSEHOLD DATA						
VARIABLE	N	MEAN	STD DEV	SUM	MINIMUM	MAXIMUM
SUIP	9	283.833333	161.661351	2554.50000	78.9000000	528.300000
FOOD	9	66.311111	29.705490	596.80000	28.4000000	106.800000
HOUSING	9	6.566667	4.118252	59.10000	1.6000000	13.600000

PEARSON CORRELATION COEFFICIENTS / PROB > |R| UNDER H0:RHO=0 / N = 9

	SUIP	FOOD	HOUSING
SUIP	1.00000 0.0000	0.99659 0.0001	0.99429 0.0001
FOOD	0.99659 0.0001	1.00000 0.0000	0.98306 0.0001
HOUSING	0.99429 0.0001	0.98306 0.0001	1.00000 0.0000

PROC CORR 에 의한 결과로서 우선 각 변수에 대한 단순통계량 (A) 이 산출되는데 이는 각 변수의 특성치를 파악하는데 도움을 준다. 다음은 분석에 사용된 상관계수의 종류를 알 수 있는데 이는 B에 Pearson Correlation이라는 용어으로써 알 수 있다. (PROC CORR 문에서 상관계수의 종류인 Pearson, Kendall, Spearman 중에서 지정이 없으면 자동으로 Pearson Correlation이 산출됨). C를 보면 상관계수에 대한 설명이 나와 있는데 이는 상관계수의 유의확률로서 귀무가설 $H_0: \rho=0$ 하에서의 신뢰성이라 말할 수 있다. 이값이 낮을수록 대응된 상관계수의 안정성이 있다고 할 수 있다.

예를들어 D를 보면 수입(SUIP)과 주거비지출(HOUSING)과의 상관계수 $r = 0.99429$ 이고 이의 유의확률은 0.0001이다.

(참고) E를 흔히 Correlation Matrix라 부른다. 이것을 보는 법은 우선 변수가 x, y, z 의 3개가 있다고 하면,

Correlation Matrix는
$$\begin{pmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{pmatrix}$$
 과 같고, 대각요소는 자기와

자기의 상관이므로 ①이 되며, 대각요소(diagonal element)를 제외한 대각요소의 오른쪽 윗부분(upper diagonal matrix; 상삼각행렬), 또는 왼쪽 아래부분(Lower diagonal matrix; 하삼각행렬) 한쪽만 보면 임의의 두 변수의 상관관계를 알 수 있다. 이러한 상관관계 Matrix는 좀더 고등적인 다변량기법인 요인분석(Factor Analysis), 경로분석(Path Analysis) 등에서 중요하게 사용된다.

자. 모든 기술통계치 계산

평균, 표준편차 등의 단순통계량외에 단변량통계에 대한 모든 정보를 알고자 할때 PROC UNIVARIATE를 쓰면 편리하다.

[자 - 1]

```

-----
COMMAND ==>                                PROGRAM EDITOR
00001 DATA STATEPOP;
00002   INPUT STATE $ POP @@;
00003   LABEL POP = '1970 CENSUS POPULATION IN MILLIONS';
00004   CARDS;
00005 ALA 3.44 ALASKA 0.30 ARIZ 1.77 ARK 1.92 CALIF 19.95
00006 COLO 2.21 CONN 3.03 DEL 0.55 FLA 6.79 GA 4.59
00007 HAW 0.77 IDAHO 0.71 ILL 11.01 IND 5.19 IOWA 2.83
00008 KAN 2.25 KY 3.22 LA 3.64 ME 0.99 MD 3.92
00009 MASS 5.69 MICH 8.88 MINN 3.81 MISS 2.22 MO 4.68
00010 MONT 0.69 NEB 1.48 NEV 0.49 NH 0.74 NJ 7.17
00011 NM 1.02 NY 10.24 NC 5.08 ND 0.62 OHIO 10.65
00012 OKLA 2.56 ORE 2.09 PA 11.79 RI 0.95 SC 2.59
00013 SD 0.67 TENN 3.92 TEXAS 11.2 UTAH 1.06 VT 0.44
00014 VA 4.65 WASH 3.41 W.VA 1.74 WIS 4.42 WYO 0.33
00015 YY ZZ
00016 PROC UNIVARIATE;
00017   TITLE 'OUTPUT FROM UNIVARIATE PROCEDURE';
00018   RUN ;
    
```

화면 [자 - 1]은 미국의 각주별 센서스인구를 이용하여 기술통계치를 계산하고자 작성한 SAS Program이다.

INPUT 문에서 STATE 변수는 문자변수로, POP 변수는 숫자변수로 읽혀들이고 POP 이란 변수에는 설명을 위하여 LABEL 문을 이용하여 '1970 CENSUS POPULATION IN MILLION S'라는 LABEL 를 주었다. 그리고 PROC UNIVARIATE 를 부르고 TITLE 문을 주었다.

INPUT 문의 @@부호는 STATE 와 POP에 대한 자료가 한 Line에 두 번이상 반복되어 입력되고 있을때 사용한다.

YY.ZZ. 는 YY주와 ZZ주에 대한 인구를 알지 못하여 Missing Value로 처리한 것이다.

화면 [자 - 1]에서 PF 3 (or PF 15) Key 를 누르면 작업이 수행되고 결과 화면 [자 - 2]가 나온다.

화면 [자-2]에는 결과로서 N (관측치수) MEAN, STD DEV (표준편차), SKEWNESS (왜도), USS (Uncorrected Sum of Square), CV (Coefficient of Variation), SUM WGTS (총가중치), SUM (합), VARIANCE (분산), KURTOSIS (첨도), CSS (Corrected Sum of Square), STD MEAN (표준오차), NUM >= 0 (0이 아닌 관측치수), T ; MEAN=0, PROB > | T | (MEAN = 0 이라는 귀무가설의 검정) 등이 산출되어 있다.

[자-2]

OUTPUT FROM UNIVARIATE PROCEDURE					
UNIVARIATE					
VARIABLE=POP	1970 CENSUS POPULATION IN MILLIONS				
MOMENTS					
N	50	SUM WGTS	50		
MEAN	4.0472	SUM	202.36		
STD DEV	4.32932	VARIANCE	18.743		
SKEWNESS	2.05522	KURTOSIS	4.54561		
USS	1737.4	CSS	918.407		
CV	106.971	STD MEAN	0.612258		
T; MEAN=0	6.61028	PROB > T	0.0001		
SGN RANK	637.5	PROB > S	0.0001		
NUM >= 0	50				

다시 **PF 3** (or **PF 15**) Key 를 누르면 화면 [자-3]이 나온다.

[자-3]

OUTPUT FROM UNIVARIATE PROCEDURE						
UNIVARIATE						
VARIABLE=POP	1970 CENSUS POPULATION IN MILLIONS					
QUANTILES(DEF=4)			EXTREMES			
100% MAX	19.95	99%	19.95		LOWEST	HIGHEST
75% Q3	4.78	95%	14.6925		0.3	19.95
50% MED	2.71	90%	10.974		0.33	19.95
25% Q1	0.98	10%	0.557		0.43	19.95
0% MIN	0.3	5%	0.3905		0.45	19.95
RANGE	19.65	1%	0.3		0.5	19.95
Q3-Q1	3.8					
MODE	3.92					

이결과 화면에서는 100 분위수 (Percentile), 4 분위수 (Quantile), RANGE (범위), MODE (최빈수), EXTREMES (최상, 하 극단치 5개씩)을 보여준다.

차. 뜻수분포, PLOTTING, 정규분포성 검증

어떤 자료에 대하여 뜻수분포표 작성, PLOT 작성 또는 정규분포성 여부 검증 등을 하기 위해서는 PROC UNIVARIATE 에 ① FREQ (뜻수분포표), ② PLOT (Plotting), ③ NORMAL (정규분포함수) 을 OPTION 으로 사용함으로써 가능하다.

[차 - 1]

```

-----
COMMAND ---->                                PROGRAM
00001 DATA STATEPOP.
00002   INPUT STATE * POP @@.
00003   LABEL POP = '1970 CENSUS POPULATION IN MILLIONS'.
00004   CARDS;
00005 ALA 3.44 ALASKA 0.30 ARIZ 1.77 ARK 1.92 CALIF 19.95
00006 COLO 2.21 CONN 3.03 DEL 0.53 FLA 6.79 GA 4.59
00007 HAW 0.77 IDAHO 0.71 ILL 11.01 IND 5.19 IOWA 2.83
00008 KAN 2.25 KY 3.22 LA 3.64 ME 0.99 MD 3.92
00009 MASS 5.69 MICH 8.88 MINN 3.81 MISS 2.22 MO 4.68
00010 MONT 0.69 NEB 1.48 NEV 0.49 NH 0.74 NJ 7.17
00011 NM 1.02 NY 18.24 NC 5.08 ND 0.62 OHIO 10.65
00012 OKLA 2.56 ORE 2.09 PA 11.79 RI 0.95 SC 2.59
00013 SD 0.67 TENN 3.92 TEXAS 11.2 UTAH 1.06 VT 0.44
00014 VA 4.65 WASH 3.41 W.VA 1.74 WIS 4.42 WYO 0.33
00015 YY ZZ
00016 PROC UNIVARIATE FREQ PLOT NORMAL;
00017   TITLE 'OUTPUT FROM UNIVARIATE PROCEDURE';
00018   VAR POP;
00019   ID STATE;
00020   RUN;

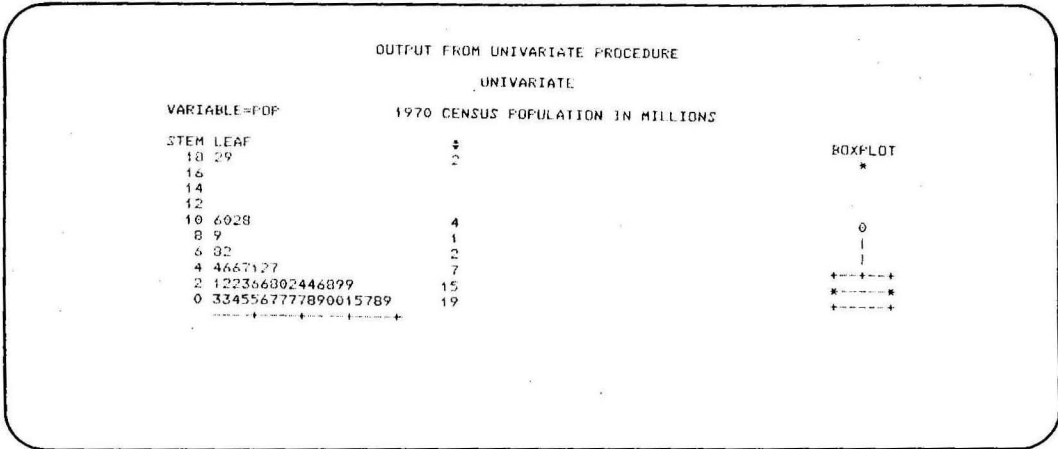
```

화면 [차 - 1]은 PROC UNIVARIATE 문에 OPTION 으로서 FREQ PLOT NORMAL 을 지정하였고 VAR 은 분석에 사용할 변수이름을 명시하는 명령어이며 ID 는 관측치들을 식별하기 위하여 사용할 변수를 지정하는 명령어로 여기서는 STATE 라는 변수를 사용하였다.

PF 3 (or **PF 15**) Key 를 눌러 작업을 수행시키면 화면 [차 - 2]가 나온다.

다시 **PF 3** (or **PF 15**) Key 를 누르면 화면 [차 - 4] 가 나온다.

[차 - 4]



화면 [차 - 4] 는 PLOT Option에 의해 그려지는 나머지 2개의 Plot인데 왼쪽에 있는 것이 STEM & LEAF Plot 이고 오른쪽에 있는 것이 BOXPLOT 이다.

STEM & LEAF Plot 은 말 그대로 줄기와 잎사귀란 의미로 줄기는 자료에 대한 계급치를 의미하고 잎사귀는 그 계급에 속하는 자료를 나타내 준다. 맨오른쪽은 그 계급에 속해 있는 dot수를 나타낸다. 자료수가 50 이 넘을 때는 수평막대도표가 산출된다.

다음 BOXPLOT 는 맨밑의 선이 25th 백분위점을 나타내고 2번째 선은 Median, 그리고 맨위의 선은 표본평균의 위치를 나타내준다. 그리고 수직선 위의 0 은 1.5 Inter Quantile ($Q_3 - Q_1$)보다 멀리 위치해 있는 자료를 의미하고 *는 그보다 더 떨어져 있는 자료라는 것을 의미한다.

이 두가지의 자료기술방법은 자료분석의 대가인 Tukey 박사에 의해 고안된 것이다. 상세한 내용에 대해서는 Tukey 의 ' 데이터 분석법 ' 을 참고하기 바란다.

다시 **PF 3** (or **PF 15**) Key 를 누르면 화면 [차 - 5] 가 나온다.

[차 - 5]

```

OUTPUT FROM UNIVARIATE PROCEDURE
UNIVARIATE
VARIABLE=POP      1970 CENSUS POPULATION IN MILLIONS

QUANTILES(DEF=4)
100% MAX      19.95      99%      19.95
75% Q3        4.78      95%      14.6925
50% MED        2.71      90%      10.974
25% Q1         0.98      10%      0.557
0% MIN         0.3       5%       0.3905
              1%       0.3
RANGE          19.65
Q3-Q1           3.8
MODE            3.92
    
```

PF 3 (or PF 15) Key 를 누르면 화면 [차 - 6] 이 나온다.

[차 - 6]

```

OUTPUT FROM UNIVARIATE PROCEDURE
UNIVARIATE
VARIABLE=POP      1970 CENSUS POPULATION IN MILLIONS

EXTREMES
LOWEST  ID      HIGHEST  ID
0.3(ALASKA )    11.01(ILL )
0.33(WYD )      11.2(TEXAS )
0.44(VT )       11.79(PA )
0.49(NEV )      18.24(NY )
0.55(DEL )      19.95(CALIF )

MISSING VALUE      -
COUNT              2
% COUNT/NOBS      3.85
    
```

화면 [차 - 6]에는 ID 변수를 명시해줌으로써 얻은 효과가 반영되어 있다. 즉, 양극단치들에 대한 값과 그값을 식별할 수 있는 주이름이 나오게 된다. 그리고 누락치 (Missing Value) 대 관측치수에 대한 비율이 나와 있다.

다시 **PF 3** (or **PF 15**) Key 를 누르면 화면 [차 - 7] 이 나온다.
 [차 - 7]

```

    OUTPUT FROM UNIVARIATE PROCEDURE
              UNIVARIATE
VARIABLE=POP      1970 CENSUS POPULATION IN MILLIONS
              FREQUENCY TABLE
    
```

PERCENTS				PERCENTS			
VALUE	COUNT	CELL	CUM	VALUE	COUNT	CELL	CUM
0.3	1	2.0	2.0	1.06	1	2.0	30.0
0.33	1	2.0	4.0	1.48	1	2.0	32.0
0.44	1	2.0	6.0	1.74	1	2.0	34.0
0.49	1	2.0	8.0	1.77	1	2.0	36.0
0.55	1	2.0	10.0	1.92	1	2.0	38.0
0.62	1	2.0	12.0	2.09	1	2.0	40.0
0.67	1	2.0	14.0	2.21	1	2.0	42.0
0.69	1	2.0	16.0	2.22	1	2.0	44.0
0.71	1	2.0	18.0	2.25	1	2.0	46.0
0.74	1	2.0	20.0	2.56	1	2.0	48.0
0.77	1	2.0	22.0	2.59	1	2.0	50.0
0.95	1	2.0	24.0	2.83	1	2.0	52.0
0.99	1	2.0	26.0	3.03	1	2.0	54.0
1.02	1	2.0	28.0	3.22	1	2.0	56.0

화면 [차 - 7] 은 OPTION 변수인 FREQ 에 의해서 산출된 dots 분포표이다.
 이 화면에는 dots 분포표가 다 나오지 못하므로 **PF 3** (or **PF 15**)
 Key 를 눌러 화면 [차 - 8] 을 찾아야 한다.

[차 - 8]

```

    OUTPUT FROM UNIVARIATE PROCEDURE
              UNIVARIATE
VARIABLE=POP      1970 CENSUS POPULATION IN MILLIONS
              FREQUENCY TABLE (CONT.)
    
```

PERCENTS				PERCENTS			
VALUE	COUNT	CELL	CUM	VALUE	COUNT	CELL	CUM
3.41	1	2.0	58.0	5.69	1	2.0	60.0
3.44	1	2.0	60.0	6.79	1	2.0	62.0
3.64	1	2.0	62.0	7.17	1	2.0	64.0
3.81	1	2.0	64.0	8.88	1	2.0	66.0
3.92	2	4.0	68.0	10.65	1	2.0	68.0
4.42	1	2.0	70.0	11.01	1	2.0	70.0
4.59	1	2.0	72.0	11.2	1	2.0	72.0
4.65	1	2.0	74.0	11.79	1	2.0	74.0
4.68	1	2.0	76.0	18.24	1	2.0	76.0
5.08	1	2.0	78.0	19.95	1	2.0	78.0
5.19	1	2.0	80.0				

화면 [차 - 8] 에는 전 화면에서 다 나오지 못한 dots 분포표의 나머지 부분이 나와 있다.

이렇게 하여 PROC UNIVARIATE 에 의해 산출된 모든 결과가 나왔다.
 PROC UNIVARIATE 는 단변량 변수에 대한 거의 모든 정보를 제공해 주는 강력한 기술 통계 PROCEDURE 이다.

카. Histogram (막대그래프)

화면 [카-1]은 어느 회사의 연도별 판매액과 비용으로 연도별 순이익을 산출하고 순이익에 관한 Histogram을 그리기 위한 SAS Program이다. Histogram을 그리기 위한 SAS 명령은 PROC CHART이다.

[카-1]

```

COMMAND ==>
SAS(R) LOG 11 16
-----
COMMAND ==>
PROGRAM EDITOR
00001 DATA A;
00002 INPUT YEAR SALES COST;
00003 PROFIT=SALES-COST;
00004 CARDS;
00005 1977 2132 1021
00006 1977 5643 3521
00007 1978 19823 12928
00008 1979 16982 14002
00009 1979 9967 5761
00010 1980 18432 14590
00011 ;
00012 PROC CHART ;
00013 TITLE 'PROC CHART REPORT ON PROFIT';
00014 VBAR YEAR / SUMVAR=PROFIT DISCRETE SYMBOL='XOA';
00015 RUN ;
00016
00017
00018
00019
    
```

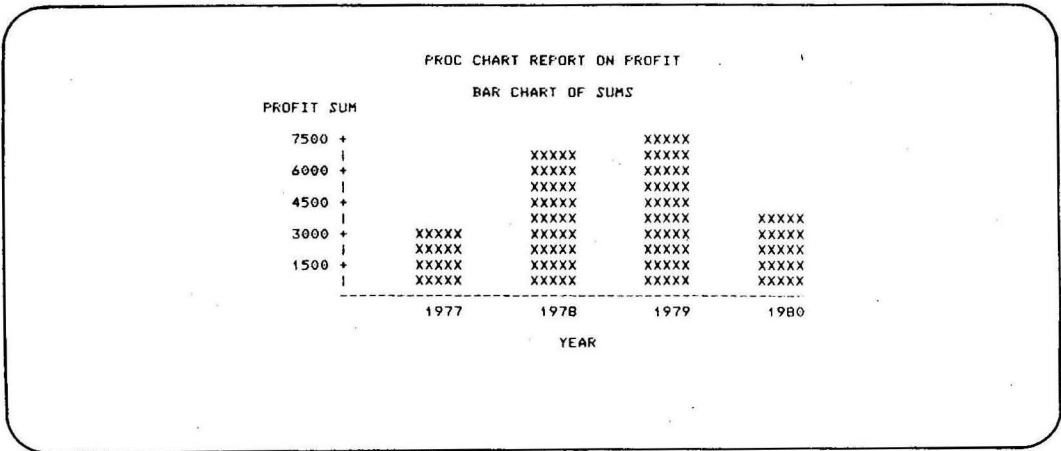
상기 프로그램에서는 연도별 판매액과 비용에 대한 DATA를 입력시키고 순이익은 $PROFIT = SALES - COST$ 문장으로 계산하여 SAS DATA Set을 만들었다. 그리고 연도별 순이익에 대한 막대도표를 그리기 위하여 PROC CHART라는 명령어를 사용했다. VBAR는 Vertical Chart라는 의미이고 YEAR는 연도별 도표를 그리라는 뜻이다.

⊙ (Slash) 다음의 OPTION에서 SUMVAR=PROFIT는 연도별 도표를 그리되 같은 연도의 자료를 PROFIT 변수로 합제해서 그리라는 것이며 DISCRETE는 연도(YEAR)가 연속변수가 아니고 명목변수라는 뜻이다.

그리고 SYMBOL='XOA'는 Plotting을 할때 막대도표의 무늬를 알려주는 것이다.

[PF 3] (or [PF 15]) Key 를 눌러 작업을 수행시키면 그 결과가 화면 [카- 2]와 같이 나온다.

[카- 2]



타. 추세선 작성

시계열 자료에 대한 시간적 변동추이를 용이하게 파악하기 위하여 추세선을 많이 사용한다. 이때 사용하는 명령어는 PROC PLOT 이다.

[타- 1]

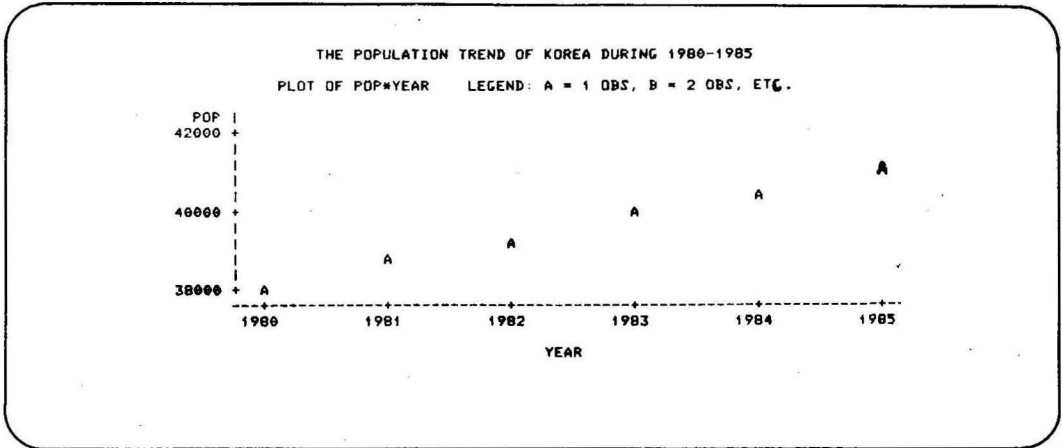
```

COMMAND ==>
SAS(R) LOG 11 29
-----
COMMAND ==>
PROGRAM EDITOR
00001 DATA STATEPOP,
00002 INPUT YEAR $ POP @@ ;
00003 CARDS;
00004 1980 38124 1981 38723
00005 1982 39331 1983 39951
00006 1984 40578 1985 41209
00007 ;
00008 PROC PLOT ;
00009 PLOT POP*YEAR;
00010 TITLE ' THE POPULATION TREND OF KOREA DURING 1980-1985 ' ;
00011 RUN ;

```

화면 [타 - 1] 은 1980 년부터 1985 년까지의 우리나라 인구추계자료를 가지고 추세선을 그리기 위하여 작성한 SAS Program이다.

[타 - 2]



화면에서 PLOT POP*YEAR 은 종축에 인구 (POP) 를 횡축에 연도 (YEAR) 를 그리도록 명시한 것이다.

(or) Key 를 누르면 화면 [타 - 2] 와 같이 그 결과가 나온다.

파. Table 및 Histogram 작성

일반적으로 자료중에는 여러가지 형태의 Table 작성이 요구되는 경우가 많다. 이때 PROC TABULATE 를 이용하면 편리하다.

<참고> 이 PROC 의 내용은 익히 알고 있는 TPL (미국 노동부에서 개발한 Table 작성패키지) 의 내용을 그대로 이용 한 것이다.
또한 MVS 하의 SAM file, VSAM file 의 Access 가, 가능하다.

[파 - 1]

```

COMMAND ==>
00001 DATA UNIV;
00002 INPUT KYEYUL $ GRADE ;
00003 RETAIN INWON 1 ;
00004 CARDS ;
00005 KYO_YUK 3
00006 KYO_YUK 3
00007 SA_WHOI 3
00008 KYO_YUK 4
00009 IN_MOON 3
00010 IN_MOON 2
00011 SA_WHOI 3
00012 IN_MOON 3
00013 KYO_YUK 4
00014 SA_WHOI 4
00015 IN_MOON 1
00016 ;
00017 PROC TABULATE ;
00018 CLASS KYEYUL GRADE ;
00019 VAR INWON ;
00020 TABLE KYEYUL*GRADE, INWON ;
00021 TITLE 'THE SURVEY OF LEISURE TIME ACTIVITY OF DONGDEOK WOMENS U';
00022 PROC CHART ;
00023 VBAR KYEYUL ;
00024 PROC CHART ;
00025 VBAR GRADE/DISCRETE ;
00026 RUN ;

```

화면 [파 - 1]은 대학생들의 전공계열 및 학년자료를 이용, 계열별, 학년별 인원수 Table 과 막대그래프를 작성하기 위한 SAS Program이다.

RETAIN 문에서 매 관측치마다 INWON=1 로 Setting 시켰다.

PROC TABULATE 에서 CLASS 이후에 나오는 변수는 명목변수가 된다. 그리고 VAR 이후에 나오는 변수는 실제 분석되는 자료의 이름이 된다.

TABLE 문에서는 TABLE 의 모양을 결정하게 된다.

PROC CHART 문에서 VBAR KYEYUL 은 계열별 인원수에 대한, VBAR GRADE/DISCRETE 는 학년별 인원수에 대한 막대그래프를 그리라는 명령이다. 여기서 /DISCRETE 는 GRADE 라는 변수가 명목변수임을 알려주고 있다.

[PF 3] (or [PF 15]) Key 를 누르면 첫번째 PROC 명령에서 정의한 Table 이 화면 [파 - 2] 와 같이 나온다.

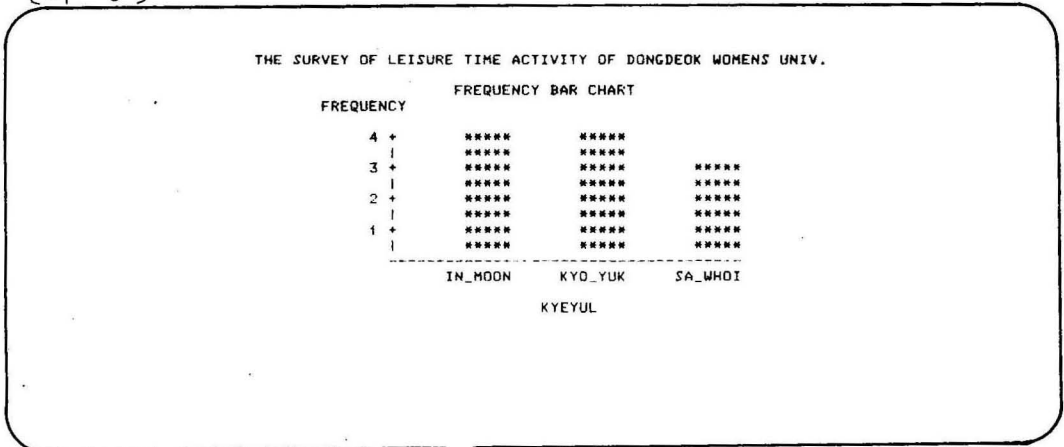
[파 - 2]

THE SURVEY OF LEISURE TIME ACTIVITY OF DONGDEOK WOMENS UNIV.

