SAS 소개

Park Beomjin[1]
[1] *University of Seoul*

1 SAS

- SAS(Statistical Anlysis System)는 고급 분석, 다변량 분석, 데이터 관리, 예측 분석 등을 위한 소프트웨어.

2 SAS 언어

- SAS언어는 다른 자연언어나 computer 언어와 마찬가지로 고유의 어휘와 문법체계를 가지고 있으며 이러한 어휘와 문법체계로 작성된 일련의 SAS문장을 SAS Program 이라 한다.
- SAS언어는 statements, expressions(수식), options, formats, functions(함수) 등으로 구성되어 있다.
- Expressions (수식)
  - 산술연산자 : **, *, /, +, -
  - 비교연산자 : =, ∧=, >, <, >=, <=
  - 논리연산자 : AND &, OR |, ∧ NOT
- Function (함수)
  - 산술 : ABS, LOG, MAX, EXP. ...
  - 삼각 : COS, SIN, TAN, ...
  - 통계 : MEAN, SUM, POISSON, ...
  - 난수 : RANNOR, RANUNI, ...

3 SAS program의 구성

- DATA Step (자료입력단계)

  - Raw data나 이미 만들어진 SAS dataset을 읽어 하나의 새로운 SAS dataset을 만드는 과정, DATA keyword로 시작하여, 일련의 statements로 구성되며 RUN으로 끝난다.
  - 이 단계에서 자료의 입력 및 생성, 자료 값 변환, 자료선택, 자료검색 등이 이루어진다.

- PROC Step (자료분석단계)
  - 만들어진 SAS dataset을 사용해 자료를 처리하고 분석한다.

- Step (단계)은 새로운 step이나 RUN 명령을 만날 때 끝난다. PROC step이 나오면 DATA step은 끝이 난다. PROC step은 RUN 명령을 만나면 끝난다.

```
DATA distance;
miles = 26.22;
kilometer = 1.61 * miles; /*곱하기*/
coef = kilometer / miles; /* 나누기*/
log = log(coef); /*log 함수*/
PROC PRINT data = distance;
RUN;
```

4 SAS dataset

- SAS에서 dataset에 대한 접근, 즉 데이터를 읽거나 구동하기 위해 프로그램을 작성하기 전 먼저 작업환경(library)을 설정하면 지정된 작업환경 내에서 쉽게 프로그램을 구동 할 수 있다.

```
libname database "C:/Users/User/desktop/";
options validvarname = any;
/* database라는 이름으로 작업환경의 경로를 설정
옵션은 한글 변수를 사용할 수 있게 하는 옵션임 */
```

- SAS dataset은 descriptor portion (요약정보영역)과 data portion (데이터영역)으로 구성된 파일이다.

- 요약정보영역 : SAS dataset에 대한 정보 (dataset 이름, 관측치 수, 변수속성 등)를 포함한다. 요약정보영억을 보기 위해서는 PROC contents를 사용한다.

```
PROC contents data = distance;
RUN;

PROC contents data = Sasuser.Iris_dat;
RUN;
```

- 데이터영역 : Table 형태로 배열된 자료값의 집합. 데이터영역을 보기 위해서는 PROC PRINT, Viewtable 등을 사용 할 수 있다.

```
PROC PRINT data = distance;
RUN;

PROC PRINT data = Iris_dat;
RUN;
```

- 변수의 속성 : 요약정보영역은 dataset에 대한 일반적인 정보 및 dataset이 보유하고 있는 각 변수의 속성에 대한 정보를 포함하고 있다.

```
PROC contents data = Sasuser.Iris_dat;
RUN;
```

  - Data type : numeric (숫자), character (문자)
  - Variable type : numeric variable, character variable (변수 선언 뒤 $표시로 숫자변수와 구별)
  - Missing data : numeric data는 single period(.), character data는 blank로 표현됨

5 SAS program 작성방법

- 하나의 SAS 문장은 keyword, SAS 이름 (변수, filename), 연산자, 특수문자로 구성된다.
- SAS의 keyword는 SAS문의 종류를 구별하기 위해 SAS가 규정한 단어로 DATA, FORMAT, IF, PRINT, PROC 등이 있다.
- 항상 SAS keyword로 시작하여 모든 문장의 끝은