		INWON
		SUM
KYEUUL	GRADE	
IN_MOON	1	1.00
	2	1.00
	3	2.00
KYO_YUK	GRADE	
	3	2.00
KYO_YUK	4	2.00
SA_WHOI	GRADE	
	3	2.00
	4	1.00

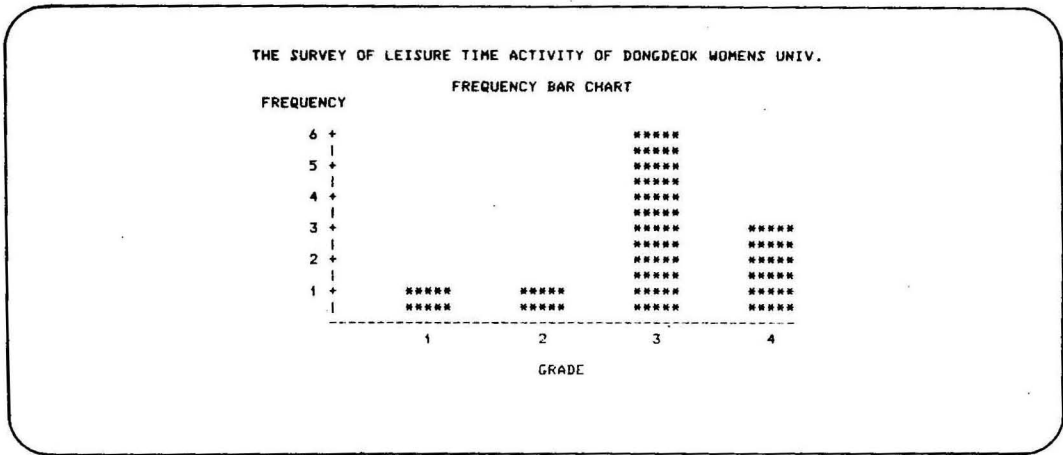
다시 [PF 3] (or [PF 15]) Key 를 누르면 두번째 PROC 명령문에 의하여 작성된 계열별 인원수에 대한 막대그래프가 화면 [파 - 3] 과 같이 나온다.

[파 - 3]



다시 PF 3 (or PF 15) Key 를 누르면 세번째 PROC 명령으로 작성된 학년별 인원수에 대한 막대그래프가 화면 [파 - 4]와 같이 나온다.

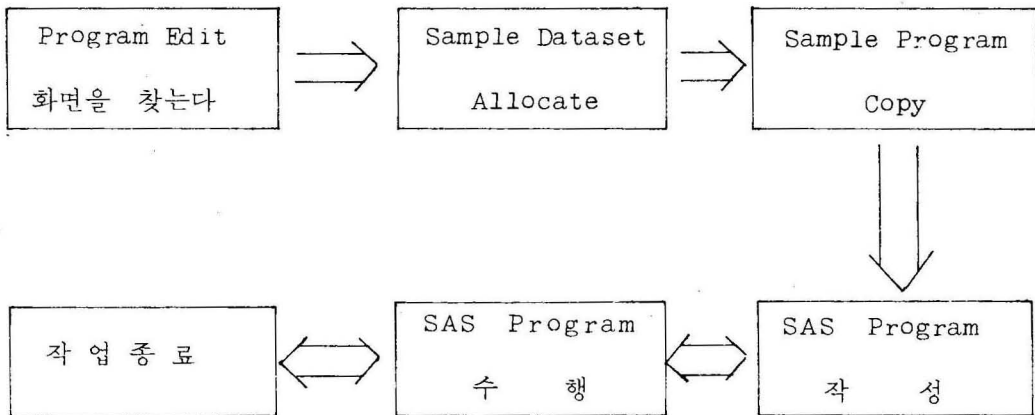
[파 - 4]



2. DB 를 이용한 SAS Program 작성

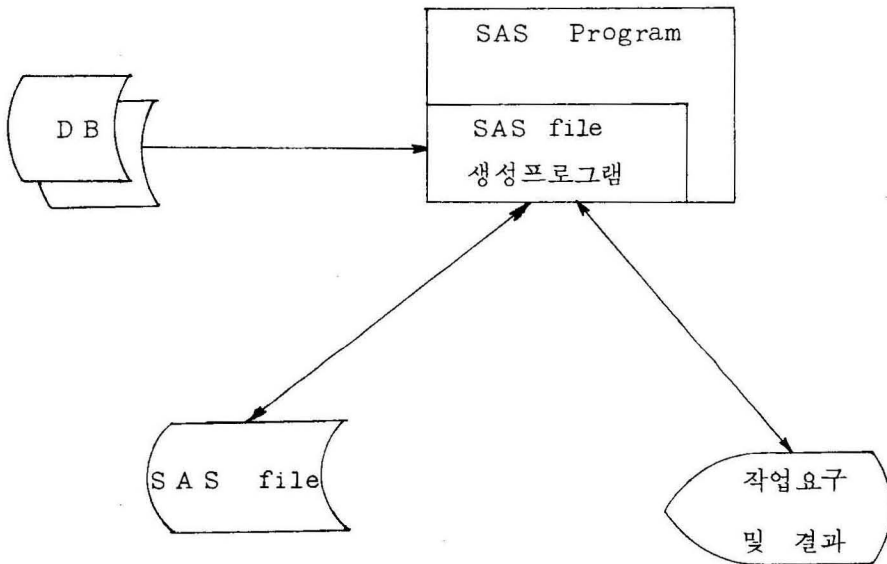
가. 작업요령

우선 Program Edit 화면을 1항 너의 요령에 따라 찾아야 한다. DB자료를 이용하는 방법은 일반적인 SAS 이용법과 차이가 있으므로 이에 대한 몇 가지 Sample Program 을 작성하여 DB·SAS 라는 Data set 에 수록, 필요할 때마다 복사해서 사용할 수 있도록 하였다. 물론 사용자마다 이용자료와 처리내용이 다르기 때문에 Sample Program 을 그때마다 수정해야 할 것이다. 작업단계를 정리하면 다음과 같다.



나. DB 와 SAS 연결

DB에 수록된 주요통계정보를 SAS 패키지에서 직접 읽어 분석에 사용하면 자료를 일일이 Typing 해야하는 번거러움을 없앨 수가 있다. 당국의 시스템에서는 DB에서 자료를 읽어 SAS file 을 만들 수 있는 프로그램을 준비하여 이용자들이 언제라도 쉽게 사용할 수 있도록 하였다. 작업요령에서 언급한 Sample Program 들도 역시 이 프로그램을 사용한 것이다.



다. Sample Dataset Allocate

SAS 를 이용하여 Interactive 작업을 하기 위해서는 Sample Dataset 를 SAS 와 연결시켜 주어야 하는데 이때 사용하는 명령어는 ALLOCATE이다. 화면 [다1]의 Program Edit 화면은 그 방법을 예시해 주고 있다.

[다1]

```

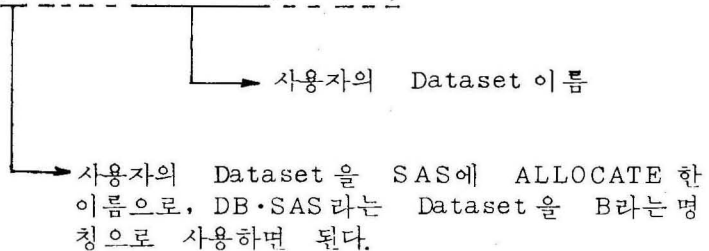
Command ===)                                     SAS(r) Log 14 51
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE-TRIAL SITE (02122001).
NOTE: CPUID VERSION = 00 SERIAL = 080381 MODEL = 4381 .
-----
Command ===)                                     Program Editor
00001 TSO ALLOC F(A) DA(Z90.PROGRAM.LIBRARY) SHR ,
00002 TSO ALLOC F(B) DA(DB.SAS) SHR ,
00003
00004
00005
00006
00007
00008
  
```

화면 [다1]은 이용자들이 편리하게 이용할 수 있도록 여러가지 SAS Sample Program 을 모아놓은 'DB·SAS' 라는 Sample Dataset 을 B라는 이름으로 Allocate 하겠다는 뜻이다.

- ALLOCATE 문 작성요령

```

-----
TSO ALLOC F(B) DA(DB·SAS) SHR ;
-----
  
```



PF3

(or

PF 15

) key 를 누르면 화면 [다 2] 가 나온다.

[다 2]

```

COMMAND ===>                                SAS(R) LOG 14:54
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE TRIAL SITE (02122001).
NOTE: CPUID VERSION = 00 SERIAL = 080381 MODEL = 4381 .
      1 TSD ALLOC F(A) DA(290.PROGRAM.LIBRARY) SHR ,
      2 TSD ALLOC F(B) DA(DB.SAS) SHR ;
-----
Command ===>                                Program Editor
00001
00002
00003
00004
00005
00006
00007
00008

```

라. Sample Program Copy

DB · SAS 라는 Dataset 에는 다음과 같은 여섯가지 Sample 프로그램 이 수록되어 있다.

- EXEM 1 → 작성된 프로그램의 코딩내용만을 출력해 주는 Sample 프로그램명
- EXEM 2 → DB자료를 읽어서 출력해 주는 Sample 프로그램명
- EXEM 3 → DB자료를 읽어 새로운 Dataset 을 만드는 Sample 프로그램명
- EXEM 4 → 회귀분석 Sample 프로그램명
- EXEM 5 → 시계열자료 PLOTTING Sample Program 명
- EXEM 6 → Forecasting Sample Program 명

이용자들은 상기 6 가지 Sample 프로그램을 수행해 보고 이들을 변형하여 원하는 Program 을 작성한다면 훨씬 용이할 것이다.

(1) Source Program 을 출력하는 Sample Program (EXEM1)

작성된 Program 의 코딩내용을 확인하기 위해서는 코딩내용을 그대로 출력해 주는 Sample Program을 복사하여 이용하면 편리하다.

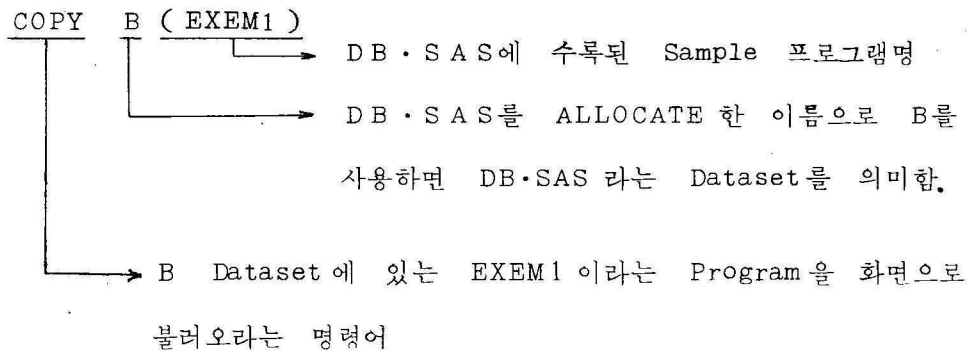
[라 1]

```

Command ==>                                     SAS(r) Log 14:52
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE TRIAL SITE (02122001).
NOTE: CPUID   VERSION = 00   SERIAL = 080381   MODEL = 4381
      1 TSO ALLOC F(A) DA(Z90.PROGRAM.LIBRARY) SHR ;
      2 TSO ALLOC F(B) DA(DB.SAS) SHR ;
-----
COMMAND ==> COPY B(EXEM1)                                PROGRAM EDITOR
00001
00002
00003
00004
00005
00006
00007
00008
    
```

화면 [라 1]의 하단 Program Edit 화면에서 'COMMAND ==>' 란에 'COPY B(EXEM1)' 를 Typing 하고, **ENTER** Key 를 누르면 화면 [라 2]와 같이 Sample 프로그램이 불러어 나온다.

- COPY 명령어 작성요령



[라 2]

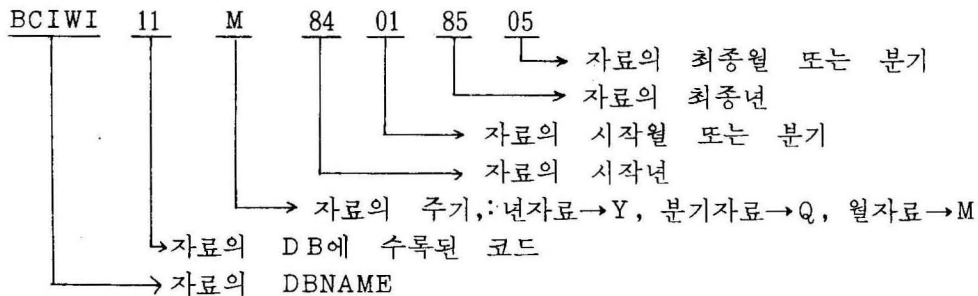
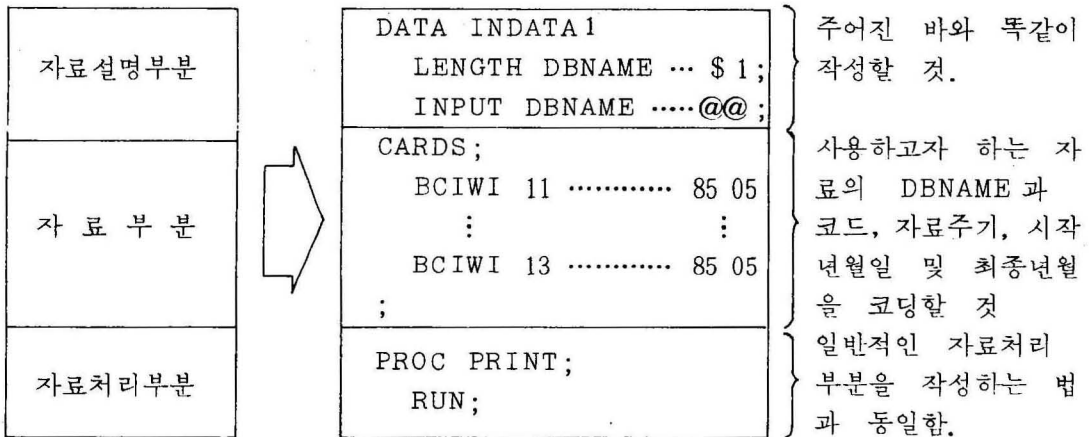
```

COMMAND ==>
SAS(R) LOG 14:56
NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.08 AT SAS INSTITUTE-TRIAL SITE (02122001).
NOTE: CPUID VERSTON = 00 SERIAL = 080381 MODEL = 4381 .
      1 TSO ALLOC F(A) DA(Z90.PROGRAM.LIBRARY) SHR ,
      2 TSO ALLOC F(B) DA(DB.SAS) SHR ;
-----
COMMAND ==>
PROGRAM EDITOR
00001 DATA INDATA1;
00002 LENGTH DBNAME $5 DBKEY $9 MQY $1;
00003 INPUT DBNAME $ DBKEY $ MQY $ FROMY FROMM TOY TOM @@;
00004 CARDS;
00005 BCIWI 11 M 84 01 85 05
00006 BCIWI 12 M 84 01 85 05
00007 BCIWI 13 M 84 01 85 05
00008 ;
00009 PROC PRINT ;
00010 RUN ;

```

화면 [라 2]의 Program Edit 화면은 EXEM1 이라는 프로그램의 코딩내용이다.

- 프로그램 작성요령



[PF 3] (or [PF 15]) key 를 누르면 자료처리부분의 PRINT 명령에 따라 화면 [라 3]이 나온다.

[라 3]

OBS	DBNAME	DBKEY	HQY	FROMY	FROMM	TOY	TOM
1	BCIWI	11	M	84	1	85	5
2	BCIWI	12	M	84	1	85	5
3	BCIWI	13	M	84	1	85	5

화면 [라 2]의 결과화면이다. 이용자가 원하는 자료를 얻기 위해 제공한 정보가 제대로 SAS에 전해 졌는지 확인하기 위해서 앞 화면에서 Run 을 시켜 보았다.

위 화면을 확인함으로써 원하는 자료의 검색을 위해 필요한 정보가 제대로 전해 졌음을 알 수 있다. 다시 원 화면인 PROGRAM Edit 화면으로 가기 위해선 [PF 3] (or [PF 15]) key 를 누르면 된다.

(2) DB 자료를 출력해 주는 Sample Program (EXEM2)

다음은 요구한 DB 자료의 정보를 가지고 DB에 가서 검색해 온 결과를 Print 하여 어떠한 자료가 얻어졌는가를 보도록 하자. 화면 [라 1]과 같은 방법으로 미리 만들어 놓은 Sample Program EXEM 2를 불러 오자.

[라4]

```
COMMAND ==> SAS(R) LOG 14:57
PERSONNEL OR INSTALLATION SAS REPRESENTATIVE.
 12 RUN ;
-----
COMMAND ==> COPY B(EXEM2) PROGRAM EDITOR
00001
00002
00003
00004
00005
00006
00007
00008
00009
```

화면 [라4]와 같이 Program Edit 화면의 'COMMAND ==>'란에 'COPY B(EXEM2)'를 Typing 하고, ENTER key를 누르면 Sample Program EXEM2가 화면 [라5]와 같이 나온다.

[라5]

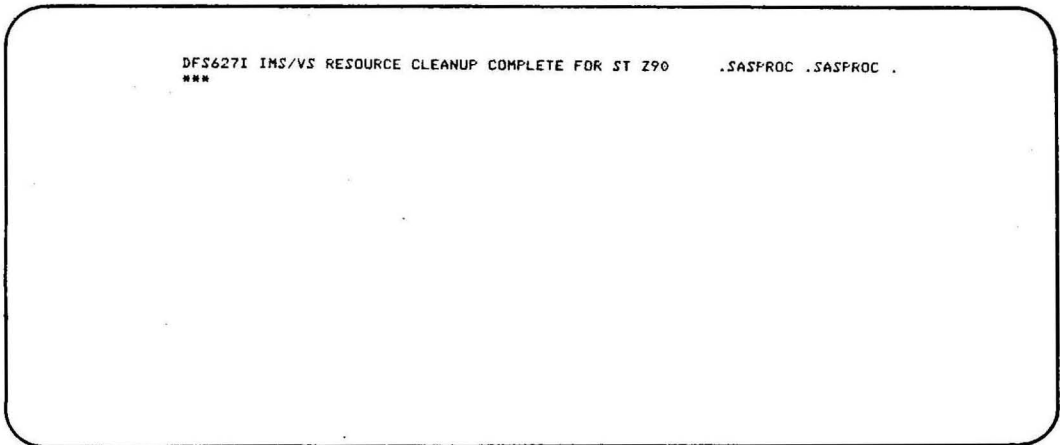
```
COMMAND ==> SAS(R) LOG 15:01
 24 PROC PRINT ,
 25 RUN ;
-----
COMMAND ==> PROGRAM EDITOR
00001 OPTIONS NOSOURCE2 ;
00002 DATA INDATA1;
00003 LENGTH DBNAME $5 DBKEY $9 MQY $1;
00004 INPUT DBNAME $ DBKEY $ MQY $ FROMY FROMH TOY TOM @@;
00005 CARDS;
00006 BCIWI 11 M 84 01 85 05
00007 BCIWI 12 M 84 01 85 05
00008 BCIWI 13 M 84 01 85 05
00009 ;
00010 XINCLUDE B(DBREAD) ;
00011 /*** NOW ! SAS WORK.BOSDB IS CREATED ==> ((BOSDB)) ***/
00012 PROC PRINT ,
00013 RUN ;
```

화면 [라5]의 내용은 'OPTIONS NOSOURCE2'와 'INCLUDE % B (DBREAD)'라는 명령문 외에는 EXEM 1과 똑같다.

- OPTIONS NOSOURCE2 → INCLUDE %명령을 수행한 후에 발생될 프로그램의 SOURCE LIST를 출력시키지 말라는 명령문
- INCLUDE % B (DBREAD)
 - DB를 읽어내기 위하여 사용되는 Subroutine Program의 이름
 - DBREAD라는 프로그램이 수록되어 있는 Datasets 이름 (DB·SAS라는 Dataset을 B라는 이름으로 ALLOCATE했음)
 - Dataset B에 있는 DBREAD라는 프로그램을 현재 % INCLUDE 문이 있는 위치에 Copy해 넣으라는 명령어

PF3 (or **PF15**) key 를 누르면 화면 [라6]이 나온다.

[라6]



이는 SAS 프로그램이 이용자가 원하는 자료를 Data Base에서 성공적으로 검색했다는 것을 알려주는 message이다. 이 화면을 확인한 후 **ENTER** Key를 누르면 다음 결과인 화면 [라7]을 얻게 된다.

[라 7]

OBS	COL1	COL2	COL3
1	1263	1201	1193
2	1278	1210	1198
3	1286	1224	1207
4	1287	1228	1217
5	1284	1236	1244
6	1265	1239	1262
7	1287	1243	1281
8	1264	1247	1300
9	1275	1245	1303
10	1273	1249	1310
11	1279	1251	1310
12	1296	1261	1328
13	1295	1256	1337
14	1284	1257	1341
15	1276	1258	1356
16	1289	1265	1375
17	1305	1265	1407

화면 [라 7]에는 년월 표시 및 변수명이 없으므로 자료의 내용을 알 수가 없다. 사용자는 다만 순서에 따라 다음과 같이 대응시켜 가면서 결과를 알 수 있을 따름이다.

OBS	COL1	COL2	COL3
1	1263	1201	1193
2	1278	1210	1198
⋮	⋮	⋮	⋮
16	1289	1265	1375
17	1305	1265	1407



년 월	BCIWI11	BCIWI12	BCIWI13
84년 1월	1263	1201	1193
84년 2월	1278	1210	1198
⋮	⋮	⋮	⋮
85년 4월	1289	1265	1375
85년 5월	1305	1265	1407

참고) BCIWI 11 → 선행지수, BCIWI 12 → 동행지수, BCIWI 13 → 후행지수

(3) DB 자료를 읽어 새로운 Dataset을 만드는 Sample Program(EXEM3)

(2)항에서 본바와 같이 DB에서 자료를 읽어오면 변수명이나 년월표시가 사용자들이 알 수 있는 상태로 되어있지 않아 매우 불편하다. 이러한 문제는 SET 이란 명령을 이용하여 새로운 Dataset을 만듦으로써 해결할 수가 있다.

[라 8]

```

COMMAND ===)                                SAS(R) LOG 15:03

 99      /*** NOW ! SAS WORK.BOSDB IS CREATED ==) ((BOSDB)) ***/
100
100      PROC PRINT ;
101      RUN ;
-----
COMMAND ===) COPY B(EXEM3)                    PROGRAM EDITOR

00001
00002
00003
00004
00005
00006
00007
00008
00009
00010
    
```

새로운 Dataset을 만들기 위한 Sample Program명은 EXEM3 이므로 화면 [라 8]의 Program Edit 화면과 같이 'COMMAND===)'란에 'COPY B(EXEM3)'를 Typing 하고, **ENTER** key를 누르면 화면 [라 9]가 나온다.

[라 9]

```

COMMAND ===)                                SAS(R) LOG 15:06
-----
COMMAND ===)                                PROGRAM EDITOR
00001  OPTIONS NQSOURCE2 ;
00002  DATA INDATA1 ;
00003  LENGTH DBNAME $5 DBKEY $9 MQY $1 ;
00004  INPUT DBNAME $ DBKEY $ MQY $ FROMY FROMM TOY TOM @@ ;
00005  CARDS ;
00006  BCIWI 11          M 84 01 85 05
00007  BCIWI 12          M 84 01 85 05
00008  BCIWI 13          M 84 01 85 05
00009
00010  XINCLUDE B(DBREAD) ;
00011  /*** NOW ! SAS WORK.BOSDB IS CREATED ==) ((BOSDB)) ***/
00012
00013  DATA C(KEEP=LEADING COINCID LAGGING DATE YEAR) ;
00014  RENAME COL1=LEADING COL2=COINCID COL3=LAGGING ;
00015  SET BOSDB ;
00016  COL1=COL1/10 ;
00017  COL2=COL2/10 ;
00018  COL3=COL3/10 ;
00020  YEAR =INT(((_N_-1)/12) + 84 ;
00021  MONTH=MOD(_N_-1,12)+1 ;
00022  DATE=MDY(MONTH,1,YEAR) ;
00023  FORMAT DATE MONYY. ;
00024  IF 1(<=_N_<18) ;
00025
00026  PROC PRINT ;
00027  RUN ;
    
```

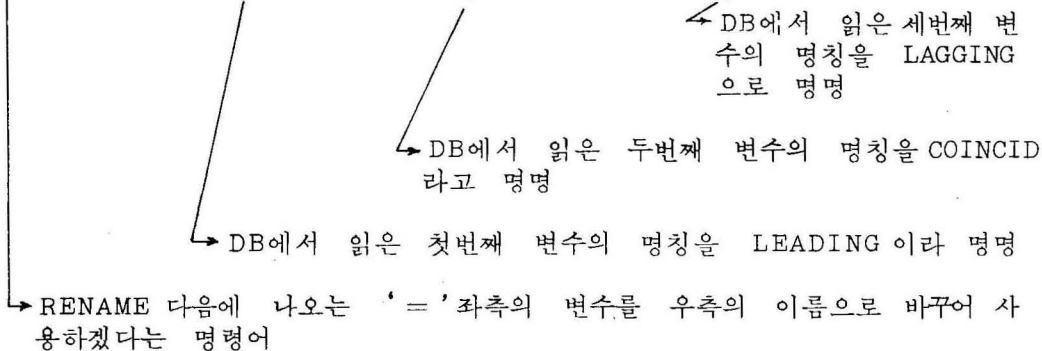
화면 [라9]의 내용은 □ 로 둘러싸인 부분을 제외하고는 화면 [라5]와 똑 같다. 그러므로 여기서는 새로운 Dataset 을 만들기 위한 □ 부분의 내용만을 설명하기로 한다.

○ DATA C(KEEP = LEADING COINCID LAGGING DATE)

OBS	LEADING	COINCID	LAGGING	DATE
⋮	⋮	⋮	⋮	⋮

위와 같은 형태의 Dataset 을 만들겠다는 문장으로 _____ 이 그어진 부분은 새로운 Dataset 에서 사용할 변수명이며, 이는 사용자가 임의로 정해준 것이다. ----- 이 그어진 부분은 주어진 그대로 사용하면 된다.

○ RENAME COL1=LEADING COL2=COINCID COL3=LAGGING



_____ 이 그어진 부분은 사용자가 임의로 정해주지만 ----- 이 그어진 부분은 주어진대로 사용해야 한다.

○ SET BOSDB

새로운 Dataset 을 만들기 위하여 사용할 SAS dataset 이름

이미 만들어진 하나 혹은 둘 이상의 SAS dataset 으로 부터 새로운 SAS dataset 을 만들기 위하여 사용하는 명령어

○ $\boxed{\begin{array}{l} \text{COL1} = \text{COL1}/10 \\ \text{COL2} = \text{COL2}/10 \\ \text{COL3} = \text{COL3}/10 \end{array}}$ 오른쪽 변수의 값을 10으로 나누어 그 결과를
 왼쪽 변수에 넣으라는 명령문

○ $\boxed{\text{YEAR} = \text{INT}((_N_ - 1)/12) + 84}$

- ① INT ()라는 명령어는 ()의 값을 계산하여 정수만을 취하라는 명령어
- ② $_N_$ 는 1부터 시작하여 관찰치의 갯수(여기서는 17)에 이를 때까지 하나씩 증가해 가는 변수
- ③ YEAR는 INT ()값에 84를 더한 값을 할당할 장소로써 여기서는 $_N_$ 의 값이 1~12일 때는 84가, 13~17일 때는 85가 YEAR에 수록된다.

○ $\boxed{\text{MONTH} = \text{MOD}(_N_ - 1, 12) + 1}$

- ① MOD ($_N_ - 1, 12$)는 $_N_ - 1$ 을 12로 나누고 그 나머지를 취하라는 명령어.
- ② MONTH에는 MOD ()값에 1을 더한 값을 할당하라는 뜻으로 $_N_$ 의 값이 .

1, 2, 3 12

13, 14, 15 24일 때

MONTH 값은

1, 2, 3 12

1, 2, 3 12임을 알 수 있다.

○ $\boxed{\begin{array}{l} \text{DATE} = \text{MDY}(\text{MONTH}, 1, \text{YEAR}) \\ \text{FORMAT DATE MONYY.} \end{array}}$

- ① MDY (, ,)명령어에서 (, ,)안에는 첫번째는 월, 두번째는 날짜, 세번째는 년을 기억하고 있는 변수를 넣고 이 변수들의 값을 조합하여 DATE

라는 변수에 기억시키되 FORMAT 문에서 지정한 방식으로 할당하라는 뜻이다.

② DATE 변수의 형식은 MONYY 방식, 즉 월 세자리, 년 두자리가 되도록 할 것.

○ $\boxed{\text{IF } 1 \leq N < 18}$

-N-은 SAS가 DATA Step을 수행할때 한개의 관측치(observation)를 취할때 마다 1,2,3,..... 등으로 증가되어가는 SAS 자동변수이므로 위문장은 1부터 17까지의 관측치를 취하겠다는 뜻이다.

화면 [라 9]에서 $\boxed{\text{PF 3}}$ (or $\boxed{\text{PF 15}}$) key 를 누르면 그 결과가 화면 [라 10]과 같이 나온다.

[라 10]

OBS	LEADING	COINCID	LAGGING	DATE
1	126.3	120.1	119.3	JAN84
2	127.8	121.0	119.8	FEB84
3	128.6	122.4	120.7	MAR84
4	128.7	122.8	121.7	APR84
5	128.4	123.6	124.4	MAY84
6	128.5	123.9	126.2	JUN84
7	128.7	124.3	128.1	JUL84
8	128.4	124.7	130.0	AUG84
9	127.5	124.5	130.3	SEP84
10	127.3	124.9	131.0	OCT84
11	127.9	125.1	131.0	NOV84
12	129.6	126.1	132.8	DEC84
13	129.5	125.6	133.7	JAN85
14	128.4	125.7	134.1	FEB85
15	127.6	125.8	135.6	MAR85
16	128.9	126.5	137.5	APR85
17	130.5	126.5	140.7	MAY85

(4) 회귀분석 Sample 프로그램 (EXEM4)

회귀분석 Sample 프로그램명은 EXEM4이다. 화면 [라 11]과 같이 Program Edit 화면의 'COMMAND ==>'란에 'COPY B (EXEM4)'를 Typing 하고,

ENTER key 를 누르면 화면 [라 12]가 나온다.

[라 11]

```

COMMAND ==>
SAS(R) LOG 15:09
-----
COMMAND ==> COPY B(EXEM4)
PROGRAM EDITOR

00001
00002
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012
00013

```

[라 12]

```

-----
Command ==>
Program Editor

00001  OPTIONS NOSOURCE2 ;
00002  TSO ALLOC F(DB) DA(DBSAS) SHR ;
00003  DATA INDATA1;
00004  LENGTH DBNAME $5 DBKEY $9 MQY $1;
00005  INPUT DBNAME $ DBKEY $ MQY $ PROMY FROMN TOY TOM @@;
00006  CARDS;
00007  BCIV1 11      N 84 01 85 05
00008  BCIV1 12      N 84 01 85 05
00009  BCIV1 13      N 84 01 85 05
00010  ;
00011  $INCLUDE DB(DBREAD) ;
00012  /*** HOW ! SAS WORK.BOSDB IS CREATED **>> <<BOSDB>> ***/
00013  /*** HOW ! SAS WORK.BOSDB IS CREATED **>> <<BOSDB>> ***/
00014  DATA C(KEEP=LEADING COINCID LAGGING DATE YEAR) ;
00015  RENAME COL1=LEADING COL2=COINCID COL3=LAGGING ;
00016  SET BOSDB ;
00017  COL1=COL1/10 ;
00018  COL2=COL2/10 ;
00019  COL3=COL3/10 ;
00020
00021  YEAR =INT((_N_-1)/12) + 84 ;

```

화면 [라 12]를 보면 RUN 명령이 없는 것으로 보아 Program의 끝이 안 나와 있음을 알 수 있다. 이때 **PF 8** (or **PF 20**) key를 누르면 프로그램의 다음 부분이 화면 [라 13]과 같이 나온다. 물론 이 화면에서 **PF3** (or **PF15**) key 를 눌러 작업을 수행시키면 화면 [라 14]가 나온다.

[라 13]

```

-----
Command ==>                                     Program Editor
00022      MONTH=MOD(_N_-1,12)+1 ;
00023      DATE=MDY(MONTH,1,YEAR) ;
00024      FORMAT DATE MONYY. ;
00025      IF _J<= _N <18 ;
00026      PROC REG DATA=C ;
00027      MODEL COINCID = LEADING / DV ;
00028      OUTPUT OUT=O P=PRED L95=L95 U95=U95 R=RESID ;
00029      TITLE 'OUTPUT OF SIMPLE REGRESSION PROCEDURE' ;
00030      PROC PLOT DATA = O ;
00031      PLOT COINCID*DATE='A' PRED*DATE='P' U95*DATE='U' L95*DATE='L'
00032      / OVERLAY ;
00033      PLOT RESID*DATE / VREF=0 ;
00034      RUN ;
00035

```

화면 [라 13]에는 나머지 프로그램이 모두 나와 있다. 이 내용을 확인하고 다시 화면 [라 12]로 돌아가려면 PF7 (or PF 19) key를 누르면 된다.

상기 프로그램은 앞에서 언급한 구상에 의해서 동행 종합지수(COINCID)에 대한 선행지수(LEADING)의 회귀모형을 PROC REG를 불러서 MODEL COINCID = LEADING을 이용하여 단순회귀모형으로 84.1월부터 85.10월까지의 표본자료를 이용하여 추정해 보고자 하는 것과 이렇게 하여 추정된 예측치와 실측치를 Plotting 하는 것 그리고 예측치와 실측치의 차이인 잔차들의 평가를 하여 추후의 좀더 나은 모형을 구상하기 위한 잔차 Plotting 하는 법을 보여주고 있다. (잔차에 대한 평가 및 제거방법에 대한 구체적인 방법은 전문적인 회귀분석책이나 계량경제학책을 이용하기 바람)

PF3 (or PF15) key를 누르면 화면 [라 14]가 나온다.

[라 14]

OUTPUT OF SIMPLE REGRESSION PROCEDURE					
DEP VARIABLE: COINCID					
ANALYSIS OF VARIANCE					
SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PROB>F
MODEL	1	18.55728016	18.55728016	7.628	0.0145
ERROR	15	36.49330808	2.43288721		
C TOTAL	16	55.05058824			
ROOT MSE		1.559772	R-SQUARE	0.3371	
DEP MEAN		124.3235	ADJ R-SQ	0.2929	
C.V.		1.254607			

화면 [라 14]에서 R-SQUARE 값을 보면 모형의 결정력이 그다지 높지 않음을 알 수 있다. 이는 선행과 동행지수의 개념상 시차문제의 개입이 내재해 있음을 짐작하게 되는데 이에 대한 여부는 실제자료에 대한 좀더 다각적인 연구가 필요하다. 다시 PF3 (or PF 15) key 를 누르면 화면 [라 15]가 나온다.

[라 15]

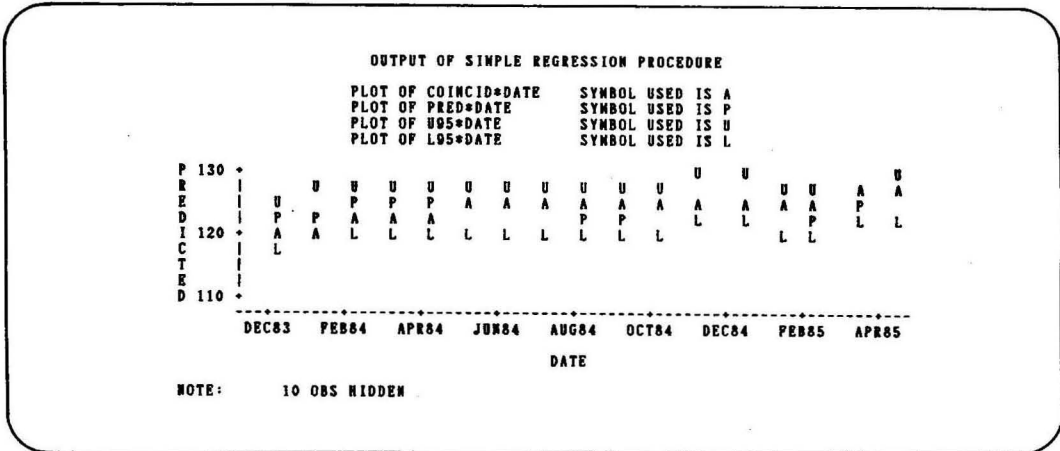
OUTPUT OF SIMPLE REGRESSION PROCEDURE					
VARIABLE	DF	PARAMETER ESTIMATE	STANDARD ERROR	T FOR H0: PARAMETER=0	PROB > T
INTERCEP	1	-17.45440278	51.33626346	-0.340	0.7386
LEADING	1	1.10429068	0.39984092	2.762	0.0145
DURBIN-WATSON D		0.310			
(FOR NUMBER OF OBS.)		17			
1ST ORDER AUTOCORRELATION		0.794			

화면 [라 15]는 계속적인 회귀분석결과의 하나로서 모형의 parameter(모수)에 대한 추정치, 추정치에 대한 표준오차, t-통계량값이 나오며 이는 모수 추정치의 평가에 사용된다. 즉 유의수준을 $\alpha = 0.05$ 라 했을 때 $PROB > |T|$ 값이 이보다는 작아야 상정된 모형의 모수 추정치로서 받아들일 수가 있게 된다.

그리고 자기상관의 여부를 확인하기 위한 Durbin-Watson D 통계량이 나오게 된다.

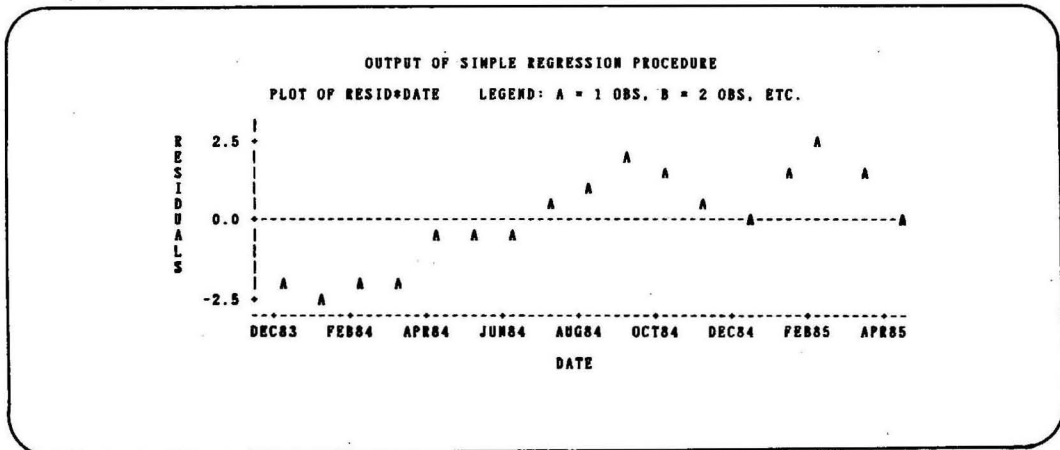
(Durbin-Watson D 통계량에 대해서는 회귀이론서를 참고하기 바람)
 다시 **PF3** (or **PF15**) key 를 누르면 화면 [라 16]이 나온다.

[라 16]



상기 화면은 실측치와 예측치, 예측치의 신뢰상, 하한을 Plotting 한 결과이다. 다시 **PF3** (or **PF15**) key 를 누르면 화면 [라 17] 이 나온다.

[라 17]



이 화면은 화면 [라 16]에서의 실측치와 예측치의 차(이를 잔차라고 함)을 plotting 시킨 결과이다.

(5) 시계열자료 PLOTTING (EXEM 5)

아래 화면 [라 18]과 [라 19]는 시계열자료를 PLOTTING하기 위한 Sample Program으로써 'COPY B(EXEM5)'를 key in하여 얻을 수 있다.

[라 18]

```
-----  
Command ==>                                     Program Editor  
  
00001  OPTIONS NOSOURCE2 ;  
00002  TSO ALLOC F(DB) DA(DBSAS) SHR ;  
00003  DATA INDATA1 ;  
00004  LENGTH DBNAME $5 DBKEY $9 MQY $1 ;  
00005  INPUT DBNAME $ DBKEY $ MQY $ FROMY FROMM TOY TOM ## ;  
00006  CARDS ;  
00007  BCIWI 11          M 83 01 85 07  
00008  BCIWI 12          M 83 01 85 07  
00009  BCIWI 13          M 83 01 85 07  
00010  ;  
00011  XINCLUDE DB(DBREAD) ;  
00012  /** NOV ! SAS WORK.BOSDB IS CREATED ==> <<BOSDB>> **/  
00013  /** NOV ! SAS WORK.BOSDB IS CREATED ==> <<BOSDB>> **/  
00014  DATA C(KEEP=LEADING COINCID LAGGING DATE YEAR) ;  
00015  RENAME COL1=LEADING COL2=COINCID COL3=LAGGING ;  
00016  SET BOSDB ;  
00017  COL1=COL1/10 ;  
00018  COL2=COL2/10 ;  
00019  COL3=COL3/10 ;  
00020  YEAR =INT((_M_-1)/12) + 83 ;  
00021
```

[라 19]

```
-----  
Command ==>                                     Program Editor  
  
00022  MONTH=MOD(_M_-1,12)+1 ;  
00023  DATE=NDY(MONTH,1,YEAR) ;  
00024  FORMAT DATE MONYY. ;  
00025  IF 1<=_M_<32 ;  
00026  PROC TIMEPLOT DATA=C ;  
00027  PLOT LEADING ;  
  
00028  ID DATE ;  
00029  TITLE 'OUTPUT OF TIMEPLOT PROCEDURE' ;  
00030  RUN ;  
00031
```

화면 [라 18]이나 [라 19]에서 PF 3 (or PF 15) key 를 누르면 화면 [라 20]이 나온다.

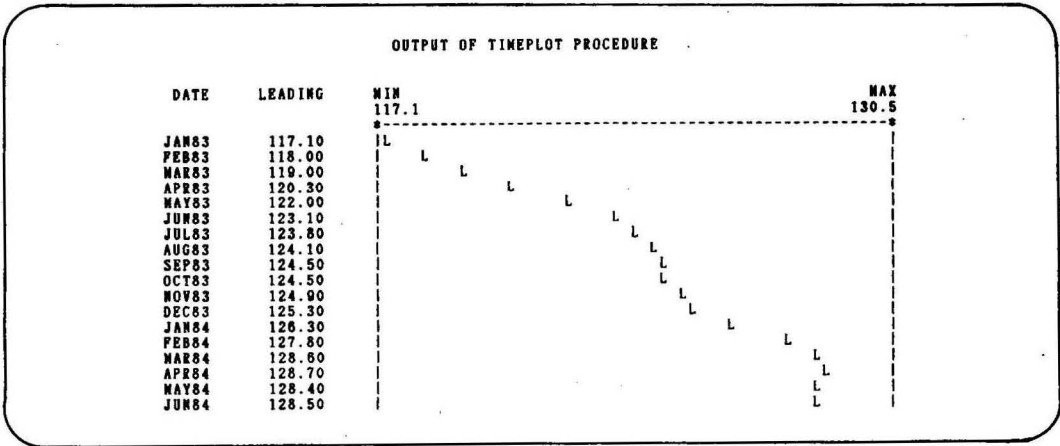
[라 20]

```

DFS6271 IMS/VS RESOURCE CLEANUP COMPLETE FOR ST Z90 .SASPROC .SASPROC .
***
    
```

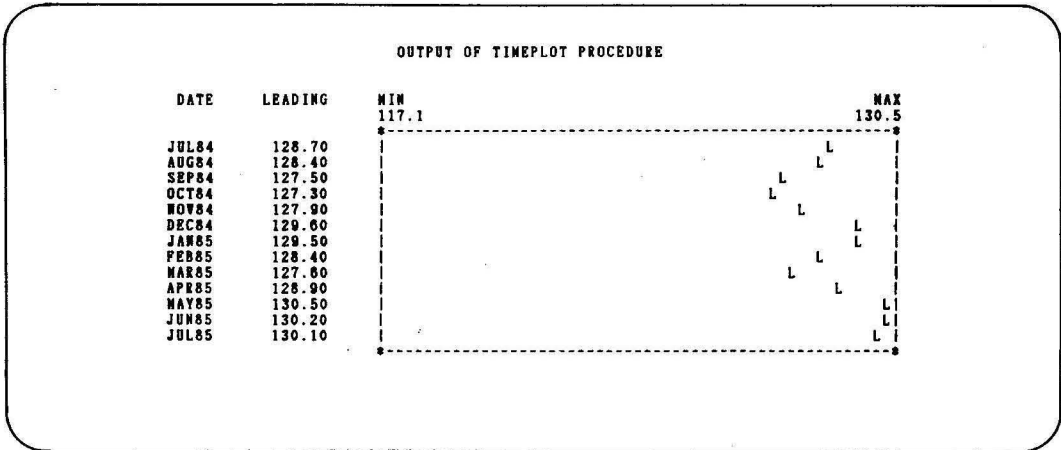
[ENTER] key 나 [PF3] (or [PF15]) key 를 누르면 결과화면 [라 21]이 나온다.

[라 21]



다시 [PF3] (or [PF15]) key 를 누르면 화면 [라 22]가 나온다.

[라 22]



(6) FORECASTING (EXEM6)

Forecasting 하기 위해서는 Program Edit 화면의 Command 란에 'COPY B(EXEM6)' 을 key in 하고, ENTER key 를 누르면 화면 [라 23]이 나오고 이어서 화면 [라 24]를 볼 수 있다.

[라 23]

```
-----
Command ***>                                     Program Editor
00001  OPTIONS NOSOURCE2 ;
00002  TSO ALLOC F(DB) DA(DRSAS) SHR ;
00003  DATA INDATA1;
00004  LENGTH DBNAME $5 DBKEY $9 MQY $1;
00005  INPUT DBNAME $ DBKEY $ MQY $ FROMY FROMM TOY TOM @@;
00006  CARDS;
00007  BC1V1 11          M 83 01 85 07
00008  BC1V1 12          M 83 01 85 07
00009  BC1V1 13          M 83 01 85 07
00010  ;
00011  XINCLUDE DB(DBREAD) ;
00012  /**/ NOW ! SAS WORK.BOSDB IS CREATED ==> <<BOSDB>> ***/
00013  /**/ NOW ! SAS WORK.BOSDB IS CREATED ==> <<BOSDB>> ***/
00014  DATA C(KEEP=LEADING COINCID LAGGING DATE YEAR) ;
00015  RENAME COL1=LEADING COL2=COINCID COL3=LAGGING ;
00016  SET BOSDB ;
00017  COL1=COL1/10 ;
00018  COL2=COL2/10 ;
00019  COL3=COL3/10 ;
00020
00021  YEAR =INT(((_M_-1)/12) + 83 ;
```

[라 24]

```

-----
Command ==>                                     Program Editor
00022      MONTH=MOD(.N.-1,12)+1 ;
00023      DATE=MDY(MONTH,1,YEAR) ;
00024      FORMAT DATE MONYY. ;
00025      IF 1<=.N.<32 ;
00026 PROC PRINT DATA=C;
00027      TITLE 'ORIGINAL DATA';
00028 PROC FORECAST DATA=C OUT=O OUTEST=T
00029      INTERVAL=MONTH OUTDATA OUT1STEP OUTLIMIT;
00030      ID DATE;
00031      VAR LEADING COINCID LAGGING ;
00032 PROC PRINT DATA=O;
00033      TITLE 'THE OUTPUT ACTUAL AND FORECASTED FROM PROC FORECAST';
00034 PROC PLOT DATA=O;
00035      PLOT (LEADING COINCID LAGGING)*DATE=_TYPE_ ;
00036      TITLE 'PLOT OF FORECAST';
00037 PROC PRINT DATA=T;
00038      TITLE 'THE ESTIMATES FROM PROC FORECAST';
00039 PROC FORECAST DATA=C OUT=D INTERVAL=MONTH;
00040      ID DATE;
00041      VAR LEADING COINCID LAGGING ;
00042 PROC PRINT DATA=D;
00043      TITLE 'PROC FORECAST OUTPUT IN SIMPLEST INVOCATION';
00044      RUN;
00045

```

화면 [라 24]의 문번호 28에서 PROC FORECAST 를 불러낸다.

O라는 Data set 이름에 이 PROC 의 결과를 담고 (OUT=0), 추정된 값들은 DATA SET T에 담으며 (OUTEST = T), 예측에 필요한 기간의 단위로 MONTH를 사용한다. (INTERVAL = MONTH). 결과로 얻어진 변수중 OUT DATA, OUT1STEP, OUTLIMIT 변수를 취하겠다는 것을 PROC 문장의 OPTION으로 주었다. ID는 DATE로 하고(문번호 30). VAR 문장에 LEADING COINCID LAGGING 을 명시함으로써 3 변수 모두에 대한 예측을 요구하였다. (문번호 31). 문번호 32에서는 생성된 DATA SET을 프린트 하였으며 문번호 34에서는 이 자료를 이용 Plotting 을 하였고 문번호 37에서는 예측치들을 PRINT 하였다. 그리고 문번호 39에서는 이용자가 가장 간단히 PROC FORECAST 를 이용하는 법을 Coding 하였다.

화면 [라 23]이나 [라 24]에서 PF3 (or PF15) key 를 누르면 화면 [라 25]가 나온다.

[라 25]

```
DFS0271 IMS/VS RESOURCE CLEANUP COMPLETE FOR ST Z90 .SASPROC .SASPROC .  
***
```

[ENTER] key 나 [PF 3] (or [PF 15]) key 를 누르면 분석에 사용할
자료가 화면 [라 26] 과 같이 나온다.

[라 26]

ORIGINAL DATA					
OBS	LEADING	COINCID	LAGGING	YEAR	DATE
19	128.7	124.3	128.1	84	JUL84
20	128.4	124.7	130.0	84	AUG84
21	127.5	124.5	130.3	84	SEP84
22	127.3	124.9	131.0	84	OCT84
23	127.9	125.1	131.0	84	NOV84
24	129.6	126.1	132.8	84	DEC84
25	129.5	125.6	133.7	85	JAN85
26	128.4	125.7	134.1	85	FEB85
27	127.6	125.8	135.6	85	MAR85
28	128.9	126.5	137.5	85	APR85
29	130.5	126.5	140.7	85	MAY85
30	130.2	125.4	141.4	85	JUN85
31	130.1	125.5	143.7	85	JUL85

계속 [PF 3] (or [PF 15]) key 를 누르면 결과화면이 [라 27], [라
28], [라 29], [라 30], [라 31] 이 나온다.

[라 27]

THE OUTPUT ACTUAL AND FORECASTED FROM PROC FORECAST						
OBS	DATE	_TYPE_	_LEAD_	LEADING	COINCID	LAGGING
1	JAN83	ACTUAL	0	117.100	108.300	108.400
2	JAN83	FORECAST	0	120.652	110.911	106.559
3	FEB83	ACTUAL	0	118.000	109.000	109.900
4	FEB83	FORECAST	0	118.460	109.304	109.225
5	MAR83	ACTUAL	0	119.000	109.600	110.200
6	MAR83	FORECAST	0	119.209	109.992	110.664
7	APR83	ACTUAL	0	120.300	110.500	111.200
8	APR83	FORECAST	0	120.031	110.598	110.502
9	MAY83	ACTUAL	0	122.000	111.800	110.900
10	MAY83	FORECAST	0	121.089	111.455	111.414
11	JUN83	ACTUAL	0	123.100	113.200	111.700
12	JUN83	FORECAST	0	122.394	112.485	111.658
13	JUL83	ACTUAL	0	123.800	114.500	112.500
14	JUL83	FORECAST	0	123.288	113.943	112.572
15	AUG83	ACTUAL	0	124.100	115.800	114.200
16	AUG83	FORECAST	0	123.894	115.144	113.920
17	SEP83	ACTUAL	0	124.500	116.800	115.800
18	SEP83	FORECAST	0	124.212	116.174	115.640

[라 28]

THE OUTPUT ACTUAL AND FORECASTED FROM PROC FORECAST						
OBS	DATE	_TYPE_	_LEAD_	LEADING	COINCID	LAGGING
19	OCT83	ACTUAL	0	124.500	117.300	117.000
20	OCT83	FORECAST	0	124.602	117.119	117.279
21	NOV83	ACTUAL	0	124.900	118.300	117.500
22	NOV83	FORECAST	0	124.704	117.808	118.288
23	DEC83	ACTUAL	0	125.300	118.800	118.500
24	DEC83	FORECAST	0	125.094	118.753	118.758
25	JAN84	ACTUAL	0	126.300	120.100	119.300
26	JAN84	FORECAST	0	126.308	119.270	119.788
27	FEB84	ACTUAL	0	127.800	121.000	119.800
28	FEB84	FORECAST	0	127.006	120.471	120.849
29	MAR84	ACTUAL	0	128.600	122.400	120.700
30	MAR84	FORECAST	0	128.040	121.331	121.517
31	APR84	ACTUAL	0	128.700	122.800	121.700
32	APR84	FORECAST	0	128.500	122.617	122.580
33	MAY84	ACTUAL	0	128.400	123.600	124.400
34	MAY84	FORECAST	0	128.384	123.050	123.828
35	JUN84	ACTUAL	0	128.500	123.900	128.200
36	JUN84	FORECAST	0	128.079	123.824	128.335

[라 29]

THE OUTPUT ACTUAL AND FORECASTED FROM PROC FORECAST						
OBS	DATE	_TYPE_	_LEAD_	LEADING	COINCID	LAGGING
37	JUL84	ACTUAL	0	128.700	124.300	128.100
38	JUL84	FORECAST	0	128.176	124.171	128.071
39	AUG84	ACTUAL	0	128.400	124.700	130.000
40	AUG84	FORECAST	0	128.436	124.604	129.322
41	SEP84	ACTUAL	0	127.500	124.500	130.300
42	SEP84	FORECAST	0	128.314	125.036	130.873
43	OCT84	ACTUAL	0	127.300	124.900	131.000
44	OCT84	FORECAST	0	127.853	124.956	131.077
45	NOV84	ACTUAL	0	127.900	125.100	131.000
46	NOV84	FORECAST	0	127.803	125.389	131.609
47	DEC84	ACTUAL	0	129.600	126.100	132.800
48	DEC84	FORECAST	0	128.328	125.650	132.099
49	JAN85	ACTUAL	0	129.500	125.800	133.700
50	JAN85	FORECAST	0	129.508	126.595	133.935
51	FEB85	ACTUAL	0	128.400	125.700	134.100
52	FEB85	FORECAST	0	129.272	126.259	135.265
53	MAR85	ACTUAL	0	127.600	125.800	135.800
54	MAR85	FORECAST	0	128.482	126.435	135.584

[라 30]

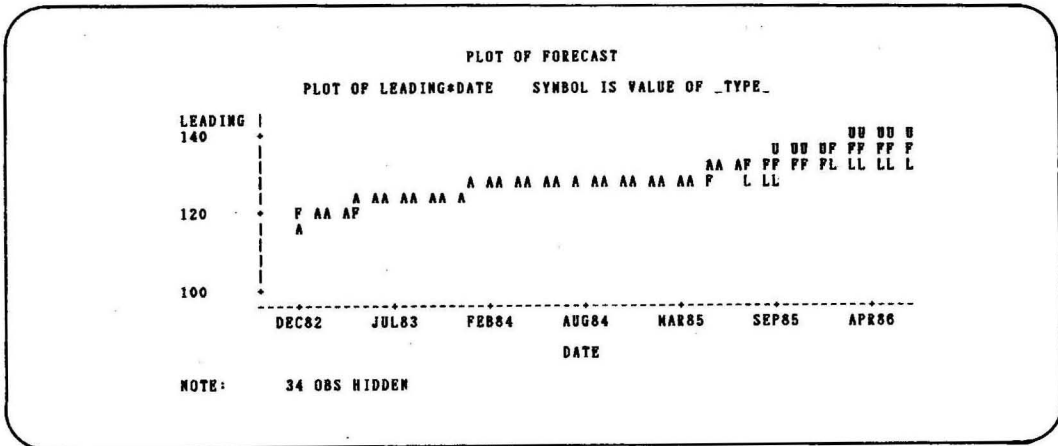
THE OUTPUT ACTUAL AND FORECASTED FROM PROC FORECAST						
OBS	DATE	_TYPE_	_LEAD_	LEADING	COINCID	LAGGING
55	APR85	ACTUAL	0	128.900	126.500	137.500
56	APR85	FORECAST	0	128.089	126.612	137.107
57	MAY85	ACTUAL	0	130.500	126.500	140.700
58	MAY85	FORECAST	0	129.261	127.300	139.125
59	JUN85	ACTUAL	0	130.200	125.400	141.400
60	JUN85	FORECAST	0	130.576	127.391	141.845
61	JUL85	ACTUAL	0	130.100	125.500	143.700
62	JUL85	FORECAST	0	130.500	126.543	142.377
63	AUG85	FORECAST	1	130.684	126.719	143.790
64	AUG85	L95	1	128.661	124.514	142.014
65	AUG85	U95	1	132.707	128.924	145.566
66	SEP85	FORECAST	2	131.500	127.851	144.221
67	SEP85	L95	2	129.049	125.015	141.970
68	SEP85	U95	2	133.951	130.687	146.472
69	OCT85	FORECAST	3	132.320	128.908	144.398
70	OCT85	L95	3	129.670	125.686	141.874
71	OCT85	U95	3	134.970	132.131	146.923
72	NOV85	FORECAST	4	132.957	129.902	145.105

[라 31]

THE OUTPUT ACTUAL AND FORECASTED FROM PROC FORECAST							
OBS	DATE	_TYPE_	_LEAD_	LEADING	COINCID	LAGGING	
73	NOV85	L95	4	130.204	126.423	142.545	
74	NOV85	U95	4	135.711	133.382	147.684	
75	DEC85	FORECAST	5	133.208	130.842	146.132	
76	DEC85	L95	5	130.398	127.183	143.559	
77	DEC85	U95	5	136.018	134.501	148.704	
78	JAN86	FORECAST	6	133.599	131.736	147.507	
79	JAN86	L95	6	130.754	127.948	144.877	
80	JAN86	U95	6	136.443	135.523	150.137	
81	FEB86	FORECAST	7	134.321	132.589	148.992	
82	FEB86	L95	7	131.454	128.709	146.299	
83	FEB86	U95	7	137.189	136.470	151.885	
84	MAR86	FORECAST	8	135.214	133.400	150.460	
85	MAR86	L95	8	132.329	129.450	147.730	
86	MAR86	U95	8	138.098	137.360	153.190	
87	APR86	FORECAST	9	135.741	134.201	151.798	
88	APR86	L95	9	132.841	130.197	149.055	
89	APR86	U95	9	138.841	138.205	154.542	
90	MAY86	FORECAST	10	135.935	134.987	152.993	
91	MAY86	L95	10	133.021	130.922	150.239	
92	MAY86	U95	10	138.849	139.013	155.747	
93	JUN86	FORECAST	11	136.331	135.713	154.076	
94	JUN86	L95	11	133.404	131.634	151.306	
95	JUN86	U95	11	139.259	139.792	156.846	
96	JUL86	FORECAST	12	136.826	136.440	155.110	
97	JUL86	L95	12	133.885	132.334	152.322	
98	JUL86	U95	12	139.787	140.546	157.897	

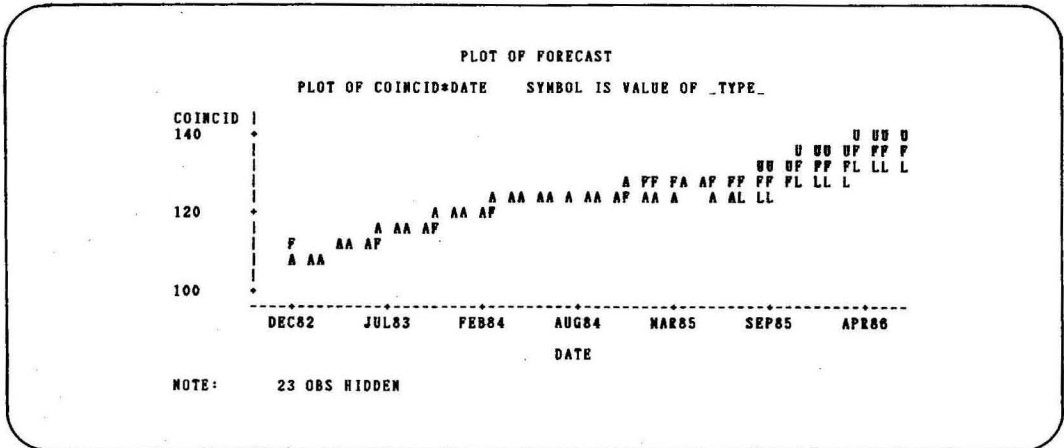
다시 **PF3** (or **PF15**) key 를 누르면 선행지수에 대한 FORECAST 결과를 이용하여 plotting 한 화면이 [라 32]와 같이 나온다.

[라 32]



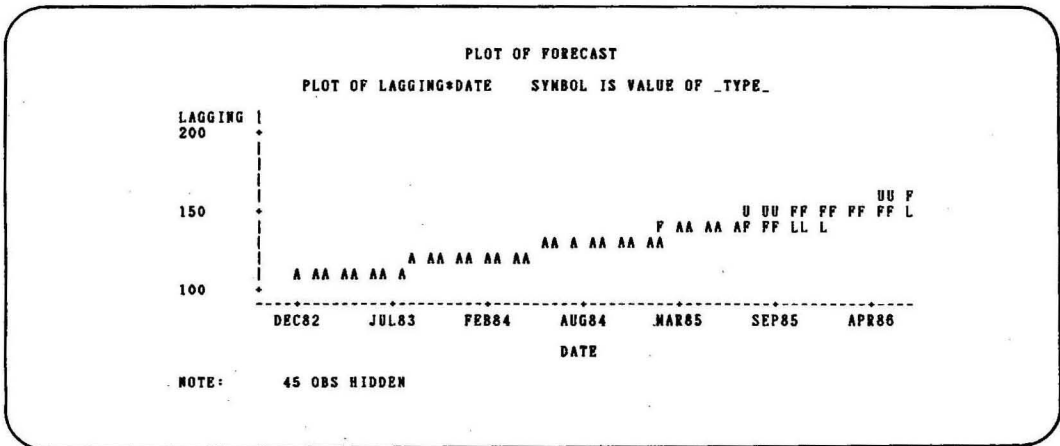
다시 **PF3** (or **PF15**) key 를 누르면 동행지수에 대한 FORECAST 결과를 이용하여 Plotting 한 결과가 화면 [라 33]과 같이 나온다.

[라 33]



다시 **PF3** (or **PF15**) key 를 누르면 후행지수에 대한 Forecast 결과를 이용하여 Plotting 한 결과가 화면 [라 34]와 같이 나온다.

[라 34]



다시 **PF3** (or **PF15**) key 를 누르면 화면 [라 35]가 나온다.

[라 35]

THE ESTIMATES FROM PROC FORECAST					
OBS	_TYPE_	DATE	LEADING	COINCID	LAGGING
1	N	JUL85	31	31	31
2	SIGMA	JUL85	0.9685446	1.05598	0.8503816
3	CONSTANT	JUL85	120.2884	110.289	105.4052
4	LINEAR	JUL85	0.3640323	0.6224597	1.15371
5	AR1	JUL85	0.7197519	0.8539776	0.8217204
6	AR2	JUL85	.	.	.
7	AR3	JUL85	.	.	-0.333515
8	AR4	JUL85	.	.	.
9	AR5	JUL85	.	.	.
10	AR6	JUL85	.	.	.
11	AR7	JUL85	.	.	.
12	AR8	JUL85	.	.	.
13	AR9	JUL85	.	.	.
14	AR10	JUL85	.	.	.
15	AR11	JUL85	.	.	.
16	AR12	JUL85	-0.232086	.	.
17	AR13	JUL85	.	.	.

화면 [라 35]에는 PROC FORECAST 에 의해 얻어진 각 시계열 자료에 대한 모수추정치가 일목요연하게 나와있다. LEADING, COINCID, LAGGING COLUMN 별로 _TYPE_ Column 과 비교하여 보면 각 시계열에 대한 모수추정치를 대응시켜 볼 수 있을 것이다.

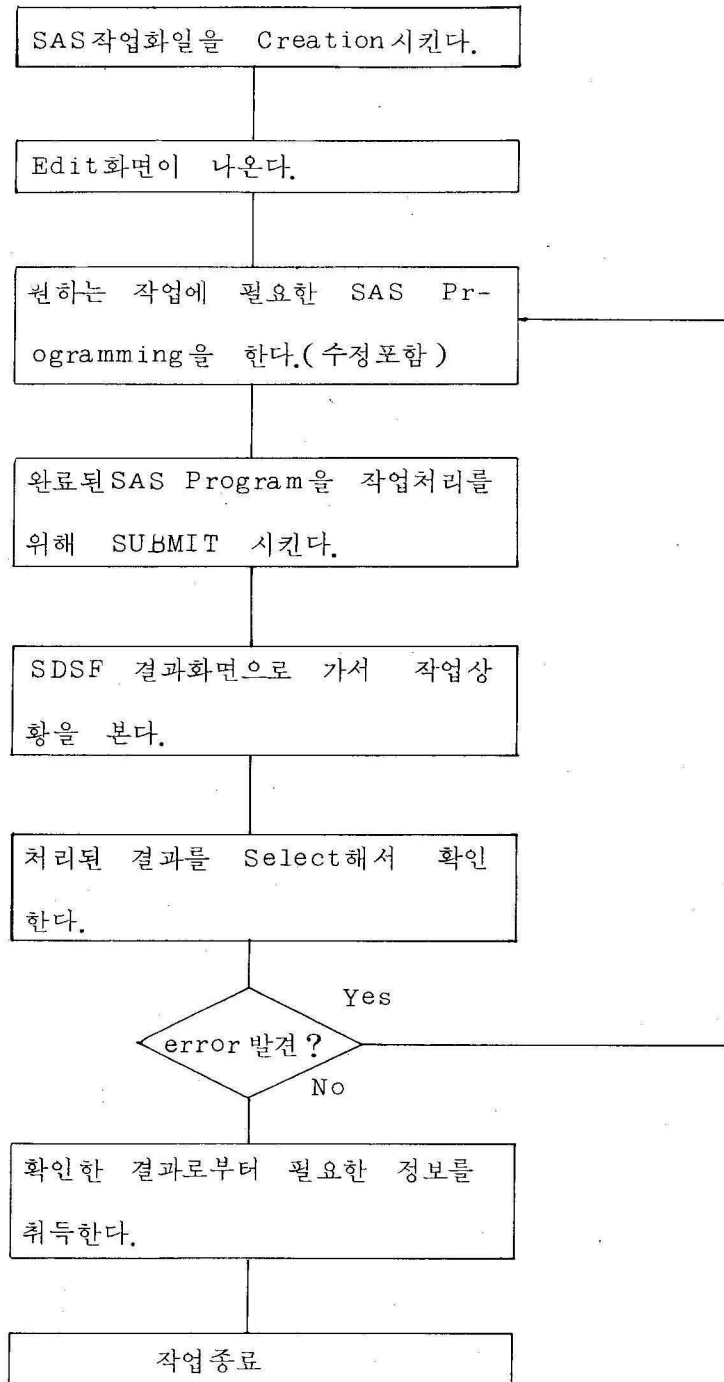
화면 [라 35]에서 PF 3 (or PF 15) key 를 누르면 최종결과로서 향후 1년간의 예측치가 화면 [라 36]과 같이 나온다.

[라 36]

PROC FORECAST OUTPUT IN SIMPLEST INVOCATION						
OBS	DATE	_TYPE_	_LEAD_	LEADING	COINCID	LAGGING
1	AUG85	FORECAST	1	130.684	126.719	143.790
2	SEP85	FORECAST	2	131.500	127.851	144.221
3	OCT85	FORECAST	3	132.320	128.908	144.398
4	NOV85	FORECAST	4	132.957	129.902	145.105
5	DEC85	FORECAST	5	133.208	130.842	146.132
6	JAN86	FORECAST	6	133.599	131.736	147.507
7	FEB86	FORECAST	7	134.321	132.589	148.992
8	MAR86	FORECAST	8	135.214	133.409	150.460
9	APR86	FORECAST	9	135.741	134.201	151.798
10	MAY86	FORECAST	10	135.935	134.967	152.993
11	JUN86	FORECAST	11	136.331	135.713	154.076
12	JUL86	FORECAST	12	136.826	136.440	155.110

3. Batch 방식 SAS 이용법

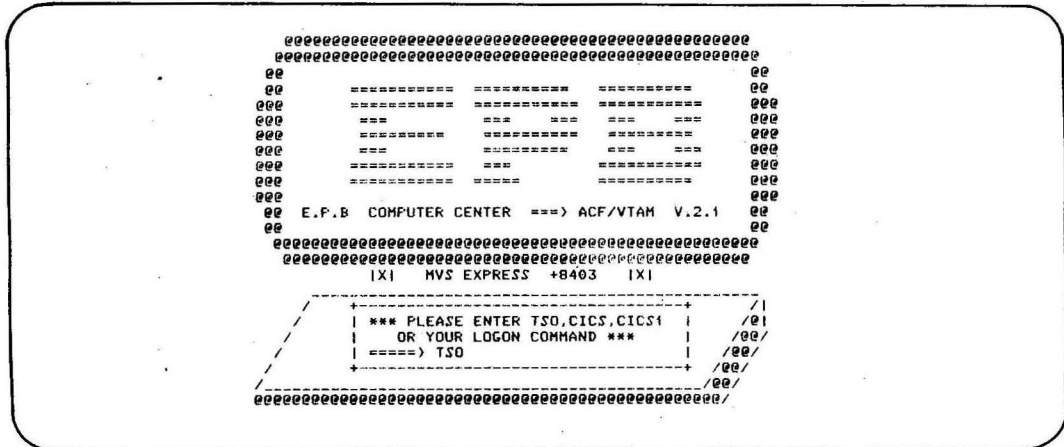
가. 작업순서



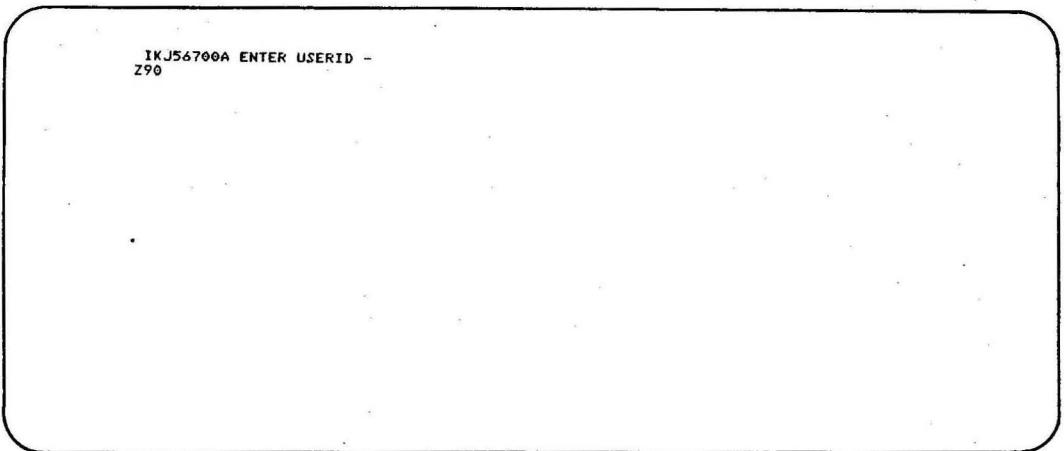
나. Program Edit 화면을 찾는 법

SAS Batch작업을 위해서는 Terminal의 최초화면 [나-1]에 'TSO'를 Key in하고 **ENTER** Key를 누르면 화면 [나-2]가 나온다.

[나-1]



[나-2]



화면 [나-2]에 UBER ID를 Key in하고 **ENTER** Key를 누르면 화면 [나-3]이 나온다.

[나- 3]

```
----- VS2 REL 3.8 TIME SHARING OPTION -----
PF1/PF13 ==> HELP   PF3/PF15 ==> LOGOFF   PA1 ==> ATTENTION   PA2 ==> RESHOW
YOU MAY REQUEST SPECIFIC HELP INFORMATION BY ENTERING A '?' IN ANY ENTRY FIELD.
ENTER LOGON PARAMETERS BELOW:                RACF LOGON PARAMETERS:

USERID   ==> Z90

PASSWORD ==>                                NEW PASSWORD ==>

PROCEDURE ==> SASPROC                        GROUP IDENT ==>

ACCT NMBR ==> 90

SIZE     ==> 4096

PERFORM  ==>

COMMAND  ==>

ENTER AN 'S' BEFORE EACH OPTION DESIRED BELOW:

-NOMAIL      -NONOTICE      -RECONNECT      -OIDCARD
```

화면 [나- 3]의 'PASSWORD==>'란에 이용자의 PASSWORD를 Key in 하고 **ENTER** Key 를 누르면 화면 [나- 4]가 나온다.

[나- 4]

```
ICH70001I Z90      LAST ACCESS AT 13:41:08 ON TUESDAY, DECEMBER 24, 1985
IKJ56455I Z90 LOGON IN PROGRESS AT 15:22:33 ON DECEMBER 24, 1985
IKJ56951I NO BROADCAST MESSAGES
***
```

화면 [나- 4]가 나올때까지 기다렸다가 화면 [나- 4]의 맨마지막줄에 '***'가 나온후, **ENTER** Key를 누르면 화면 [나- 5]가 나온다.

[나 - 5]

```
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ==> 2
                                USERID - Z90
0 ISPF PARMS - SPECIFY TERMINAL AND USER PARAMETERS    TIME - 15:22
1 BROWSE     - DISPLAY SOURCE DATA OR OUTPUT LISTINGS  TERMINAL - 3278
2 EDIT       - CREATE OR CHANGE SOURCE DATA           PF KEYS - 24
3 UTILITIES  - PERFORM UTILITY FUNCTIONS
4 FOREGROUND - INVOKE LANGUAGE PROCESSORS IN FOREGROUND
5 BATCH      - SUBMIT JOB FOR LANGUAGE PROCESSING
6 COMMAND    - ENTER TSO COMMAND OR CLIST
7 DIALOG TEST - PERFORM DIALOG TESTING
8 SDSF       - SPOOL DISPLAY AND SEARCH FACILITY
C CHANGES   - DISPLAY SUMMARY OF CHANGES FOR THIS RELEASE
T TUTORIAL   - DISPLAY INFORMATION ABOUT ISPF/PDF
X EXIT       - TERMINATE ISPF USING LOG AND LIST DEFAULTS

ENTER END COMMAND TO TERMINATE ISPF.
```

화면 [나 - 5] 'OPTION ==>' 란에 '2' 를 Key in 하고 **ENTER** Key 를 누르면 화면 [나 - 6] 이 나온다.

[나 - 6]

```
----- EDIT - ENTRY PANEL -----
COMMAND ==>
ISPF LIBRARY:
PROJECT ==> Z90
LIBRARY ==> PROGRAM ==> ==> ==>
TYPE ==> LIBRARY
MEMBER ==> SASSTEST (BLANK FOR MEMBER SELECTION LIST)
OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==> (IF NOT CATALOGED)
DATA SET PASSWORD ==> (IF PASSWORD PROTECTED)
PROFILE NAME ==> (BLANK DEFAULTS TO DATA SET TYPE)
```

화면 [나 - 6] 가 나오면 'MEMBER ==>' 란에 SAS Program 을 만들 새로운 이름을 Key in 하고 **ENTER** Key 를 누르면 화면 [나 - 7] 이 나온다.

[다 - 2]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
***** TOP OF DATA *****
000001 //Z90CDSAS JOB CLASS=B,MSGCLASS=X
***** BOTTOM OF DATA *****
```

화면 [다 - 2]에서 계속되는 문장을 Key 하기 위해서는 화면 [다 - 3]과 같은 방법을 쓴다.

[다 - 3]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 1
COMMAND ==> SCROLL ==> F
***** TOP OF DATA *****
I00001 //Z90CDSAS JOB CLASS=B,MSGCLASS=X
***** BOTTOM OF DATA *****
```

화면 [다 - 3]에서는 line command I(Insert)를 이용해서 한줄한줄 문
을 Key in할 공백 line을 생성시키는 방법을 보여준다. **ENTER** Key
를 누르면 화면 [다 - 4]와 같이된다.

[다 - 4]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> PAGE
***** TOP OF DATA *****
000001 //Z90CDSAS JOB CLASS=B,MSGCLASS=X
*****
***** BOTTOM OF DATA *****
```

화면 [다 - 4]에서는 공백 line이 생성된 것을 보여준다. 이렇게 하여 JCL문이 화면 [다 - 5]와 같이 완성된다. JCL문에 대해서 다음 화면에서 설명한다.

[다 - 5]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
COMMAND ==>                                SCROLL ==> PAGE
***** TOP OF DATA *****
000001 //Z90CDSIN JOB CLASS=C,MSGCLASS=X
000002 //STEP1 EXEC SAS
000003 //SYSIN DD *
***** BOTTOM OF DATA *****
```

화면 [다 - 5]에 나와 있는 3줄은 SAS Batch 작업에 필요한 JCL(Job Control Language : 작업제어언어)문 인데 이것은 이용자가 작업을 할시에는 반드시 명시해주어야 한다. JCL 작성이 끝난후 다 SAS PROGRAM을 화면 [라 - 1]과 같이 Key in한다.

라. 작업수행

[라-1]

```

EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
COMMAND ==) SUB                               SCROLL ==) PAGE
***** ***** TOP OF DATA *****
000001 //Z90CDSIN JOB CLASS=B,MSGCLASS=X
000002 //STEP1 EXEC SAS
000003 //SYSIN DD *
000004 DATA A;
000005 INPUT ENGLISH MATH SCIENCE NAME $ SEX $ ;
000006 CARDS;
000007 86.7 98.4 87.6 BOKHAM MALE
000008 78.5 89.6 98.6 CHULSU MALE
000009 98.4 76.7 87.4 YONHEE FEMALE
000010 89.5 95.4 96.4 SOOKJA FEMALE
000011 85.3 79.5 92.1 JUNGU MALE
000012 85.4 92.1 93.7 MINHEE FEMALE
000013 92.4 98.3 91.0 YONGGU MALE
000014 ;
000015 PROC PRINT ;
000016 PROC MEANS ;
000017 TITLE ' AVERAGE SCORE OF CLASS ' ;
000018 /*
000019 //
***** ***** BOTTOM OF DATA *****

```

화면 [라-1]에 있는 SAS program 문은 평균계산에서 다른 예제를 그대로 실어놓았다. 이렇게 SAS program이 완성되면 작업을 수행시켜야 하는데 그것은 'COMMAND ==) SUB'란에 SUBMIT의 앞자만 딴 'SUB'를 Key in 하고 ENTER Key를 누르고 잠시 기다리면 화면 [라-2]과 같이 된다.

[라-2]

```

EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
COMMAND ==) SUB                               SCROLL ==) PAGE
***** ***** TOP OF DATA *****
000001 //Z90CDSIN JOB CLASS=B,MSGCLASS=X
000002 //STEP1 EXEC SAS
000003 //SYSIN DD *
000004 DATA A;
000005 INPUT ENGLISH MATH SCIENCE NAME $ SEX $ ;
000006 CARDS;
000007 86.7 98.4 87.6 BOKHAM MALE
000008 78.5 89.6 98.6 CHULSU MALE
000009 98.4 76.7 87.4 YONHEE FEMALE
000010 89.5 95.4 96.4 SOOKJA FEMALE
000011 85.3 79.5 92.1 JUNGU MALE
000012 85.4 92.1 93.7 MINHEE FEMALE
000013 92.4 98.3 91.0 YONGGU MALE
000014 ;
000015 PROC PRINT ;
000016 PROC MEANS ;
000017 TITLE ' AVERAGE SCORE OF CLASS ' ;
IKJ56250I JOB Z90CDSIN(JOB05981) SUBMITTED
***

```

화면 [라-2]의 맨 마지막줄의 '***'이 나오면 이용자가 보낸 SAS program이 제대로 전달이 되었음을 의미한다. 이것을 확인한 후 다시 ENTER Key를 누르면 화면 [라-3]로 원상복귀된다.

[라 - 3]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
COMMAND ===) SCROLL ===) PAGE
***** TOP OF DATA *****
000001 //Z90C0SIN JOB CLASS=B,MSGCLASS=X
000002 //STEP1 EXEC SAS
000003 //SYSIN DD *
000004 DATA A;
000005 INPUT ENGLISH MATH SCIENCE NAME $ SEX $ ;
000006 CARDS;
000007 86.7 98.4 87.6 BOKMAH MALE
000008 78.5 89.6 98.6 CHULSU MALE
000009 98.4 76.7 87.4 YONHEE FEMALE
000010 89.5 95.4 96.4 SOKJA FEMALE
000011 85.3 79.5 92.1 JUNGU MALE
000012 85.4 92.1 93.7 MINHEE FEMALE
000013 92.4 98.3 91.0 YONGGU MALE
000014
000015 PROC PRINT ;
000016 PROC MEANS ;
000017 TITLE ' AVERAGE SCORE OF CLASS ' ;
000018 /*
000019 //
***** BOTTOM OF DATA *****
```

화면 [라 - 3]의 상태에서 이용자가 <가. 작업순서> 흐름의 순서를 따른다면 작업시킨 결과를 보고자 할 것이다. 이때 화면 [라 - 3] 상태에서 [PF2] key를 누름으로써 화면을 양분(split)시킨다. 그러면 화면 [라 - 4]이 나타난다.

[라 - 4]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ===) 8
0 ISPF PARMS - SPECIFY TERMINAL AND USER PARAMETERS USERID - Z90
1 BROWSE - DISPLAY SOURCE DATA OR OUTPUT LISTINGS TIME - 09:19
2 EDIT - CREATE OR CHANGE SOURCE DATA TERMINAL - 3278
3 UTILITIES - PERFORM UTILITY FUNCTIONS PF KEYS - 24
4 FOREGROUND - INVOKE LANGUAGE PROCESSORS IN FOREGROUND
5 BATCH - SUBMIT JOB FOR LANGUAGE PROCESSING
6 COMMAND - ENTER TSO COMMAND OR CLIST
7 DIALOG TEST - PERFORM DIALOG TESTING
8 SDSF - SPOOL DISPLAY AND SEARCH FACILITY
C CHANGES - DISPLAY SUMMARY OF CHANGES FOR THIS RELEASE
T TUTORIAL - DISPLAY INFORMATION ABOUT ISPF/PDF
X EXIT - TERMINATE ISPF USING LOG AND LIST DEFAULTS

ENTER END COMMAND TO TERMINATE ISPF.
```

화면 [라 - 4]에서 작업결과를 확인하기 위해 'OPTION ===>'란에 '8'을 Key in하고 ENTER Key를 누르면 화면 [라 - 5]가 나타난다.

[라- 5]

```

EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
----- SDSF PRIMARY OPTION MENU -----
COMMAND INPUT ==> H                                SCROLL ==> PAGE

DA      - DISPLAY ACTIVE USERS OF SYSTEM
I       - DISPLAY JOBS IN THE JES2 INPUT QUEUE
O       - DISPLAY JOBS IN THE JES2 OUTPUT QUEUE
H       - DISPLAY JOBS IN THE JES2 TSO HELD OUTPUT QUEUE
WHO     - DISPLAY USER ID, LOGON PROC NAME, AND TERMINAL NAME

END     - TO TERMINATE SDSF

USE HELP KEY FOR SDSF INFORMATION
    
```

화면 [라- 5]에서 결과를 보기 위해서는 ‘H’를 ‘COMMAND INPUT==>’란에 Key in하면 화면 [라- 6]이 나타난다.

[라- 6]

```

EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
SDSF HELD OUTPUT DISPLAY ALL CLASSES          164 LINES DATA SET DISPLAYED
COMMAND INPUT ==>                               SCROLL ==> PAGE
NP C JOBNAM TYPE JNUM DN CRDATE C FORM FCB RMT TOT REC RNUM PROGRAMMER NAME
S   Z90COSIN JOB 5982  5 85.358 X STD ****          164
    
```

화면 [라- 6]을 보면 164 Record 짜리의 결과가 나와있다는 것을 확인할 수 있고 이의 내용을 보기위해 ‘NP’란에 Select의 약자인 ‘S’를 Key in 하고 **ENTER** Key를 누르면 화면 [라- 7]이 나온다.

[라-7]

```

EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
SDSF OUTPUT DISPLAY Z90COSIN JOB 5982 2 LINE 0 COLUMNS 2 81
COMMAND INPUT ==> SCROLL ==> PAGE
***** TOP OF DATA *****
JES2 JOB LOG -- SYSTEM BOS -- NO

09.22.17 JOB 5982 $HASP373 Z90COSIN STARTED - INIT 2 - CLASS B - SYS BOS
09.22.31 JOB 5982 -
09.22.31 JOB 5982 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CLOCK
09.22.31 JOB 5982 -Z90COSIN STEP1 SAS 00 402 .01 .00 .2
09.22.32 JOB 5982 -Z90COSIN ENDED. NAME- TOTAL CPU TIME=
09.22.32 JOB 5982 $HASP395 Z90COSIN ENDED
----- JES2 JOB STATISTICS -----
24 DEC 85 JOB EXECUTION DATE
18 CARDS READ
164 SYSOUT PRINT RECORDS
0 SYSOUT PUNCH RECORDS
9,056 SYSOUT SPOOL BYTES
0.25 MINUTES EXECUTION TIME
1 //Z90COSIN JOB CLASS=B,MSGCLASS=X
2 //STEP1 EXEC SAS
3 XKSAS PROC ENTRY=SAS,OPTIONS=,SORT=20
*****

```

화면 [라-7]이 나오면 결과를 보기위해 [PF8] Key 계속눌러가면 화면 [라-8]이 나오게 된다.

[라-8]

```

EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
SDSF OUTPUT DISPLAY Z90COSIN JOB 5982 103 LINE 1 COLUMNS 2 81
COMMAND INPUT ==> SCROLL ==> PAGE
SAS

OBS ENGLISH MATH SCIENCE NAM
1 86.7 98.4 87.6 BOKM
2 78.5 89.6 98.6 CHUL
3 98.4 76.7 87.4 YONH
4 89.5 95.4 96.4 SOOK
5 85.3 79.5 92.1 JUNG
6 85.4 92.1 93.7 MINH
7 92.4 98.3 91.0 YONG
AVERAGE SCORE OF CLASS

VARIABLE N MEAN STANDARD DEVIATION MINIMUM VALUE MAXIMUM VALUE
ENGLISH 7 88.02857143 6.26836350 78.50000000 98.40000000
MATH 7 90.00000000 8.75899538 76.70000000 98.40000000
SCIENCE 7 92.40000000 4.20753293 87.40000000 98.60000000
***** BOTTOM OF DATA *****

```

화면 [라-8]을 보면 이용자가 원하는 정보를 취득할 수 있게 된다. 이렇게 정보를 취한후 다시 원상태로 돌아가기위해 [PF3] Key 를 누르면 그 결과 화면 [라-9]가 나타난다.

[라-9]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
SDSF HELD OUTPUT DISPLAY ALL CLASSES          164 LINES DATA SET DISPLAYED
COMMAND INPUT ===)                          SCROLL ===) PAGE
MP C JOBNAM TYPE JNUM DN CRDATE C FORM FCB RMT TOT REC RNUM PROGRAMMER NAME
Z90COSIN JOB 5982 5 85.358 X STD ***** 164
```

화면 [라-9]가 온후 다시 PROGRAM으로 돌아가고자 할때 PF9

Key를 누르면 그결과 화면 [라-10]이 나타난다.

[라-10]

```
EDIT --- Z90.PROGRAM.LIBRARY(SASTEST) ----- COLUMNS 001 072
COMMAND ===)                          SCROLL ===) PAGE
***** TOP OF DATA *****
000001 //Z90COSIN JOB CLASS=B,MSGCLASS=X
000002 //STEP1 EXEC SAS
000003 //SYSIN DD *
000004 DATA A;
000005 INPUT ENGLISH MATH SCIENCE NAME * SEX * ;
000006 CARDS;
000007 86.7 98.4 87.6 BOKHAM MALE
000008 78.5 89.6 98.6 CHULSU MALE
000009 98.4 76.7 87.4 YONHEE FEMALE
000010 89.5 95.4 96.4 SOKJA FEMALE
000011 85.3 79.5 92.1 JUNGU MALE
000012 85.4 92.1 93.7 MINHEE FEMALE
000013 92.4 98.3 91.0 YONGGU MALE
000014
000015 PROC PRINT ;
000016 PROC MEANS ;
000017 TITLE ' AVERAGE SCORE OF CLASS ' ;
000018 /*
000019 //
***** BOTTOM OF DATA *****
```

화면 [라-10]과 같이 원래 program이 나오게 되고 이 program을 보관 시키고자 할때 PF3 Key를 누르면 되는데 이때 결과 화면 [라-11]이 나온다.

[라 - 11]

```
----- EDIT - ENTRY PANEL ----- MEMBER SASTEST SAVED
COMMAND ==>
ISPF LIBRARY:
PROJECT ==> Z90
LIBRARY ==> PROGRAM ==> ==> ==>
TYPE ==> LIBRARY
MEMBER ==> (BLANK FOR MEMBER SELECTION LIST)
OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==> (IF NOT CATALOGED)
DATA SET PASSWORD ==> (IF PASSWORD PROTECTED)
PROFILE NAME ==> (BLANK DEFAULTS TO DATA SET TYPE)
SDSF HELD OUTPUT DISPLAY ALL CLASSES 0 LINES COMMAND ISSUED
```

화면 [라 - 11] 이 나오면 맨 우측상단의 'MEMBER SASTEST SAVED' 라는 Message 를 확인함으로써 제대로 보관이 된것을 알 수 있게 된다.

마. 작업종료

[마 - 1]

```
----- EDIT - ENTRY PANEL -----
COMMAND ==>
ISPF LIBRARY:
PROJECT ==> Z90
LIBRARY ==> PROGRAM ==> ==> ==>
TYPE ==> LIBRARY
MEMBER ==> =X (BLANK FOR MEMBER SELECTION LIST)
OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==> (IF NOT CATALOGED)
DATA SET PASSWORD ==> (IF PASSWORD PROTECTED)
PROFILE NAME ==> (BLANK DEFAULTS TO DATA SET TYPE)
SDSF HELD OUTPUT DISPLAY ALL CLASSES 0 LINES LINE 0 TO 0 OF 0
```

화면 [마 - 1]에서는 작업종료의 방법을 보여주는데 이를 위해서는 'MEMBER ==>'란에 '=X'를 Key in 하고 **ENTER** Key 를 누르면 화면 [마 - 2]가 나오게 된다.

[마 - 2]

```
SDSF HELD OUTPUT DISPLAY ALL CLASSES          0 LINES LINE 0 TO 0 OF 0
COMMAND INPUT ==> =X                          SCROLL ==> PAGE
NP C JOBNAM TYPE JNUM DN CRDATE C FORM FCB RMT TOT REC RNUM PROGRAMMER NAME
```

화면 [마 - 2]에서 'COMMAND INPUT ==>' 란에 '=X'를 Key in하고,
ENTER Key 를 누르면 화면 [마 - 3]이 나온다.

[마 - 3]

```
SPECIFY DISPOSITION OF LOG DATA SET -----
COMMAND ==> .
LOG OPTIONS FOR THIS SESSION          NO LIST DATA SET THIS SESSION
-----
PROCESS OPTION ==> D
SYSOUT CLASS ==> A
LOCAL PRINTER ID ==>

VALID PROCESS OPTIONS:
J - SUBMIT JOB TO PRINT (AND DELETE)      K - KEEP DATA SET (DO NOT PRINT)
L - ROUTE TO LOCAL PRINTER (AND DELETE)  D - DELETE DATA SET (DO NOT PRINT)

INSTRUCTIONS:
PRESS ENTER KEY TO COMPLETE ISPF TERMINATION.
ENTER END COMMAND TO RETURN TO THE PRIMARY OPTION MENU.

JOB STATEMENT INFORMATION: (IF OPTION 'J' SELECTED)
==> //Z90A      JOB (ACCOUNT), 'NAME'
==> /**
==> /**
==> /**
```

화면 [마 - 3]에서 'PROCESS OPTION ==>' 란에 'D'를 Key in하고,
ENTER Key 를 누르면 화면 [마 - 4]가 나온다.

[마 - 4]

```
Z90.SPFLOG1.LIST HAS BEEN DELETED.  
READY  
LOGOFF
```

화면 [마 - 5]에서 'LOGOFF'를 READY 바로 아래에 Key in하고, **ENTER** Key를 누르면 작업을 종료하게 된다.

이상이 Batch방식 SAS 작업의 흐름이다.

4. 분석기능별 SAS 명령어

가. 기술통계분석

(1) 평균 (MEANS)

```
PROC MEANS [DATA=SASdataset NOPRINT MAXDEC=n  
VARDEF=DF|WEIGHT|WGT|N|WDF N NMIS  
MEAN STD MIN MAX RANGE SUM VAR USS CSS  
STDERR CV SKEWNESS KURTOSIS T PRT SUMWGT];
```

```
[VAR variables;]
```

```
[BY variables;]
```

```
[FREQ variable;]
```

```
[WEIGHT variable;]
```

```
[ID variables;]
```

```
[OUTPUT [OUT=SASdataset keyword=names];]
```

```
where keyword is chosen from N NMIS MEAN STD MIN  
MAX RANGE SUM VAR USS CSS STDERR CV  
SKEWNESS KURTOSIS T PRT SUMWGT
```

수치변수에 대한 평균 및 분산등의 기술통계치를 구할 때 사용

(2) 상관분석 (CORR)

```
PROC CORR [DATA=SASdataset OUTP=SASdataset  
OUTS=SASdataset OUTK=SASdataset  
OUTH=SASdataset PEARSON SPEARMAN  
KENDALL Hoeffding RANK BEST=n  
VARDEF=DF|WGT|WEIGHT|N|WDF  
NOSIMPLE NOPRINT NOPROB NOMISS SSCP  
COV NOCORR];
```

```
[VAR variables;]
```

```
[WITH variables;]
```

```
[WEIGHT variable;]
```

```
[FREQ variable;]
```

```
[BY variables;]
```

수치변수 및 순위변수에 대한 상관계수 및 그 유의성을 구하고자할 때 사용

(3) 횡수분포표 (FREQ)

```
PROC FREQ [DATA=SASdataset ORDER=FREQ|  
DATA|INTERNAL|FORMATTED  
FORMCHAR(1,2,7)='string'];  
TABLES requests [/ MISSING LIST OUT=SASdataset  
CHISQ EXPECTED DEVIATION CELLCHI2  
CUMCOL MISSPRINT SPARSE NOFREQ  
NOPERCENT NOROW NOCOL NOCUM NOPRINT];
```

```
[WEIGHT variable;]
```

```
[BY variables;]
```

횡수분포표 작성시 사용

(4) Summary 통계치 (SUMMARY)

```
PROC SUMMARY [DATA=SASdataset MISSING
              NWAY IDMIN DESCENDING ORDER=FREQ |
              DATA |INTERNAL |EXTERNAL |FORMATTED
              VARDEF=DF |WEIGHT |WGT |N |WDF];
  [CLASS |CLASSES variables;]
  VAR variables;
  [BY variables;]
  [FREQ variable;]
  [WEIGHT variable;]
  [ID variables;]
  OUTPUT [OUT=SASdataset] keyword
         [(variables)]=[names]...;
where keyword is chosen from N NMISS MEAN
      STD MIN MAX RANGE SUM VAR USS
      CSS CV STDERR T PRT SUMWGT
```

유목별로 구분될 수 있는 데이터셋들에 대한 단순통계치를 차후분 석에 이용시 주로 사용

(5) 모든 기술 통계치 (UNIVARIATE)

```
PROC UNIVARIATE [DATA=SASdataset NOPRINT
                PLOT FREQ NORMAL PCTLDEF=value
                VARDEF=DF |WGT |WEIGHT |N |WDF];
  [VAR variables;]
  [BY variables;]
  [FREQ variable;]
  [WEIGHT variable;]
  [ID variables;]
  [OUTPUT OUT=SASdataset keyword=names ...;]
where keyword is chosen from N NMISS NOBS MEAN
      SUM STD VAR SKEWNESS KURTOSIS SUMWGT
      MAX MIN RANGE Q3 MEDIAN Q1 QRANGE P1 P5
      P10 P90 P95 P99 MODE SIGNRANK NORMAL
```

수치변수에 대한 모든 가능한 기술통계치를 구하고자 할 때 사용

(6) 막대도표, 블록도표, 원도표 및 별도표 (CHART)

```
PROC CHART [DATA = SASdataset LPI = p];
  [VBAR variables [/ any options in the option list,
    below, and LEVELS = n SYMBOL = 'char'
    GROUP = variable SUBGROUP = variable
    NOSYMBOL NOZEROS G100 ASCENDING
    DESCENDING REF = value NOSPAC];]
  [HBAR variables [/ any option in the option list, below,
    and LEVELS = n SYMBOL = 'char'
    GROUP = variable SUBGROUP = variable
    NOSYMBOL NOZEROS G100 ASCENDING
    DESCENDING REF = value NOSTAT FREQ CFREQ
    PERCENT CPERCENT SUM MEAN];]
  [BLOCK variables [/ any option in the option list,
    below, and LEVELS = n SYMBOL = 'char'
    GROUP = variable SUBGROUP = variable
    NOSYMBOL NOZEROS G100];]
  [PIE variables [/ any option in the option list, below];]
  [STAR variables [/ any option in the option list,
    below];]
  [BY variables;]
option list: MISSING DISCRETE TYPE = FREQ
  TYPE = PERCENT | PCT TYPE = CFREQ
  TYPE = CPERCENT | CPCT TYPE = SUM TYPE = MEAN
  SUMVAR = variable MIDPOINTS = values
  FREQ = variable AXIS = value
```

여러 가지 도표 작성에 사용

(7) 변수간의 점도표 (PLOT)

```
PROC PLOT [DATA = SASdataset UNIFORM NOLEGEND];
  PLOT vertical*horizontal | vertical*horizontal = 'character' |
    vertical*horizontal = variable ... [/ VAXIS = values
    HAXIS = values VZERO HZERO VREVERSE
    VREF = values VREFCHAR = 'c' HREF = values
    HREFCHAR = 'c' VPOS = n HPOS = n
    YSPACE = n HSPACE = n OVERLAY
    CONTOUR = value S1 = value S2 = value];
  [BY variables;]
```

점도표를 그려 보고자 할 때 사용

(8) 시계열 자료 PLOTTING (TIME PLOT)

```
PROC TIMEPLOT [UNIFORM MAXDEC=n];
  PLOT variable | variable='symbol' |
    (variables)='symbol' ... [/ OVERLAY HILOC
  JOINREF REVERSE POS=n AXIS=specification
  REF=values REFCHAR=charactervalue
  OVPCHAR=charactervalue NOSYMMNAME];
  [CLASS variables;]
  [ID variables;]
  [BY variables;]
```

시계열 자료에 대한 점도표 작성시 사용 (시계열수가 많을 때)

(9) 기술통계표 작성 (TABULATE)

```
PROC TABULATE [DATA=SASdataset MISSING
  FORMAT=format ORDER=FREQ | DATA | INTERNAL |
  FORMATTED FORMCHAR[(indexlist)='string'];
  [CLASS variables;]
  [VAR variables;]
  [FREQ variable;]
  [WEIGHT variable;]
  [FORMAT variables format ...;]
  [LABEL variable='label' ...;]
  [BY variables;]
  TABLE [expression,] [expression,]expression
    [/ MISSTEXT='text' FUZZ=nnn RTSPACE |
    RTS=n BOX=PAGE |variablename |'string'
    ROW=FLOAT |CONSTANT |CONST CONDENSE];
  [KEYLABEL keyword='text' ...;]
  where keyword is chosen from N NMISS MEAN
  STD MIN MAX RANGE SUM USS CSS STDERR
  CV T PRT VAR SUMWGT PCTN PCTSUM
```

관심있는 Data Set에 대한 Table 작성시 사용 (※TPL 대신 사용가능)

(10) 순위 작성 (RANK)

```
PROC RANK [DATA=SASdataset OUT=SASdataset
  TIES=MEAN |HIGH |LOW DESCENDING
  GROUPS=n FRACTION |F PERCENT |
  P NORMAL=BLOM |TUKEY |VW SAVAGE];
  [VAR variables;]
  [RANKS names;]
  [BY variables;]
```

SAS Data set의 관측치에 대한 한개 또는 그 이상의 수치변수를 이용해
서 순위값을 정해준다.

(1) 표준화점수 작성 (STANDARD)

```
PROC STANDARD [DATA=SASdataset OUT=SASdataset  
VARDEF=DF|WEIGHT|WGT|N|WDF  
MEAN|M=m STD|S=s REPLACE];  
[VAR variables];  
[FREQ variable];  
[WEIGHT variable];  
[BY variables];
```

주어진 평균과 분산을 가지고 SAS data set 내의 한개 또는 그 이상의 변수에 대해 표준화점수를 구하여 새로운 표준화점수 data set 을 만든다.

(2) 두 집단간 평균치차 검증 (TTEST)

```
PROC TTEST [DATA=SASdataset];  
CLASS variable;  
[VAR variables];  
[BY variables];
```

2 개의 관측치 집단의 모분산이 같다는 가정과 그렇지 않은 경우의 귀무가설 “ 두 집단간의 평균이 같다 ”에 대한 t 통계량을 산출하여 줌으로써 집단간 평균치차의 검정을 가능케 한다.

(3) BMDP를 이용한 통계분석 (BMDP)

```
PROC BMDP [DATA=SASdataset PROG=BMDPnn  
UNIT=n CODE=savefile CONTENT=DATA |  
CORR|MEAN|FREQ LABEL=variable  
LABEL2=variable NOMISS WRKSPCE|PARAM=nn];  
[VAR variables];  
[BY variables];  
PARMCARDS;  
BMDP control statements  
;
```

BMDP의 Module들을 SAS Session에서 불러와 이용시 사용

나. 회귀분석기법

(1) 회귀분석 (PROC REG)

```
PROC REG [DATA=SASdataset OUTEST=SASdataset
OUTSSCP=SASdataset NOPRINT SIMPLE
USSCP ALL COVOUT SINGULAR=n];
[label:] MODEL dependents=regressors [/ NOPRINT
NOINT ALL XPX I SS1 SS2 STB TOL VIF COVB
CORRB SEQB COLLIN COLLINOINT ACOV
SPEC PCORR1 PCORR2 SCORR1 SCORR2
P R CLM CLI DW INFLUENCE PARTIAL];
[VAR variables;]
[FREQ variable;]
[WEIGHT variable;]
[ID variable;]
[OUTPUT {OUT=SASdataset PREDICTED |P=names
RESIDUAL |R=names L95M=names U95M=names
L95=names U95=names STDP=names
STDR=names STDI=names STUDENT=names
COOKD=names H=names PRESS=names
RSTUDENT=names DFFITS=names
COVRATIO=names};]
[RESTRICT equation, ...;]
[label: TEST equation, ... [/ PRINT];]
[label: MTEST equation, ... [/ PRINT CANPRINT
DETAILS];]
[BY variables;]
```

제반 선형 회귀모형의 모수에 대한 추정치 및 모형 자체의 적합도, 예측치의 신뢰구간 등을 산출하여 준다.

(2) 모든 가능한 회귀모형에 대한 규명 (PROC RSQUARE)

```
PROC RSQUARE [DATA=SASdataset SIMPLE]
S CORR | C NOINT NOPRINT OUTEST=SASdataset;
[label:] MODEL dependents=independents
[/ SELECT=n INCLUDE=i START=n
STOP=n SIGMA=n ADJR SQ AIC BIC
CP GMSEP JP MSE PC RMSE SBC SP SSE B];
[FREQ variable;]
[WEIGHT variable;]
[BY variables;]
```

여러개의 독립변수의 존재시 가능한 모든 독립변수조합에 대한 결정 계수값 및 독립변수조합 list 를 산출하여 준다.

(3) 단계별 회귀분석 (PROC STEPWISE)

```
PROC STEPWISE [DATA=SASdataset];
MODEL dependents=independents [/ NOINT
FORWARD | F BACKWARD | B STEPWISE MAXR
MINR SLENTRY | SLE=value SLSTAY | SLS=value
INCLUDE=n*START=s STOP=s DETAILS];
[WEIGHT variable;]
[BY variables;]
```

여러개의 독립변수의 존재시 이중 가장 최적의 모형을 산출하는 독립변수들로 구성된 회귀모형을 추출해 준다.

(4) 고장시간의 회귀모형 규명 (PROC LIFEREG)

```
PROC LIFEREG [DATA=SASdataset OUTEST=SASdataset
COVOUT NOPRINT];
[CLASS variables;]
[label:] MODEL variable[*censor(n ...)] = variables
[/ DISTRIBUTION | DIST | D = WEIBULL |
EXPONENTIAL | LNORMAL | LLOGISTIC | GAMMA
NOLOG COVB CORRB NOINT
INTERCEPT | INTERCPT = number NOSCALE
SCALE = number
NOSHAPE1 SHAPE1 = number INITIAL = numbers
MAXIT = number ITPRINT CONVERGE = number
SINGULAR = number];
[OUTPUT {OUT=SASdataset QUANTILES | QUANTILE |
Q = quantiles CONTROL = variable
PREDICTED | P = name XBETA = name
STD_ERR | STD = name SURVIVAL = name
CENSORED = name};]
[BY variables;]
```

고장시간 자료에 대한 모수회귀모형을 추정하여 준다.

(5) 라이프 테스트 (PROC LIFETEST)

```
PROC LIFETEST [DATA = SASdataset METHOD = PL |
  KM | LT | LIFE | ACT NOPRINT NOTABLE
  MISSING OUTSURV | OUTS = SASdataset
  OUTTEST | OUTT = SASdataset PLOTS = (plots)
  MAXTIME = number INTERVALS = numbers
  NINTERVAL = number WIDTH = number
  ALPHA = number SINGULAR = number];
[TIME timevariable[*censor(n,...)];
[STRATA variable[(ranges)] ...];
[TEST variables];
[ID variables];
[FREQ variable];
[BY variables];
```

라이프 테스트 분석을 수행해 준다.

(6) 비선형 회귀분석 (PROC NLIN)

```
PROC NLIN [DATA = SASdataset OUTEST = SASdataset
  BEST = n PLOT METHOD = GAUSS | MARQUARDT |
  GRADIENT | DUD NOHALVE SIGSQ = value
  G4 G4SINGULAR TAU = value RHO = value
  EFORMAT MAXITER = i CONVERGE = c];
PARAMETERS | PARS parameter = value | sequence, ...;
[BOUNDS expression, ...];
other programming statements
MODEL dependent = expression;
[DER.parameter = expression];
[_WEIGHT_ = expression];
[OUTPUT OUT = SASdataset [PREDICTED | P = variable
  RESIDUAL | R = variable L95M = variable
  U95M = variable L95 = variable
  U95 = variable STDP = variable
  STDR = variable STUDENT = variable
  H = variable PARS = variables
  SSE | ESS = variable];
[ID variables];
```

비선형 회귀모형의 모수에 대한 최소자승 또는 가중 최소자승 추정치를 산출하여 준다.

(7) 반응표면 회귀분석 (PROC RSREG)

```
PROC RSREG [DATA=SASdataset  
OUT=SASdataset NOPRINT];  
MODEL response=independents [/ LACKFIT  
NOOPTIMAL|NOOPT COVARIATES|COVAR=n  
ACTUAL PREDICT RESIDUAL U95M L95M  
U95 L95 D BYOUT];  
[WEIGHT variable;]  
[ID variables;]  
[BY variables;]
```

완비이차 반응표면의 추정치를 산출하고 모형내의 요인들에 대한 반응치를 최적화 하기 위한 임계치를 구하여 준다.

(8) Probit 모형 추정 (PROC PROBIT)

```
PROC PROBIT [DATA=SASdataset OPTC|C=rate  
HPROB=p LOG|LN LOG10];  
VAR dose subjects response;  
[BY variables;]
```

Probit모형을 갖는 자료에 대해 절편, 기울기, 자연반응율의 최우 추정치를 구하여 준다.

다. 분산분석기법

(1) 분산분석 (PROC ANOVA)

```
PROC ANOVA [DATA=SASdataset];
  [CLASS variables;]
  [MODEL dependents=effects[/ NOUNI INIT |
    INTERCEPT];]
  [MEANS effects [/ BON DUNCAN GABRIEL
    REGWF REGWQ SCHEFFE SIDAK SMM | GT2 SNK
    T | LSD TUKEY ALPHA=p WALLER
    KRATIO=value LINES CLDIFF E=effect];]
  [ABSORB variables;]
  [FREQ variable;]
  [TEST H=effects E=effect;]
  [MANOVA [H=effects E=effect M=equation, ...
    [MNames=names PREFIX=name]] [/ PRINTH
    PRINTE ORTH SHORT CANONICAL SUMMARY];]
  [REPEATED factorname levels (levelvalues)
    [CONTRAST [(ordinalreferencelevel)] |
    POLYNOMIAL | HELMERT | MEAN
    [(ordinalreferencelevel)] | PROFILE], ...]
  [/ NOM NOU PRINTM PRINTH PRINTE
    PRINTRV SHORT SUMMARY CANONICAL];]
  [BY variables;]
```

여러가지 기법의 실험계획에 의해 얻어진 균형된 자료에 대해 분산 분석을 수행하며 중간에 얻어진 여러 통계량을 출력시켜 준다.

(2) 일반선형모형분석 (PROC GLM)

```
PROC GLM [DATA=SASdataset
  ORDER=FREQ | DATA | INTERNAL | FORMATTED];
  [CLASS | CLASSES variables;]
  MODEL dependents=independents [/ NOINT INT |
    INTERCEPT NOUNI SOLUTION TOLERANCE
    E E1 E2 E3 E4 SS1 SS2 SS3 SS4 P CLM
    CLI ALPHA=p XPX INVERSE | I
    SINGULAR=value ZETA=value];
  [CONTRAST 'label' [INTERCEPT value] effect values ...
    [/ E E=effect ETYPE=n SINGULAR=number];]
  [ESTIMATE 'label' [INTERCEPT value] effect values ...
    [/ E DIVISOR=number SINGULAR=number];]
  [LSMEANS effects [/ E STDERR PDIFF E=effect
    ETYPE=n SINGULAR=number];]
  [MANOVA [H=effects E=effect
    M=equation, ... [MNames=names
    PREFIX=name]] [/ PRINTH PRINTE
    HTYPE=n ETYPE=n ORTH SHORT
    CANONICAL SUMMARY];]
  [OUTPUT [OUT=SASdataset PREDICTED | P=variables
    RESIDUAL | R=variables];]
  [RANDOM effects [/ Q];]
  [REPEATED factorname levels (levelvalues)
    [CONTRAST [(ordinalreferencelevel)] |
```



```

POLYNOMIAL | HELMERT | MEAN
[(ordinalreferencelevel)]
PROFILE[, ...] [/ NOM NOU PRINTM PRINTH
PRINTI PRINTRV SHORT SUMMARY CANONICAL
HTYPE=n];
[TEST H=effects E=effect [/ HTYPE=n ETYPE=n];]
[ABSORB variables;]
[BY variables;]
[FREQ variable;]
[ID variables;]
[MEANS effects [/ DEONLY BON DUNCAN
GABRIEL REGWF REGWQ SCHEFFE SIDAK SMM |
GT2 SNK T | LSD TUKEY ALPHA=p
WALLER KRATIO=value LINES CLDIFF
NOSORT E=effect ETYPE=n HTYPE=n];]
[WEIGHT variable;]

```

PROC ANOVA가 균형된 자료(누락치가 없는 자료)에 대한 분산분석을 수행할 수밖에 없으나 여기서는 누락치가 있을 수 있는 여러 가지 실험설계에 의해 얻어질 수 있는 자료에 대해 분산분석을 수행해주며 보다 포괄적으로 통계모형에서 선형성을 갖는 모형은 모두가 PROC으로 분석을 행할 수 있다.

(3) 실험설계 (PROC PLAN)

```

PROC PLAN [SEED=number];
FACTORS name=[m OF] n [ORDERED] ...;

```

실험계획법중 난괴법(randomized design)을 사용할시 필요한 요인 배치계획을 생성해 준다.

(4) Nested 분산분석 (PROC NESTED)

```

PROC NESTED [DATA=SASdataset AOV];
CLASS variables;
[VAR variables;]
[BY variables;]

```

Nested 구조를 갖는 실험계획에 의해 얻어진 자료에 대해 공분산분석 및 분산분석을 수행한다.

(5) 분산인자추정 (PROC VARCOMP)

```
PROC VARCOMP [METHOD=TYPE1 | METHOD=MIVQUE0  
METHOD=ML | METHOD=REML MAXITER=n  
EPSILON=n DATA=SASdataset];  
CLASS variables;  
MODEL dependents=effects [/ FIXED=n];  
[BY variables;
```

일반선형모형에 있어서 분산인자의 추정치를 계산해준다.

라. 요인분석기법

(1) 요인분석 (FACTOR)

```
PROC FACTOR [DATA = SASdataset TARGET = SASdataset
OUT = SASdataset OUTSTAT = SASdataset
METHOD | M = PRINCIPAL | PRIN | P METHOD |
M = PRINIT METHOD | M = ULS | U METHOD | M = ALPHA |
A METHOD | M = ML | M METHOD | M = HARRIS |
H METHOD | M = IMAGE | I METHOD |
M = PATTERN METHOD | M = SCORE PRIORS = ONE |
O PRIORS = MAX | M PRIORS = SMC | S
PRIORS = ASMC | A PRIORS = INPUT |
I PRIORS = RANDOM | R COVARIANCE |
COV WEIGHT MAXITER = n CONVERGE | CONV = n
NFACTORS | NFACT | N = n PROPORTION |
PERCENT | P = n MINEIGEN | MIN = n HEYWOOD |
HEY ULTRAHEYWOOD | ULTRA ROTATE | R = VARIMAX |
V ROTATE | R = QUARTIMAX | Q ROTATE | R = EQUAMAX |
E ROTATE | R = ORTHOMAX ROTATE | R = HK
ROTATE | R = PROMAX | P ROTATE |
R = PROCRUSTES ROTATE | R = NONE | N
GAMMA = n HKPOWER | HKP = n POWER = n
PREROTATE | PRE = name NORM = name SIMPLE |
S CORR | C MSA SCREE EIGENVECTORS |
EV PRINT RESIDUALS | RES PREPLOT PLOT
N PLOT = n SCORE ALL REORDER |
RE ROUND FLAG = n FUZZ = n
NOINT NOCORR SINGULAR | SING = p];
[PRIORS communalities;]
[VAR variables;]
[PARTIAL variables;]
[FREQ variable;]
[WEIGHT variable;]
[BY variables;]
```

여러가지의 공통요인 및 인자분석을 수행해 준다.

인자추출법으로는 주성분분석, 주요인분석, 반복주요인분석, 비가중치소자
승요인분석, 정준요인분석, 알파요인분석, 이미지 (image) 성분분석, 해
리스 성분분석이 가능하며 회전방법으로는 배리맥스, 쿼티맥스, 이쿼맥
스, 오쏘맥스, 프로맥스, 해리스카이저, oblique procrustean 방법이
있다.

(2) 주성분분석 (PROC PRINCOMP)

```
PROC PRINCOMP [DATA=SASdataset OUT=SASdataset  
OUTSTAT=SASdataset NOINT COVARIANCE |  
COV N=n STANDARD |STD PREFIX=name  
NOPRINT];  
[VAR variables;]  
[PARTIAL variables;]  
[FREQ variable;]  
[WEIGHT variable;]  
[BY variables;]
```

잠재적 모형에 의존치 않는 변수들의 직교변환에 의해 요인들을 추출해준다.

(3) 점수추출 (PROC SCORE)

```
PROC SCORE [DATA=SASdataset SCORE=SASdataset  
OUT=SASdataset TYPE=value PREDICT NOSTD];  
[VAR variables;]
```

2개의 SAS data set에 있는 값들을 곱해주는데 이용된다. 하나는 계수(예를 들어 인자점수계수 또는 회귀계수)이고 다른 하나는 계수를 얻어냈을 때 쓰인 원자료 data set 일 때를 말한다.

마. 정준분석 (PROC CANCORR)

```
PROC CANCORR [DATA=SASdataset OUT=SASdataset  
OUTSTAT=SASdataset SIMPLE | S CORR | C  
REDUNDANCY | RED ALL SHORT NOPRINT  
NCAN=n EDF=errordf RDF=regressiondf  
NOINT SINGULAR | SING=p VPREFIX | VP=name  
WPREFIX | WP=name VNAME | VN='label'  
WNAME | WN='label'];  
[VAR variables];  
[WITH variables];  
[PARTIAL variables];  
[FREQ variable];  
[WEIGHT variable];  
[BY variables];
```

2 개의 변수군간의 상호관계를 분석하는데 사용된다.

(다변량분석기법중의 하나)

바. 판별분석기법

(1) 판별분석 (PROC DISCRIM)

```
PROC DISCRIM [DATA=SASdataset OUT=SASdataset  
TESTDATA=SASdataset SIMPLE |S POOL=YES |  
NO |TEST SLPOOL=n WCOV W CORR PCOV  
PCORR LIST LISTERR THRESHOLD=n  
NOSUMMARY TESTLIST TESTLISTERR];  
CLASS variable;  
[VAR variables;]  
[ID variable;]  
[PRIORS probabilities |PROPORTIONAL |PROP;]  
[TESTCLASS variable;]  
[TESTID variable;]  
[BY variables;]
```

한개 또는 그이상의 수치변수정보에 의해서 전자료를 2개 또는 그이상의 군으로 판측치들을 구분하기 위해 선형 또는 이차함수를 계산하여 분석을 수행한다.

(2) 정준판별분석 (PROC CANDISC)

```
PROC CANDISC [DATA=SASdataset OUT=SASdataset  
OUTSTAT=SASdataset NCAN=n PREFIX=name  
UNIVARIATE |UNI STDMEAN TCORR W CORR  
BCORR TCOV WCOV BCOV TSSCP WSSCP BSSCP  
MAHALANOBIS |MAH ALL SHORT NOPRINT  
SINGULAR |SING=p EDF=n RDF=n];  
[VAR variables;]  
CLASS variable;  
PROB variables;  
[FREQ variable;]  
[WEIGHT variable;]  
[BY variables;]
```

정준판별분석 수행 및 마할라노비스 거리계산 (Mahalanobis distance)과 단변량 및 다변량 일원분산분석을 수행한다.

출력 Data set 내에는 정준계수와 정준변수들에 대한 점수가 들어갈 수 있도록 할 수 있다.

(3) 단계별 판별분석 (PROC STEPDISC)

```
PROC STEPDISC [DATA=SASdataset STEPWISE |  
  SW FORWARD | FW BACKWARD | BW SLENTRY |  
  SLE=p SLSTAY | SLS=p PR2ENTRY | PR2E=p  
  PR2STAY | PR2S=p SINGULAR | SING=p  
  INCLUDE=n MAXSTEP=n SIMPLE  
  STDMEAN TCORR WCORR SHORT];  
[VAR variables;]  
[CLASS variable;]  
[PROB variables;]  
[FREQ variable;]  
[WEIGHT variable;]  
[BY variables;]
```

주어진 자료에 대해 몇개의 급을 나누기 위한 유용한 변수를 추출할 때 전방추출법, 후방추출법, 단계별 추출법에 의한 판별분석을 수행한다.

(3) 최근방거리 판별분석 (PROC NEIGHBOR)

```
PROC NEIGHBOR [DATA=SASdataset  
  TESTDATA=SASdataset K=k IDENTITY  
  THRESHOLD=p LIST LISTERR TESTLIST  
  TESTLISTERR];  
CLASS variable;  
[VAR variables;]  
[ID variable;]  
[PRIORS probabilities | PROPORTIONAL | PROP;]  
[TESTCLASS variable;]  
[TESTID variable;]  
[BY variables;]
```

관측치들을 최근방방법 또는 K-최근방방법에 의해 분류하는 최근방 판별분석을 수행한다.

사. 집락분석기법

(1) 집락분석 (PROC CLUSTER)

```
PROC CLUSTER [DATA = SASdataset  
  OUTTREE = SASdataset SIMPLE | S NOEIGEN  
  STANDARD | STD NOSQUARE TRIM = p DIM = n  
  K = n R = n HYBRID METHOD | M = AVERAGE | AVE  
  METHOD | M = CENTROID | CEN METHOD |  
  M = COMPLETE | COM METHOD | M = DENSITY | DEN  
  METHOD | M = EML METHOD | M = FLEXIBLE | FLE  
  METHOD | M = MCQUITTY | MCQ METHOD | M = MEDIAN |  
  MED METHOD | M = SINGLE | SIN METHOD |  
  M = TWOSTAGE | TWO METHOD | M = WARD | WAR  
  BETA = n PENALTY = p MODE = n PRINT | P = n  
  RMSSTD RSQUARE | RSQ CCC PSEUDO NOID  
  NONORM NOPRINT];  
[VAR variables];  
[ID variables];  
[COPY variables];  
[FREQ variable];  
[RMSSTD variable];  
[BY variables];
```

SAS data set 내의 관측치들의 계층적 집락들을 찾는 데 사용된다.

(2) 간편집락분석 (PROC FASTCLUS)

```
PROC FASTCLUS [DATA = SASdataset  
  SEED = SASdataset OUT = SASdataset  
  MEAN = SASdataset CLUSTER = name  
  MAXCLUSTERS | MAXC = n RADIUS = n  
  REPLACE = FULL | PART | NONE | RANDOM  
  RANDOM = n DRIFT STRICT | STRICT = n  
  MAXITER = n CONVERGE | CONV = n  
  DELETE = n LIST DISTANCE SHORT  
  SUMMARY NOPRINT IMPUTE NOMISS  
  VARDEF = N | DF | WEIGHT | WGT | WDF];  
[VAR variables];  
[ID variable];  
[FREQ variable];  
[WEIGHT variable];  
[BY variables];
```

양이 매우 큰 자료에 대해 서로 중복되지 않는 집락화를 하도록 고안되어졌고 2~3회의 자료검색으로 바람직한 집락들을 구할 수 있다.

(3) Tree diagram 작성 (PROC TREE)

```
PROC TREE [DATA=SASdataset OUT=SASdataset
HEIGHT | H=NCL | N HEIGHT | H=HEIGHT |
H HEIGHT | H=MODE | M HEIGHT | H=RSQ |
R HEIGHT | H=LENGTH | L SIMILAR | SIM
DISSIMILAR | DIS LEVEL=n NCLUSTERS |
NCL | N=n DOCK=n ROOT='name'
SORT DESCENDING MINHEIGHT | MINH=n
MAXHEIGHT | MAXH=n SPACES | S=s PAGES=n
POS=n TICKPOS=n NTICK=n INC=n
LEAFCHAR | LC='c' TREECHAR | TC='c'
JOINCHAR | JC='c' FILLCHAR | FC='c'
LIST NOPRINT];
[NAME variable;]
[PARENT variable;]
[HEIGHT variable;]
[ID variable;]
[COPY variables;]
[FREQ variable;]
[BY variables;]
```

PROC CLUSTER 또는 VARCLUS에 의해 얻어진 data set 을
이용해서 Tree 도표를 인쇄하여 준다.

(4) 공분산 집락분석 (PROC VARCLUS)

```
PROC VARCLUS [DATA=SASdataset
OUTSTAT=SASdataset OUTTREE=SASdataset
SIMPLE | S CORR | C MINCLUSTERS | MINC=n
MAXCLUSTERS | MAXC=n PROPORTION | PERCENT=n
MAXEIGEN=n COVARIANCE | COV
INITIAL=RANDOM | SEED | INPUT | GROUP
CENTROID MAXITER=n MAXSEARCH=n
HIERARCHY | HI MULTIPLEGROUP | MG SHORT
SUMMARY NOPRINT TRACE];
[VAR variables;]
[SEED | SEEDS variables;]
[PARTIAL variables;]
[WEIGHT variable;]
[FREQ variable;]
[BY variables;]
```

상관 또는 공분산행렬에 근거를 두고서 중복되지 않은 또는 계층적
집락화를 수행해 준다.

(5) 집락내 합동 등공분산 추정(PROC ACECLUS)

```
PROC ACECLUS [DATA=SASdataset OUT=SASdataset  
OUTSTAT=SASdataset INITIAL=FULL|F  
INITIAL=DIAGONAL|D INITIAL=IDENTITY|I  
INITIAL=INPUT=SASdataset THRESHOLD|T=t  
PROPORTION|PERCENT|P=p ABSOLUTE  
MAXITER=n CONVERGE=c SINGULAR=g N=n  
PREFIX=name PP QQ SHORT NOPRINT  
METRIC=FULL|F METRIC=DIAGONAL|D  
METRIC=IDENTITY|I];  
[VAR variables];  
[FREQ variable];  
[WEIGHT variable];  
[BY variables];
```

집락들이 등공분산 행렬을 갖는 다변량 정규분포를 한다고 가정할 수 있을 때 집락내 합동공분산 행렬의 근사추정치를 구하는데 쓰인다.

(6) 유목별 자료분석(PROC CATMOD)

```
PROC CATMOD [DATA=SASdataset ORDER=DATA];  
[DIRECT variables];  
MODEL response_effect=design_effects  
[/ ONEWAY FREQ PROB XPX COV COVB CORRB  
ML PREDICT|PRED=FREQ|PROB NODESIGN  
NOPARM NOPROFILE NOINT NOGLS  
ADDCELL=number AVERAGED  
MAXITER=number EPSILON=number];  
[CONTRAST 'label' rowdescription, ...];  
where rowdescription has the form  
[@n] effect|INTERCEPT|ALL_PARMS values ...  
[POPULATION variables];  
[REPEATED factordescription, ...  
[/ _RESPONSE_=effects];  
where factordescription has the form  
factormame levels  
[RESPONSE transformation|MARGINAL|MARGINALS|  
MEAN|MEANS|LOGIT|LOGITS|JOINT  
[OUT=SASdataset OUTEST=SASdataset];  
[WEIGHT variable];  
[BY variables];
```

선형모형을 반응dot수함수로 적합시키거나 선형모형화나 대수선형모형화 로지스틱회귀, 반복측정분석에 사용할 수 있다.

아. 비모수 통계분석 (PROC NPAR1WAY)

```
PROC NPAR1WAY [DATA=SASdataset ANOVA  
              WILCOXON MEDIAN VW SAVAGE];  
!f [VAR variables];  
   CLASS variable;  
   [BY variables];
```

일원분류에 따른 반응변수의 서열 및 서열점수에 대한 분산분석을
수행해 준다.

자. 시계열분석기법

(1) Box - Jenkins 시계열분석 (PROC ARIMA)

```
PROC ARIMA [DATA=SASdataset OUT=SASdataset
CONVERSE];
IDENTIFY VAR=variable [(d1, d2,...dk)] [DATA=SASdataset
CROSSCORR | CROSSCOR | CROSS=(variable
[(d1, d2,...dk)... variable (d1, d2,...dk))] NLAG=number
NOPRINT CENTER CLEAR];
[ESTIMATE [P=lag | (lag,...,lag)...(lag,...,lag)
Q=lag | (lag,...,lag)... (lag,lag,...,lag) INPUT=(form1
variable1 form2 variable2 ...) ALTPARM
NOCONSTANT | NOINT NOPRINT PRINTALL PLOT NODF
GRID GRIDVAL=number METHOD=ML | ULS | CLS NOLS
AR=numbers MA=numbers INITVAL=(values variable...)
MU=number NOEST CONVERGE=number MAXIT=number
DELTA=number SINGULAR=number BACKLIM=-number
NOSTABLE)];
[FORECAST LEAD=number [BACK=number OUT=SASdataset
ID=variable INTERVAL=interval INTPER=number
NOPRINT PRINTALL NOOUTALL)];
[QUIT];
[BY variables];
```

주로 Box 와 Jenkins에 의해 이론이 정립된 Time-domain 기법의 시계열모형을 세울때 이용된다. 계절모형, 비계절모형, 전이함수모형, Intervention 모형의 정립이 가능하다.

(2) 자기회귀 시계열 모형추정 (PROC AUTOREG)

```
PROC AUTOREG [DATA=SASdataset OUTEST=SASdataset];
label:MODEL dependent=regressors[/ NLAG=p NOINT
NOPRINT ALL COEF CORR COVB GINV PARTIAL
BACKSTEP SLSTAY=value METHOD=ML | ULS | YW
NOMISS ITER ITPRINT MAXIT=number CONVERGE=number];
[LAGLIST numbers];
[OUTPUT [OUT=SASdataset ALPHACLI=number
ALPHACL M=number PREDICTED | P=name RESIDUAL | R=name
UCL=name LCL=name PREDICTEDM | PM=name
RESIDUALM | RM=name UCLM=name LCLM=name
TRANSFORM=variables CONSTANT=variable];
[BY variables];
```

시계열자료의 모형설정시 자기회귀 항들만에 의해 모형을 설정할 때 사용한다.

(3) 시계열 예측 (PROC FORECAST)

```
PROC FORECAST DATA=SASdataset
OUT=SASdataset OUTEST=SASdataset
METHOD=STEPAR | EXPO | WINTERS TREND=n
NLAGS=n AR=n SEASONS=interval { (interval, |interval2
|intervalj ) } | n SINTPER=n { (n1 |n2 |n3 ) } INTERVAL=interval
INTPER=n LEAD=n START=n OUTLIMIT OUTSTEP
OUTACTUAL OUTRESID OUTDATA OUTALL SLSTAY=n
SLENTRY=n WEIGHT= $\omega$  { ( $\omega_1$  | $\omega_2$  | $\omega_3$  ) } ALPHA=n
ZEROMISS SINGULAR=n);
[VAR variables;]
[ID variables;]
[BY variables;]
```

단계별 자기회귀방법, 지수평활법, Winters의 방법등을 이용하여
시계열자료의 예측을 할때 사용된다.

(4) 시계열의 계절조정 (PROC X11)

```
PROC X11 [DATA=SASdataset YRAHEADOUT];
[VAR variables;]
[ID variables;]
[BY variables;]
[MONTHLY [START=mmmyy END=mmmyy DATE=variable
ADDITIVE SUMMARY NDEC=number
PRINTOUT=STANDARD | LONG | FULL | NONE
CHARTS=STANDARD | NONE | ALL LENGTH
TDREGR=NONE | PRINT | ADJUST | TEST TDCOMPUTE=yy
EXCLUDE=x.x FULLWEIGHT | ZEROWEIGHT=x.x
TRENDMA=9 | 13 | 23 TRENDADJ PMFACTOR=variable
OLD];]
[PDWEIGHTS [day=x.xxx];]
[MACURVES month=specification STABLE];]
[QUARTERLY [START='yyQq' END='yyQq'
DATE=variable ADDITIVE SUMMARY
PRINTOUT=STANDARD | LONG | FULL | NONE
FULLWEIGHT | ZEROWEIGHT=x.x TRENDADJ];]
[OUTPUT[OUT=SASdataset table=variable];]
```

시계열자료에 계절성이 있을 때 이의 제거를 위해 사용된다.

(5) 시차분포 모형 추정 (PROC PDLREG)

```
PROC PDLREG [DATA=SASdataset];
  MODEL dependent=effects [/ NLAG=n NOPRINT NOINT XPF
    I COVB CORRB ALL DW STB PARTIAL];
  [BY variables;]
  [OUTPUT OUT=SASdataset PREDICTED | P=variable
    RESTRICTED | R=variables...;]
  [RESTRICT restrictions;]
  [LAGLIST lags;]
  where effects can be of the form variable(p,d,l,q).
```

Koyck 및 Almon의 시차분포 모형에 있어서의 모수추정치들을 구해 준다.

(6) 시계열의 Spectral 밀도함수 추정 (PROC SPECTRA)

```
PROC SPECTRA [DATA=SASdataset OUT=SASdataset COEF P S
  CROSS A K PH WHITETEST CENTER |ADJMEAN];
  VAR variables;
  [BY variables;]
  [WEIGHTS constants;]
```

frequency-domain 방법에 의한 시계열 모형을 세울시 시계열자료의 모형선택에 필요한 Spectral 밀도함수 식별에 이용된다.

(7) 상태공간 벡터 추정 (PROC STATESPACE)

```
PROC STATESPACE [DATA=SASdataset NOCENTER LAGMAX=number
  PRINTOUT=SHORT |LONG |NONE ARMAX=number
  MINIC OUTAR=SASdataset DIMMAX=number PASTMIN=number
  SIGCORR=number CANCELR NOEST RESIDEST COVB
  KLAG=number MAXIT=number SINGULAR=number
  PARMTOL=number DETTOL=number ITPRINT
  OUTMODEL=SASdataset OUT=SASdataset BACK=number
  LEAD=number PRINT INTERVAL=interval INTPER=number];
  [VAR variable(difference,difference, ...) ...;]
  [FORM variable value ...;]
  [RESTRICT F(row,column)=value ... G(row,column)=value ...;]
  [INITIAL F(row,column)=value ... G(row,column)=value ...;]
  [ID variable;]
  [BY variables;]
```

시계열분석중 상태공간 벡터를 구하기 위하여 사용된다.

차. 시스템 모델링 기법

(1) 모형 정립 (PROC MODEL)

```
PROC MODEL (MODEL=SASdataset(s) OUTMODEL=SASdataset
LIST XREF LISTCODE);
[ENDOGENOUS | ENDO variable {initial values} ...;]
[EXOGENOUS | EXO variable {initial values} ...;]
[PARAMETERS | PARMS variable {value} ...;]
[INSTRUMENTS | INST variables;]
[CONTROL variable {value} ... ;]
[OUTVARS variables;]
[LABEL variable='label' ...;]
[programming statements;]
```

시뮬레이션에 들어가기 전에 모형의 정립을 위한 전단계 작업으로 사용된다.
(주로 PROC SYNLIN의 전단계 작업으로 사용됨)

(2) 선형연립방정식 모형 추정 (PROC SYSLIN)

```
PROC SYSLIN (DATA=SASdataset OUT=SASdataset
OUTSSCP=SASdataset OUTTEST=SASdataset SIMPLE FIRST
USSCP USSCP2 DFNO NODFS SDIAG NOINCLUDE
NOPRINT ALL ALPHA=value K=value LIML 2SLS 3SLS
IT3SLS SUR MELO COVOUT COV3OUT REDUCED
EPSILON=number MAXIT=number CONVERGE=number);
[{{label:}MODEL response=regressors [/ NOPRINT NOINT XPX
UNREST 1 COVB CORRB ALL TPRINT DW STB PLOT
OVERID ALPHA=value K=value;]
[INSTRUMENTS variables;]
[ENDOGENOUS variables;]
[RESTRICT restriction1...restrictionk;]
[SRESTRICT restriction1...restrictionk;]
[TEST equation1...equationk;]
[STEST equation1...equationk;]
[IDENTITY identity1...identityk;]
[OUTPUT PREDICTED | PRED | P=variable
RESIDUAL | R=variable;]
[WEIGHT variable;]
[BY variables;]
[VAR variables;]
[ID variables;]
```

선형연립방정식 모형의 추정에 사용된다. 추정방법으로는 최소자승법, 2 단계 최소자승법, 3 단계 최소자승법, K-급 최소자승법, 제한정보최우추정법등이 사용된다.

(3) 선형연립방정식 모형 Simulation (PROC SIMLIN)

```
PROC SIMLIN EST=SASdataset DATA=SASdataset TYPE=value
ESTPRINT NORED INTERIM=number TOTAL START=number);
[ID variables;]
[ENDOGENOUS variables;]
[EXOGENOUS variables;]
[LAGGED lagvariable endogenousvariable number...;]
[OUTPUT {OUT=SASdataset PREDICTED | P=names
RESIDUAL | R=names;]
[BY variables;]
```

선형연립방정식 모형의 Simulation시 이용된다.

(4) 비선형 연립방정식 모형 추정 (PROC SYNLIN)

```

PROC SYNLIN [OLS 2LS | N2SLS SUR 3SLS | N3SLS ITOLS ITSUR
IT2SLS IT3SLS VARDEF=N | WGT | DF DATA=SASdataset
MODEL=SASdataset OUTMODEL=SASdataset NOSTORE
SDATA=SASdataset OUTS=SASdataset OUTSUSED=SASdataset
OUTEST=SASdataset COVOUT ESTDATA=SASdataset
OUT=SASdataset OUTPREDICT OUTACTUAL OUTRESID
OUTLAGS FRSQ COVB CORR8 COVS CORRS DW
COLLIN DETAILS PRINTALL NOPRINT LIST ITPRINT
ITDETAILS METHOD=GAUSS | MARQUARDT
CONVERGE=n MAXIT=n NESTIT MAXSUBIT=n
SINGULAR=n XPX | ITALL LISTCODE XREF MAXERROR=n];
[FIT equations {!ESTART=(parameter values ...)}
{KEEP=(parameters)} {DROP=(parameters)}];
[INSTRUMENTS |INST {instruments} {_EXOG_} {ADD=(instruments
{_EXOG_})} {EXCLUDE=(parameters)}
{DROP=(instruments {_EXOG_})} {ADD=(instruments
{_EXOG_})} {DROP=(instruments {_EXOG_})}
NOINTERCEPT | NOINT];
[RANGE variable {[- first] | TO last};]
[PARAMETERS | PARMS {variable {value} ...};]
[CONTROL variable {value} ...];
[ENDOGENOUS | ENDO variables;]
[EXOGENOUS | EXO variables;]
[WEIGHT variable;]
[BY variables;]
[ID variables;]
[OUTVARS variables;]
[LABEL variable='label' ...];

```

비선형 연립방정식 모형의 추정에 사용된다. 추정방법으로는 비선형 최소자승법, 비선형 2 단계 최소자승법, 비선형 3 단계 최소자승법, 비선형 구조별 회귀방법 등이 있다.

(5) 비선형 연립방정식 모형 Simulation (PROC SIMNLIN)

```

PROC SIMNLIN [STATIC DYNAMIC NAHEAD=n START=s
FORECAST SIMULATE NEWTON JACOBI SINGLE | ONEPASS
SEIDEL DATA=SASdataset MODEL=SASdataset(s)
OUTMODEL=SASdataset OUT=SASdataset OUTPREDICT
OUTRESID OUTACTUAL OUTERRORES OUTLAGS
RANDOM=n ESTDATA=SASdataset SDATA=SASdataset
TYPE=name SEED=n STAT THEIL BLOCK GRAPH PRINT
ITPRINT DETAILS NOPRINT LIST XREF LISTCODE
CONVERGE=n MAXIT=n];
[RANGE variable [-first] | TO last];]
[ENDOGENOUS | ENDO variable {init-lag-values} ...];]
[EXOGENOUS | EXO variable {init-lag-values}...];]
[PARAMETERS | PARMS variable {value}...];]
[CONTROL variable {value} ...];]
[BY variables;]
[ID variables;]
[OUTVARS variables;]
[LABEL variable = 'label' ...];]

```

비선형 연립방정식의 구조를 갖는 모형에 대한 Simulation 시 사용된다.

카. 기타 통계분석

(1) Citi Bank Data Base (PROC CITIBASE)

```
PROC CITIBASE [DDNAME | INDD = fileref  
TYPE | INTERVAL = QTR | MONTH | YEAR BEGINYR = yy | yyyy  
ENDYR = yy | yyyy OUT = SASdataset NOPRINT];
```

SELECT specifications;

where specifications can be a name, an alphabetic range, an order range, or a prefix.

CITI BANK에서 제공하는 그들의 Data Base 자료를 이용할 때 쓰인다.

(2) Table 정리 및 계산 (PROC COMPUTAB)

```
PROC COMPUTAB [DATA = SASdataset OUT = SASdataset  
NOPRINT NOTRANSPOSE | NOTRANS OPTIONS  
CWIDTH = w CSPACE = n CDEC = d RTS = n NORTR  
INITMISS FUZZ = nnnn SUMONLY];  
COLUMNS | COL names / [ CHAR 'text' ['text'] MTITLE | M = 'text'  
_NAME_ _LABEL_ + n _PAGE_ _TITLES_ | _TITLE_  
NOPRINT NOZERO FORMAT | F = format ZERO = 'text' LJC];  
[ROWS | ROW# names [/ CHAR 'text' ['text'] _NAME_  
_LABEL_ _PAGE_ + n UL | DUL OL | DOL SKIP NOPRINT  
NOZERO OVERPRINT | OVP FORMAT | F = format  
ZERO = 'text' LJC];
```

[programming statements];

[COLxxxxx: statements];

[ROWxxxxx: statements];

[INIT terms initialvalues];

[INIT anchorname | locatorname | values | locatorname values] + n
n*value];

[BY variables];

[SUMBY variables];

where names can be of the form name1, name2, ..., namen.

COMPU ting and TABular reporting의 약자인데 Table의 행 및 열에 대한 계산과 Table의 Title, Table의 양식등을 조절할 수 있다.

(3) 저당 및 이자율 계산 (PROC MORTGAGE)

```
PROC MORTGAGE options [AMOUNT | A = d MONTHS | M = m  
RATE | R = r PAYMENT | P = p MO = n YR = yyyy N = n  
YEARLY | Y OUT = SASdataset NOPRINT];
```

저당 및 이자율의 계산에 쓰인다.

(4) 재무분석 계산에 유용한 함수들

COMPOUND (*amount, future, rate, number*)

INTRR (*period, cash0, cash1, ..., cashn*)

IRR (*period, cash0, cash1, ..., cashn*)

MORT (*amount, payment, rate, number*)

NETPV (*rate, period, cash0, cash1, ..., cashn*)

NPV (*rate, period, cash0, cash1, ..., cashn*)

SAVING (*future, payment, rate, number*)

타. Dataset 처리

(1) SAS Dataset 에 다른 SAS Dataset 추가 (APPEND)

```
PROC APPEND BASE=OUT=SASdataset  
[DATA|NEW=SASdataset FORCE];
```

(2) SAS Dataset 내용 READ (BROWSE)

```
PROC BROWSE [DATA=SASdataset FUZZ=value];  
[FORMAT variable format ...;]  
[INFORMAT variable informat ...;]  
[FIND [VERIFY|VER ALL] range variable1  
operator1 value1 ...;]  
[LOCATE [VERIFY|VER ALL] range value ...;]  
[SEARCH [VERIFY|VER ALL] range string ...;]  
[NAME variable;]  
[STRING variable ...;]  
[VERIFY [OFF|ON RESET NOLIST|LIST];]  
[END;]  
[LIST range variable ...;]  
[TOP;]  
[BOTTOM;]  
[UP n;]  
[DOWN n;]
```

(3) 2개의 SAS Dataset 비교 (COMPARE)

```
PROC COMPARE [DATA=SASdataset  
COMPARE=SASdataset OUT=SASdataset  
OUTPERCENT OUTNOEQUAL CRITERION=n  
METHOD=name ALLOBS NOOBS ALLVARS  
STATS ALLSTATS NOSUMMARY NOLISTEQUAL  
FUZZ=n NOMISSING|NOMISS  
NOMISS1 NOMISS2];  
[VAR variables;]  
[WITH variables;]  
[ID variables;]  
[BY variables;]
```

(4) SAS file 내용 PRINT (CONTENTS)

```
PROC CONTENTS [DATA=libref.member DIRECTORY  
HISTORY MEMTYPE|MTYPE|MT=ALL|CAT|  
DATA|FORMATC|FORMATN|GCAT|  
IMSWK|MODEL NODS NOPRINT NOSOURCE  
OUT=SASdataset POSITION SHORT];
```

(5) SAS Library Member Copy (COPY)

```
PROC COPY IN | INDD = libref OUT | OUTDD = libref
  [EXPORT IMPORT MEMTYPE | MTYPE |
  MT = ALL | CAT | DATA | GCAT | IMLWK |
  MODEL MOVE NOHISTORY PROTECT | PROT = p];
[SELECT member ... [ / MEMTYPE = type PROTECT = p ];]
[EXCLUDE member ... [ MEMTYPE = type ];]
```

(6) SAS Library Member List 작성, 이름변경, 삭제 (DATASETS)

```
PROC DATASETS [LIBRARY | DDNAME = libref]
  [MEMTYPE | MTYPE | MT = ALL | CAT | DATA |
  GCAT | IMLWK | MODEL NOFS KILL FORCE
  NOLIST NOWARN];
[DELETE members [(options)] ... [/options];]
[SAVE members [(options)] ... [/options];]
[CHANGE oldname = newname ... [/options];]
[EXCHANGE name = anothername ... [/options];]
[AGE currentname name2 ... lastname [(options)]
[/options];]
[MODIFY SASdataset [(LABEL = 'newlabel'
  PROTECT = writepassword PROTECT = old/new
  READ = readpassword READ = old/new
  TYPE = specialtype)];]
[FORMAT variable ... format ...;]
[INFORMAT variable ... informat ...;]
[LABEL variable = 'newlabel' ...;]
[RENAME variable = newname ...;]
```

where options can be chosen from MEMTYPE |
MTYPE | MT = type PROTECT = p

(7) SAS Dataset 조사변경 (EDITOR)

```
PROC EDITOR [DATA = SASdataset FUZZ = value];
  [FORMAT variable format ...;]
  [INFORMAT variable informat ...;]
  [FIND [VERIFY | VER ALL] range variable | operator |
  value1 ...;]
  [LOCATE [VERIFY | VER ALL] range value ...;]
  [SEARCH | S [VERIFY | VER ALL] range string ...;]
```

Interactive SAS session에서 기존의 특정 SAS data set의 수정에 사용된다.

(8) SAS Dataset 내의 변수 및 변수값 PRINT (PRINT)

```
PROC PRINT [DATA=SASdataset N UNIFORM | U
          DOUBLE | D ROUND LABEL SPLIT='splitchar'
          NOOBS];
[VAR variables;]
[ID variables;]
[PAGEBY byvariable;]
[SUM variables;]
[SUMBY byvariable;]
[BY variables;]
```

(9) SAS Dataset Sort (SORT)

```
PROC SORT [DATA=SASdataset OUT=SASdataset
          EQUALS NODUPPLICATES | NODUP NATIONAL
          REVERSE DANISH NORWEGIAN FINNISH
          SWEDISH MESSAGE | M LIST | L LEAVE=n
          TECHNIQUE | T=xxxx SORTWKNO=number
          DIAG SORTSIZE | SIZE=parameter];
BY [DESCENDING] variable ...;
```

(10) SAS Dataset의 변수와 변수값의 위치변경 (TRANSPOSE)

```
PROC TRANSPOSE [DATA=SASdataset PREFIX=name
              OUT=SASdataset NAME=name LABEL=name
              LET];
[VAR variables;]
[ID variable;]
[IDLABEL variable;]
[COPY variables;]
[BY variables;]
```

SAS data set의 행과 열을 바꾸고자 할 때 사용

(11) 출력양식 정의 (FORMAT)

```
PROC FORMAT [DDNAME=libref | fileref DECK];
[VALUE name ((MAX=n MIN=n DEFAULT=n
            FUZZ=n)) range='label' ...];
[PICTURE name ((MAX=n MIN=n
            DEFAULT=n)) range=picture
            ((FILL='character' PREFIX='character'
            MULTIPLIER | MULT=n NOEDIT)) ...];
```

특정 SAS 변수에 들어 있는 값에 대해 이용자가 따로 정의하여 출력시키고자 할 때 사용.

(12) 정형적인 출력양식 Print (FORMS)

```
PROC FORMS [DATA=SASdataset DDNAME | DD=fileref
          WIDTH | W=number LINES | L=number
          DOWN | D=number SKIP | S=number
          NACROSS | NA=number BETWEEN | B=number
          INDENT | I=number NDOWN | ND=number
          PAGESIZE | P=number CC COPIES | C=number
          SETS=number ALIGN=number];
LINE linenum variables [/ INDENT | I=number
          PACK | P LASTNAME | L REMOVE | R];
[FREQ variable;]
[BY variables;]
```

SAS data set에 대해 좀더 정교한 출력을 원할 때 사용

(13) 월별 일정표 작성 (CALENDAR)

```
PROC CALENDAR [DATA = SASdataset  
HOLIDATA = SASdataset SCHEDULE FILL  
WEEKDAYS MISSING DATETIME  
HEADER = LARGE | SMALL  
FORMCHAR{(n ...)} = 'string'  
LEGEND MEANTYPE = NOBS | NDAYS];  
ID variable;  
[VAR variables;]  
[SUM variables [/ FORMAT | F = format];]  
[MEAN variables [/ FORMAT | F = format];]  
[DURATION variable;]  
[HOLIDAYS variables;]  
[HOLINAME variable;]  
[BY variables;]
```

(14) SAS OPTION List (OPTIONS)

```
PROC OPTIONS [SHORT CMS DU IMS];
```

현재 Install 되어 있는 SAS option 들을 알아보고자 할 때 사용

(15) PDS Dataset 의 Member List 작성, 이름변경, 삭제 (PDS)

```
PROC PDS DDNAME = fileref [NOLIST KILL];  
[DELETE member ...;]  
[CHANGE oldname = newname ...;]  
[EXCHANGE name = anothername ...;]
```

(16) PDS Dataset COPY (PDSCOPY)

```
PROC PDSCOPY INDD = fileref OUTDD = fileref  
[ALIASMATCH = TTR | NAME | BOTH | EITHER  
BLKSIZE = b DC DCBS INTAPE MAXBLOCK |  
MAXBLK = b NE NEWMOD NOALIAS | NOA  
NODCBS NOREPLACE | NOR OUTTAPE  
SHAREINPUT | SHAREIN];  
[SELECT modulename ...;]  
[EXCLUDE modulename ...;]
```

(17) Disk Dataset Space Release (RELEASE)

```
PROC RELEASE DDNAME = fileref  
[TOTAL | TRACKS = number  
UNUSED = number RELEASE = number  
EXTENTS | EXTENT | EX  
BOUNDARY | TYPE = DSCB | DATASET  
BOUNDARY | TYPE = CYL | CYLS | CYLINDER |  
CYLINDERS BOUNDARY | TYPE = TRK | TRKS |  
TRACK | TRACKS BOUNDARY | TYPE = JCL | DD |  
ALLOC];
```

(8) Backup, Member List 및 Directory 작성 (SOURCE)

```
PROC SOURCE [DIRDD=fileref INBLK=blocksize
            INDD=fileref MAXIOERROR=number
            NOALIAS NODATA NOPRINT NOSUMMARY
            NOTSORTED NULL OUTBLK=blocksize
            OUTDD=fileref PAGE PRINT
            SEARCH SUBLIB=oneletter];
[SELECT member ...;]
[EXCLUDE member ...;]
[FIRST 'model control statement' ...;]
[LAST 'model control statement' ...;]
[BEFORE 'model control statement' [ALIAS column
FULL NOBLANK RIGHT];]
[AFTER 'model control statement' [ALIAS column
FULL NOBLANK RIGHT];]
```

(9) Tape Copy (TAPECOPY)

```
PROC TAPECOPY [NOLIST DEN=density LABEL=SL|NL
              COPYVOLSER DETACH INDD=fileref
              OUTDD=fileref INVOL=volumeserial
              OUTVOL=volumeserial NORER
              NEWVOLSER=newvolumeserial NOFSNRESEQ|NFR
              TAP1 TAPn /'CM$mount'];
[INVOL [INDD=fileref INVOL=volumeserial
        OUTVOL=volumeserial NL SL DSNAME|
        DSN='datasetname' NORER];]
[FILE[S] filenumbers;]
```

(20) Tape Label PRINT (TAPELABEL)

```
PROC TAPELABEL [DDNAME=(fileref ...) MAP
               PAGE TAPn];
```

(21) 상이한 OS하의 SAS Dataset 변경 (XCOPY)

```
PROC XCOPY IN=libref OUT=libref IMPORT|EXPORT;
[SELECT SASdataset ...;]
[EXCLUDE SASdataset ...;]
```

어떤 OS에서 얻어진 SAS data set을 임의 다른 OS 하에서
이용할 때 사용

- (2) BMDP, DATATEXT, OSIRIS, SPSS 및 version 72 SAS 등의 Dataset을 현재의 SAS와 호환성 있는 Dataset으로 변환 (CONVERT)

```
PROC CONVERT [BMDP=fileref  
DATATEXT=fileref OSIRIS=fileref  
DICT=fileref2 SAS72=fileref  
SPSS=fileref FIRSTOBS=n OBS=n  
OUT=SASdataset];
```

- (3) SAS 처리결과를 특정 file에 출력 (PRINTTO)

```
PROC PRINTTO [UNIT=nn NEW];
```

SAS proc에서 산출되는 결과를 이용자가 지정한 특정 file에 write 시킬 때 사용

부

부

(1) SAS 명령어 사용규약

KEYWORD parameter ... < item | item | item > options;
where

bold 체 : 이것은 명시돼 있는 것과 똑같은 형태와 철자를 써야함을 의미한다.

italic 체 : 이용자 각각의 요구에 맞는 정보를 제공하라는 의미.

[괄호쳐진 정보] : 선택적으로 쓸 수 있다.

괄호속에 있지 않는 : 선택적이 아니다.

parameter 들 : 특정값을 제공하라는 의미.

3 개의 마침표 (...) : 이것 앞에는 한 개 이상의 parameter 들이 있음을 의미하고 이들이 선택적으로 명시될 수 있음을 뜻한다.

수직막대 (|) : 이것은 Keyword (주단어) 들과 선택변수들 사이에 놓이게 되는데 의미는 이것에 의해 분리된 Keyword 들 또는 선택변수들 중에서 오직 한 개를 선택하라는 것이다.

options (선택변수들) : 개개의 SAS proc에서 요구할 수 있는 주선택 변수 (Keyword options) 들.

(2) Data Step에서 사용하는 명령어

- | | |
|---|--|
| <p>① ABORT [ABEND][RETURN] <i>n</i>;</p> | <p>① SAS의 실행을 즉시 중지시키는 문이다. ABEND 또는 RETURN문이 유사히 사용되어 질 수 있다.</p> |
| <p>② ARRAY <i>arrayname</i> {<i>n</i>} [\$] [<i>length</i>] [<i>arrayelements</i>];
 ARRAY <i>arrayname</i> [(<i>indexvariable</i>)] [\$] [<i>length</i>]
 <i>arrayelements</i>;</p> <p><i>variable</i>=<i>expression</i>; (assignment statement)</p> | <p>② 배열문 일반 language와 같은 기능을 한다.
할당문</p> |
| <p>③ ATTRIB <i>variable</i> [FORMAT=<i>format</i>]
 [INFORMAT=<i>informat</i>] [LABEL='label']
 [LENGTH={<i>\$</i>}<i>length</i>];</p> | <p>③ SAS 변수에 대한 FORMAT 또는 INFORMAT의 기술에 쓰인다.</p> |
| <p>④ BY [DESCENDING] <i>variable</i> ... [NOTSORTED];</p> | <p>④ SAS Data Set을 Group化시켜 처리할 때 사용</p> |
| <p>⑤ CALL <i>routine</i>(<i>parameter</i>, ...);</p> | <p>⑤ FORTRAN, PL/I, ASSEMBLER Module을 부를 때 사용</p> |
| <p>⑥ CARDS;
 <i>data lines</i>
 ;
 CARDS4;
 <i>data lines</i>
 ;;;</p> | <p>⑥ Data 입력을 Card image로 받아들일 때 사용</p> |
| <p>⑦ DATA [[<i>SASdataset</i>] [(<i>dsoptions</i>)]]...;</p> | <p>⑦ SAS Data Set의 Creation시 사용</p> |
| <p>⑧ DELETE;</p> | <p>⑧ SAS Data Set Creation시 필요없는 observation을 제외시킬 때 사용</p> |

⑨ **DO;**
more SAS statements
END;
DO *indexvariable* = *start* [**TO** *stop* [**BY** *increment*]
 [**WHILE** | **UNTIL**(*expression*)]]...;
more SAS statements
END;

⑩ **DO OVER** *arrayname*;

⑪ **DO WHILE**(*expression*);
DO UNTIL(*expression*);

⑫ **DROP** *variables*;

⑬ **END;**

⑭ **ERROR** [*message*];

⑮ **FILE** *fileref* [*typeoption*] [*options*];
 where *fileref* is chosen from
fileref | *fileref*(*membername*) | **LOG** | **PRINT** | **PUNCH**
 and where *typeoption* is chosen from **DA** **VSAM**
 and where *options* are chosen from **BLKSIZE** | **BLK** = *value*
CLOSE = *closedisposition* **COLUMN** = *variable*
DCB = *fileref* **DEVTYPE** = *variable* **DROPOVER**
DSCB = *variable* **FLOWOVER** **HEADER** = *label*
JFCB = *variable* **LINE** = *variable* **LINESIZE** | **LS** = *value*
LINESLEFT | **LL** = *variable* **LRECL** = *value*
MOD N = **PAGESIZE** | **PS** | *value* **NOPRINT**
NOTITLES | **NOTITLE** **OLD PAGESIZE** | **PS** = *value*
PRINT RECFM = *recordformat* **STOPOVER**
UCBNAME = *variable* **VOLUME[S]** = *variable*

⑯ **FORMAT** *variables* [*format*] ...;

⑰ **GOTO** | **GO TO** *label*;

⑨ 반복문장

⑩ 지정된 ARRAY의 전범위 걸친
 처리시 사용

⑪ PL/I에서의 기능과 같은
 기능을 수행

⑫ SAS Data Set Creation시
 필요없는 변수를 제외시킬 때
 사용

⑬ Do 문장의 끝을 의미

⑭ error 발생시 output에 출력
 시킬 내용을 기술하고자 할 때

⑮ 외부 file을 Creation시킬
 때 그 화일에 대한 정보를 제공

⑯ 변수에 대한 출력시 format을
 지정하고자 할 때 사용

⑰ SAS의 실행 제어를 할 때 이용

⑱ **IF** *expression* **THEN** *statement*;
ELSE *statement*;

⑲ **INFILE** *fileref*[(*membername*)]|**CARDS** [*typeoption*] [*options*]; ⑲ SAS System에서 Data Set 을
where *typeoption* can be chosen from **DA POWER VSAM**
VTOC
and where *options* for standard external files (without
typeoption) can be chosen from the standard option
list below

⑳ **INFILE** *fileref* **DA** [*special and standard options*];
where *special options* are chosen from
ABSBLOCK=variable CCHHR=variable
FEEDBACK=variable KEY=variable
KEYLEN|KEYLTH=nnn RELBLOCK=variable
SEARCH=method TTR=variable
and the *standard option* is **LINESIZE|LS=**

㉑ **INFILE** *jobname* **POWER** [*special and standard options*];
where *special options* are chosen from
CC=NO|YES CLASS=class JNUM=number
LST PUN PWD=password USERID=userid
and *standard options* can be chosen from
END=variable EOF=label FIRSTOBS=linenumber
LINESIZE|LS=linesize

㉒ **INFILE** *fileref* **VSAM** [*special and standard options*];
where *special options* are chosen from
BACKWARD|BKWD BUFND=integer
BUFNI=integer CONTROLINTERVAL|CTLINTV|
CNV ERASE=variable ERRORABEND FEEDBACK|
FDBK=variable GENKEY KEY=variable |
(*list of variables*) **KEYGE KEYLEN=variable**
KEYPOS=variable PASSWD=password
RBA=variable RC4STOP RECORDS=variable
RESET RRN=variable SEQUENTIAL|SEQ SKIP
UPDATE=variable
and *standard options* can be chosen from
COL=variable END=variable EOF=variable
FLOWOVER LINE=variable LINESIZE=variable
MISSEVER N=variable STOPOVER

⑱ IF문으로서 block if문의 기
능 및 nested If문 Do문장
의 삽입이 가능하다.

⑲ SAS System에서 Data Set 을
Read 할 때 외부화일을 읽을
때는 INFILE을 사용하고
In-stream data를 읽을 때는
CARDS를 사용케 된다.

⑳ Operating System이 VSE이고
외부 화일형태가 DA일 때 자료
Read에 이용

㉑ Operating System이 VSE이고
외부 화일이 VSE의 output
queue에 있을시 그것을 Acc-
ess 하는데 이용

㉒ Operating System이 CMS,
OS, VSE일 경우의 VSAM 화
일을 Access하는데 이용된다.

- ②③ **INFILE** *fileref* **VTOC** [*special and standard options*];
 where *special options* are chosen from
CCHHR=variable **CVAF**
 and *standard options* can be chosen from the standard
 option list below
 Standard option list:
BLKSIZE=blocksize **BSAM** **CCHHR=variable**
CLOSE=REREAD | LEAVE | REWIND | FREE | DISP
COLUMN | COL=variable **DCB=fileref**
DEVTYPE=variable **DSCB=variable**
END=variable **EOF=label**
EOV=variable **FIRSTOBS=linenumber**
FLOWOVER **IFCB=variable** **LENGTH=variable**
LINE=variable **LINESIZE | LS=linesize**
LRECL=logicalrecordlength **MISSOVER** **N=number**
OBS=line number **RECFM=record format**
START=variable **STOPOVER** **UCBNAME=variable**
UNBUFFERED | UNBUF VOLUMES | VOLUME=variable
- ②④ **INFORMAT** *variables* [*informat*] ...;
- ②⑤ **INPUT** *variable* [**\$**] *startcolumn* [**- endcolumn**] [*.decimals*];
INPUT *variable* [**\$**];
INPUT *variable* [**\$**] *informat*;
INPUT [*pointercontrol*] *variable* [**\$**] [*informat*];
- ②⑥ **KEEP** *variables*;
- ②⑦ **LABEL** *variable* = '*label*' ...;
label: statement;
- ②⑧ **LENGTH** [*variables* [**\$**] *length*] ... [**DEFAULT=n**];
- ②⑨ **LINK** *label*;
- ③⑩ **LIST**;
- ②③ Operating System이 OS,
 - VSE일 때 Disk pack의 내용을
 알아 볼 때 이용
- ②④ SAS data set 내에 정보를 입
 력시키는 상태의 form을 알려
 주는데 이용
- ②⑤ Data를 Read 할 때 쓰는 In-
 put문의 여러 형태를 말함
 (입력문)
- ②⑥ 특정 Data set 내에 필요한 변
 수만을 갖고자 할 때 사용
- ②⑦ 변수에 대한 이름을 주고자 할
 때 이용
- ②⑧ 변수에 대한 길이를 지정하는데
 사용
- ②⑨ 제어를 옮기고 싶은 곳을 문장
 의 이름을 명시하여 SAS 실행
 의 제어를 옮긴다.
- ③⑩ INPUT 문에 의해 읽은 Data
 를 바로 print할 때 이용

- ③1 **LOSTCARD;**
- ③2 **MERGE** SASdataset[*(dsoptions IN = name)*]
SASdataset[*(dsoptionsIN = name)*]... **[END = name];**
- ③3 **MISSING** values;
- ③4 ; (null statement)
- ③5 **OUTPUT** [SASdataset] ... ;
- ③6 **PUT** [variable] [=] [\$] startcolumn [-endcolumn]
[.decimalplaces];
PUT variable [=] [\$];
PUT variable [=] format | (variablelist) (formatlist);
- ③7 **RENAME** oldname = newname ... ;
- ③8 **RETAIN** [variables ... [initialvalue]] ... ;
- ③9 **RETURN;**
- ④0 **SELECT** [(selectexpression)];
WHEN (whenexpression) statement;
...
[OTHERWISE statement;]
END;
- ③1 한 Record 의 데이터중 일부가 Missing일 때 error가 그 레코드에 국한되도록 할 때 사용
- ③2 SAS data set 들을 Merge 시킬 때 사용
- ③3 Missing value 시의 표시로서 이용자의 제공 정보를 이용
- ③4 null statement
- ③5 특정한 SAS data set 에 관측치를 담고 싶을 때 사용
- ③6 Data Step 부문에서 얻어진 자료를 print 시킬 때 사용된다. (출력문)
- ③7 변수명을 바꾸고자 할 때 사용
- ③8 문자변수 또는 수치변수에 초기 값을 부여할 때 사용
- ③9 Link 문에 의해 제어를 넘겨 받은 후 다시 Link 문을 발생시킨 다음 문으로 돌려 보내는 역할을 한다.
- ④0 Case 구조를 갖는 logic 표현에 쓰는데 PL/I Language 의 SELECT 문과 같은 기능을 한다.

④① SET [[SASdataset[(dsoptions IN=name)...]
[POINT=name NOBS=name END=name]];

④② STOP;

④③ variable + expression; (sum statement)

④④ UPDATE masterdataset [(dsoptions IN=variable1)]
transactiondataset [(dsoptions IN=variable2)]
[END=variable];

④① 어떤 DATA Step 부문에서 기존의 SAS Data가 필요할 때 부르는 문

④② SAS 실행을 중지시킬 때 사용

④③ 합계를 구할 때 사용되는 문

PL/I: SUM = SUM + X(I)

SAS: SUM + X(I)

④④ 기존의 Master Aas Oile을 transaction Aas Oile을 이용하여 Update 시키고자 할 때 이용

(3) PROC Step에서 사용하는 명령어

- ① **ATTRIB** variable [FORMAT=format] [INFORMAT=informat] [LABEL='label'] [LENGTH=\$length] ... ;
① PROC에서 사용되는 변수에 대해 기존에 만들어진 FORMAT를 적용시키고자 할때 사용
- ② **BY** [DESCENDING] variable ... [NOTSORTED];
② BY 이후에 나온 변수에 의해 data set을 sort 시킬때라든가 sort된 SAS data set을 가지고 BY 변수별로 원하는 PROC을 수행시킬 수 있다.
ex) PROC MEANS ;
BY SEX ;
- ③ **CLASS** variables;
③ 수치분석에 사용되지 않는 유목 (또는 명목) 변수가 사용되어진다.
- ④ **FORMAT** variables {format} ... ;
④ 어떤 변수에 대한 정보가 출력될 때 출력되는 정보의 형태를 알려주는 문이다.
- ⑤ **FREQ** variable;
⑤ 지정된 변수에 대해 계급별 돛수 정보를 알려준다.
- ⑥ **ID** variables;
⑥ 확인변수를 어떤변수를 쓰겠다는 이용자의 의도가 있을때 사용하면 편리하며 Data set merge 등에 유용히 사용될 수 있다.
- ⑦ **LABEL** variable='label' ... ;
⑦ 변수에 Label을 주고 싶을 때

⑧ **MODEL** *dependents = independenteffects / [options];*

⑨ **OUTPUT** [*SASdataset*] ... ;

⑩ **PROC** *program* [*options*];

⑪ **VAR** *variable* ;

⑫ **WEIGHT** *variable* ;

⑧ 모형에 대한 기술을 SAS에 전하는 문

⑨ 어떤 PROC에서 얻어진 결과를 특정한 SAS data set에 담고 싶을때 사용
생략

⑩ 이용자가 원하는 SAS PROCEDURE를 부를때 사용

⑪ PROC에 적용되어지는 변수를 지정할 때 사용

⑫ 가중변수를 주고자 할 때 사용

(4) Data 및 Proc Step 모두에 사용하는 명령어

① **CLEAR** [**PAUSE**] [**SMON** | **SMOFF**];

② **CMS** [**CMScommand** | **CP** **CPcommand**];

③ * | **COMMENT** *message*; (*comment statement*)

④ **ENDSAS**;

⑤ **FILENAME** *fileref* '*filename*' ... ;

① SAS/CMS companion이나 TSO/SAS에서 화면 Clear 용으로 쓰인다 (*internal invoke*)

② SAS/CMS 하에서 CMS 명령을 줄 때 사용

③ comment 을 줄 때 사용

④ TSO/SAS 또는 SAS/CMS에서 SAS session을 끝낼때 사용

⑤ File의 지정에 사용

- ⑥ **FOOTNOTE**{n} ['text'];
- ⑦ **HELP** [keyword] [/ option];
- ⑧ **%INCLUDE** | **%INC** source ... [/ SOURCE2 | NOSOURCE2
JCLEXCL S2 = length];
- ⑨ **%LIST** [n[:m]] |[n[-m]];
- ⑩ **MACRO** name text%
 %ACTIVATE | **%ACT** macronames;
 %DEACTIVATE | **%DEACT** macronames;
 %DELETE | **%DEL** macronames;
 %MLIST | **%LISTM** macronames;
- ⑪ **OPTIONS** option ... ;
- ⑫ **PAGE**;
- ⑬ **%PUT** message;
- ⑭ **RUN** [CANCEL | QUIT];
- ⑮ **%RUN**;
- ⑯ **SKIP** n;
- ⑰ **TITLE**{n} ['title'];
- ⑱ **TSO** [TSOcommand];
- ⑲ **X** [system] ['[command]'];
- ⑥ 각주를 달고 싶을 때 사용
- ⑦ 의문이 있는 SAS 명령어에 대한 참조시 사용
- ⑧ 외부 파일에 있는 SAS Program이 들어있는 member를 포함시킬 때 사용
- ⑨ 이미 수정된 SAS program의 list를 몇 line부터 몇 line까지를 보고자 할 때 사용
- ⑩ MACRO를 만들때 사용되는 문장들
- ⑪ SAS option을 줄 때 사용
- ⑫ PAGE skip시 사용
- ⑬ MACRO에서의 PAGE SKIP시 사용
- ⑭ SAS문의 시행을 알림
- ⑮ MACRO문의 시행시 사용
- ⑯ Line skip시 사용
- ⑰ 제목을 주고자 할 때 사용
- ⑱ TSO 명령을 줄 때 사용
- ⑲ system 명령을 줄 때 사용

(5) SAS data set 정의문

```
SASdataset[(BLKSIZE=blocksize BUFNO=n  
DROP=variables FILECLOSE=position  
FILEDISP=NEW | FILEDISP=OLD FIRSTOBS=n  
GEN=n IN=variable KEEP=variables  
LABEL='label' OBS=n READ=password  
RENAME=(oldname=newname ...) REPLACE=YES |  
REPLACE=NO TRANSPORT=NO |  
TRANSPORT=YES TYPE=DATA | TYPE=CORR |  
TYPE=COV | TYPE=SSCP | TYPE=EST |  
TYPE=FACTOR | TYPE=DISCAL])  
where SASdataset is  
name | libref.name | _NULL_ | _LAST_ | _DATA_
```

SAS data set 을 Cre-
ation 시키면서 지정해
줄 수 있는 option 들
을 보여 주고 있다.

(6) SAS Option 리스트

BATCH | INTERACTIVE BAUD=*rate* BLDLTABLE |
 NOBLDLTABLE BLDL | NOBLDL BLKSIZE=*n*
 BLKSIZE(*devicetype*)=*value* BUFNO=*n*
 BYERR | NOBYERR C60 | C48 | C96 CAPS | NOCAPS
 CARDS | C=MAX | *n* CASORT | NOCASORT CENTER |
 NOCENTER CHARCODE | NOCHARCODE CHKPT |
 NOCHKPT CLIST | NOCLIST CMDMSG | NOCMDMSG
 CPSP | NOCPSP DAREAD | NODAREAD DATE | NODATE
 DAUPD | NODAUPD DEFAULT | D=*name*
 DEVICE=*name* DISK=*device* DMS | NODMS DQUOTE |
 NODQUOTE DSNFERR | NODSNFERR DSRESV |
 NODSRESV DUMP | NODUMP DUMPM | NODUMPM
 DUMPSHORT | NODUMPSHORT DUMPP | NODUMPP
 DUMPLONG | NODUMPLONG DUMPSYS | NODUMPSYS
 DYNALLOC | NODYNALLOC ERASE | NOERASE
 ERRORABEND | NOERRORABEND ERRORS=*n* FILCLR |
 NOFILCLR FILEBLKSIZE(*devicetype*)=*value*
 FILLMEM='hexvalue' FILSZ | NOFILSZ FIRSTOBS=*n*
 FMTERR | NOFMTERR FORMCHAR(*printdevice*)=('characters')
 FORTG | FORTG=*filename* FS | NOFS FSDEVICE |
 FSD=*device* FSP | NOFSP GEN=*n* ICCF | NOICCF
 IMPLMAC | NOIMPLMAC IMS | NOIMS INCLUDE |
 NOINCLUDE INITSTMT='statement'
 INVALIDDATA='character' LABEL | NOLABEL
 LAST=*SASdataset* LDISK | NOLDISK LD |
 NOLD LEAVE=*n* | *n*K LINESIZE | LS=*width*
 LOG=*fileref* LPRINT | LP | NOLPRINT | NOLP LTYPE |
 LT | NOLTYPE | NOLT MACRO | NOMACRO
 MACROGEN | NOMACROGEN MCOMPILE | NOMCOMPILE
 MEMERR | NOMEMERR MEMFILL | NOMEMFILL
 MEMRPT | NOMEMRPT MERROR | NOMERROR
 MISSING='character' MLEAVE=*n* MLOGIC |
 NOMLOGIC MODECHARS='characters' MPRINT |
 NOMPRINT MSIZE=*n* MSYMSIZE=*n* MWORK=*n*
 NAME | NA=*filename* NDSVOLS=*nnnnn* NEWS |
 NONNEWS NOTES | NONOTES NULLEOF | NONULLEOF
 NUMBER | NONUMBER OBS=*n* | MAX OFFLINE=*n.nnn*
 ONLINE=*n.nnn* OPLIST | NOOPLIST OVP | NOOVP
 PAGES | P=*n* | MAX PAGESIZE | PS=*n*
 PARM='string' PARMCARDS=*fileref* PDISK |
 PD | NOPDISK | NOPD PLIO *filename* PPRINT |
 PP | NOPPRINT | NOPP PRINTDEVICE |
 PRINTDEV=*device* PRINTHOVP | NOPRINTHOVP
 PRINTINIT | NOPRINTINIT PROBSIG=*n*
 PROCSIZE=*maximum* PSEG *value* PTYPE |
 PT | NOPTYPE | NOPT REPLACE | NOREPLACE S=*n*
 S2=S | *n* S370 | NOS370 SASHELP=*fileref*
 SASLIB *filename* SASNEWS=*member* SEQ=*length*
 SERIES *value* SERROR | NOSERROR
 SIODISK *modeletter* SKIP=*n*
 SNP | NOSNP SNPPROG | NOSNPPROG SORT=*n*
 SORTDEV=*device* SORTLIB='datasetname'
 SORTLIST | NOSORTLIST SORTMSG | NOSORTMSG
 SORTMSG=*fileref* SORTPGM='utility'
 SORTSIZE=MAX | SIZE | *n* | *n*K SORTWKDD='prefix'
 SORTWKNO=*n* SOURCE | NOSOURCE SOURCE2 |
 NOSOURCE2 SPOOL | NOSPOOL SSEG SHARED |
 NONSHARED | OFF | ON STIMER | NOSTIMER
 SVCHND | NOSVCHND SYMBOLGEN | NOSYMBOLGEN
 SYNCSORT | NOSYNCSORT SYSIN=*fileref* SYSIN |
 NOSYSIN SYSPARM='characters' TAPE=*device*
 TAPECLOSE=REREAD | LEAVE | REWIND | DISP
 TEXT82 | NOTEXT82 TIME | T=MAX | *n*
 TLINESIZE | TLS=*n* TLOG | TL | NOTLOG |
 NOTL TMSGLEV | TMSG=NOTES | NOTE |
 ERRORS | ERRO | OFF | A TPAGESIZE | TPS=*n*
 TSO | NOTSO TXTLIB | NOTXTLIB UNITS=11 12
 13 14 15 16 17 18 19 20 USER=*name*
 USERPARM='data' VIOBUF=*n* VNFERR | NOVNFERR
 VSAMLOAD | NOVSAMLOAD VSAMREAD | NOVSAMREAD
 VSAMUPDATE | NOVSAMUPDATE WORK=*name*
 WORKINIT | NOWORKINIT ZEROMEM | NOZEROMEM

SAS의 Install시 SAS system에 주어 질 수 있는 모든 option들을 보여주고 있다.

이용자의 편의에 따라 이 option들을 override시킬 수 있다.

(7) 변수표현방법

NAMES OF THE FORM...	CAN BE ABBREVIATED...	TO REPRESENT...
Numbered names of the forms X1,X2,...,Xn and Xn,Xn-1,...,X1	X1-Xn Xn-X1	all variables from X1 to Xn or Xn to X1
Ranges of names of the form X P A*	X - A	all variables from X to A
	X-NUMERIC-A	all numeric variables from X to A
	X-CHARACTER-A X-CHAR-A	all character variables from X to A
Special SAS names	_NUMERIC_	all numeric variables
	CHARACTER _CHAR_	all character variables
	ALL	all variables
Groups of names of the form ABCZ ABCY ABCX**	ABC:	all variables beginning with letter group ABC

*In variable ranges of this type, the list of variables in the range is not determined alphabetically but by the order in which they are defined.
**This form of abbreviation is available only in the PROC step.

(8) Pointer 제어기호

/ to column 1 of next line
 @n | @pointvariable to column n | column variable
 +n | +pointvariable skip n columns | variable columns
 #n | #pointvariable to line n | line variable

(9) Line Hold 에 사용되는 기호

@, trailing hold data line for next INPUT or PUT statement
 @@, trailing INPUT statement only; hold data line for other executions of DATA step
 PAGE PUT statement only; to line 1 of next page
 OVERPRINT PUT statement only; overprints previous line

(10) INPUT FORMAT 지시기호

: indicates that the value is to be read from the next nonblank column until a blank column, the length of the variable, or the end of the data line is reached.
 & indicates that a character input value may contain one or more single embedded blanks and is to be read from the next nonblank column until one of the following is encountered: two consecutive blanks, the length of the variable as first defined in the DATA step, or the end of the input line
 ? INPUT statement only: suppresses invalid data message for invalid data value
 ?? INPUT statement only: suppresses printing of error message and input lines for invalid data value

(11) 연산자의 처리순서

From first-performed to last-performed:

** prefix + prefix - ~(NOT) ><(minimum) <>(maximum)
 * /
 + -
 ||
 < <= = ^= >= > -> -<
 & (AND)
 | (OR)

*Note: if the symbol ~ is not available, use the symbol ^

(2) SAS data 입력 형식

① 수치변수

Informat	Width Range	Decimal Range	Default Width
w.	1-32		
w.d		0-31	
BZw.d	1-32	0-31	1
COMMAw.d	1-32	0-31	1
Ew.d	7-32	0-31	12
HEXw.	1-16		8
IBw.d	1-8	0-10	4
PDw.d	1-16	0-10	1
PIBw.d	1-8	0-10	1
PKw.d	1-16	0-10	1
RBw.d	2-8	0-10	4
ZDw.d			
AOS/VS, PRIMOS, VMS:	1-16	0-10	1
CMS, OS, VM/PC, VSE:	1-32	0-10	1
ZDBw.d	1-32	0-10	1

② 문자변수

Informat	Width Range	Default Width
\$w.	1-200	1*
\$CHARw.	1-200	1*
\$CHARZBw.	1-200	1*
\$HEXw.	1-200	2
\$PHEXw.	1-100	2
\$VARYINGw.	1-200	8*

*or length of variable

③ 날짜입력형식

Informat	Width Range	Default Width
DATEw.	7-32	7
DATETIMEw.d	13-40	18
DDMMYYw.	6-32	8
MMDDYYw.	6-32	6
MONYYw.	5-32	5
MSECw.	8	8
PDTIMEw.	4	4
RMFDURw.	4	4
RMFSTAMPw.	8	8
SMFSTAMPw.	8	8
TIMEw.d	5-32	8
TODSTAMPw.	8	8
TUw.	4	4
YYMMDDw.	6-32	8
YYQw.	4-32	4

(3) SAS data 출력 형식

① 수치변수

Format	Width Range	Decimal Range	Default Width	Alignment
w.	1-32			right
w.d		d < w		
BESTw.	1-32		12	right
COMMAw.d	2-32	0 or 2	6	right
DOLLARw.d	2-32	0 or 2	6	right
Ew.	7-32		12	right
FRACTw.	4-32		10	right
HEXw.	1-16		8	left
IBw.d	1-8	0-10	4	left
PDw.d	1-16	0-10	1	left
PIBw.d	1-8	0-10	1	left
RBw.d	2-8	0-10	4	left
ROMANw.	2-32		6	left
SSNw.	11		11	
WORDFw.	5-200		10	left
WORDSw.	5-200		10	left
Zw.d	1-32		1	right
ZDw.d	1-16	0-10	1	left

② 문자변수

Informat	Width Range	Default Width	Alignment
\$w.	1-200	1 or length of variable	left
\$CHARw.	1-200	1 or length of variable	left
\$HEXw.	1-200	2	left
\$VARYINGw.	1-200	8 or length of variable	left

③ 날짜변수

Format	Width Range	Default Width
DATEw.	5-9	7
DATETIMEw.d	7-40	16
DDMMYYw.	2-8	8
HHMMw.d	2-20	5
HOURLw.d	2-20	2
MMDDYYw.	2-8	8
MMSSw.d	2-20	5
MONYYw.	5-7	5
TIMEw.d	2-20	8
TODw.	2-20	8
WEEKDATEw.	3-37	29
WORDDATEw.	3-32	18
YYMMDDw.	2-8	8
YYQQ.	4-6	4

(4) SAS에서 사용가능함수

ABS (argument)
ARCOS (argument)
ARSIN (argument)
ATAN (argument)
BETAINV (p,a,b)
CEIL (argument)
CMS (command)
COLLATE (n,m,l)
COMPRESS (argument)
COMPRESS (argument1,argument2)
COS (argument)
COSH (argument)
CSS (argument,argument,...)
CV (argument,argument,...)
DATE ()
DATEJUL (Julian date)
DATEPART (datetime)
DATETIME ()
DAY (date)
DHMS (date, hour, minute, second)
DIFn (argument)
DIGAMMA (x)
DIMn (arrayname)
ERF (argument)
ERFC (argument)
EXP (argument)
FIPNAME (FIPS)
FIPNAMEL (FIPS)
FIPSTATE (FIPS)
FLOOR (argument)
FUZZ (argument)
GAMINV (p,eta)
GAMMA (x)
HMS (hour, minute, second)
HOUR (time)
HOUR (datetime)
INDEX (argument1,argument2)
INDEXC (argument1,argument2,...argumentn)
INPUT (argument,informat)
INT (argument)
INTCK (interval,from,to)
INTNX (interval,from,number)

JULDATE (date)
KURTOSIS (argument,argument,...)
LAGn (argument)
LEFT (argument)
LENGTH (argument)
LGAMMA (argument)
LOG (argument)
LOG10 (argument)
LOG2 (argument)
MAX (argument,argument,...)
MDY (month,day,year)
MEAN (argument,argument,...)
MIN (argument,argument,...)
MINUTE (time | datetime)
MOD (argument1,argument2)
MONTH (date)
N (argument,argument,...)
NMISS (argument,argument,...)
NORMAL (seed)
POISSON (lambda,n)
PROBBETA (x,a,b)
PROBBNML (p,n,m)
PROBCHI (x,df)
PROBF (x,ndf,ddf)
PROBGAM (x,eta)
PROBHYP (nn,k,n,x,or)
PROBIT (argument)
PROBNEGB (p,n,m)
PROBNORM (x)
PROBT (x,df)
PUT (argument,format)
QTR (date)
RANBIN (seed,n,p)
RANCAU (seed)
RANEXP (seed)
RANGAM (seed,alpha)
RANGE (argument,argument,...)
RANNOR (seed)
RANPOI (seed,lambda)
RANTBL (seed,p1,...pn)
RANTRI (seed,h)
RANUNI (seed)
REPEAT (argument1,n)
REVERSE (argument)

RIGHT (argument)
ROUND (argument,roundoffunit)
SASVER ()
SCAN (argument1,n[,delimiters])
SECOND (time | datetime)
SIN (argument)
SINH (argument)
SKEWNESS (argument,
argument, argument,...)
STD (argument,argument,...)
STDERR (argument,argument,...)
STFIPS (postalcode)
STNAME (postalcode)
STNAMEL (postalcode)
SUBSTR (argument1,position,n)
SUBSTR (argument1,position,n)=x;
SUM (argument,argument,...)
SYMGET (variable)
SYSPARM ()
TAN (argument)
TANH (argument)

TIME ()
TIMEPART (datetime)
TODAY ()
TRANSLATE (argument1,to,from,
...,to,from)
TRIM (argument)
TSO (command)
UNIFORM (seed)
UPCASE (argument)
VERIFY (argument1,argument2,
...,argumentn)
USS (argument,argument,...)
VAR (argument,argument,...)
WEEKDAY (date)
YEAR (date)
YYQ (year,quarter)
ZIPFIPS (zipcode)
ZIPNAME (zipcode)
ZIPNAMEL (zipcode)
ZIPSTATE (zipcode)

위에 열거한 함수들은 SAS system에서 제공되는 library 내장 함수들이며 통계분석상 다른 일반 language에서도 찾아볼 수 없는 유용한 함수들이 많이 들어 있어서 통계 package로서 다른 여러종류보다 뛰어난 장점중에 한가지가 되고 있다.

(15) Function Key

① Function Key

KEY	COMMAND	KEY	COMMAND
01	HELP	13	HELP
02	SPLIT	14	SPLIT
03	SUBMIT	15	SUBMIT
04	RECALL	16	RECALL
05	RFIND	17	RFIND
06	RCHANGE	18	RCHANGE
07	BACKWARD	19	BACKWARD
08	FORWARD	20	FORWARD
09	OUTPUT	21	OUTPUT
10	LEFT	22	LEFT
11	RIGHT	23	RIGHT
12	CURSOR	24	CURSOR

위에 열거한 것을 보면 TSO의 Function Key와 거의 유사함을 알 수 있다. 다만 SAS에서 특이한 Function Key로서는 04(Recall)과 03(Submit)인데 Recall Key는 바로전에 Submit시킨 SAS Program을 부르는 것이고 Submit Key는 완성된 Program을 수행시킬 때 쓰게된다. 그리고 이러한 Function Key의 기능을 이용자의 취향에 따라 Setting시킬 수도 있다.(SAS manual 참고)

(16) Command

SAS Display Manager에서의 Command는 TSO에서의 Command와 거의 비슷한 것을 갖는다.

① Line Commands

A, B : target position, A(After) or B(Before), of an
M, MM, C, CC or COPY Command.

C : copy a line to the location indicated either by
an A(After) or B(Before) line command.

D[n] : Delete

I : Insert

M : Move

R[n] : Repeat

TF[n] : 뒷 line을 앞 line의 끝에 밀착시키고자 할 때

TS[n] : line split을 시켜서 새로운 정보를 추가하고자 할 때

CC : Block Copy

DD : Block Delete

MM : Block Move

RR : Block Repeat

>n : Data를 n Column 오른쪽으로 이동

<n : " " 왼쪽 "

»n : Data line Block을 Column 오른쪽으로 이동

«n : " " 왼쪽 "

(>, < 대신), (을 쓰면 이동이 아니고 Delete가 된다.)

② 그림의 Command

- Scrolling Commands

AUTOADD	LEFT
BACKWARD	<i>n</i>
FORWARD	RIGHT
TOP	HSCROLLC
BOTTOM	VSCROLL

- File Management Commands

COPY	END	INCLUDE
------	-----	---------

- General Editing Commands

CHANGE	CAPS ON/OFF
RCHANGE	CPRO
DES	CUNPRO
FILL	CURSOR
FIND	NULLS ON/OFF
RFIND	NUMS ON/OFF
BOUNDS	PREVCMD
CANCEL	RESET
COPY	SAVE

- Help, Function Keys, and Host-level Commands

HELP	KEYS	X command
------	------	-----------

(17) MACRO 어 문장

```
%MACRO name[(parameter[,...])][ / STMT];  
  %MEND[name];  
  %CMS command;  
  %* comment;  
  %DO;  
  %DO macrovariable = start %TO stop [%BY increment];  
  %DO %UNTIL(expression);  
  %DO %WHILE(expression);  
  %END;  
  %GLOBAL macrovariables;  
  %GOTO | %GO TO label;  
  %IF expression %THEN statement;  
    [%ELSE statement;]  
  %INPUT [macrovariables];  
  %label: statement;  
  %LET macrovariable = [value];  
  %LOCAL macrovariables;  
  %PUT text;  
  %TSO command;
```

(18) MACRO 어 함수

```
%BQUOTE(argument)  
%EVAL(expression)  
%INDEX(argument1, argument2)  
%LENGTH(argument)  
%NRBQUOTE(argument)  
%NRQUOTE(argument)  
%NRSTR(argument)  
%QUOTE(argument)  
%SCAN(argument, n[, delimiters])  
%STR(argument)  
%SUBSTR(argument, position[, length])  
%UNQUOTE(argument)  
%UPCASE(argument)
```

(19) MACRO 어와 함께 DATA STEP에서 사용되는 함수

```
SYMPUT(argument)  
SYMPUT(argument1, argument2)
```

(20) 자동 MACRO 변수들

SYSBUFFR
SYSDATE
SYSDAY
SYSDEVIC
SYSDSN
SYSINDEX
SYSENV
SYSPARM
SYSRC
SYSSCP
SYSTIME
SYSVER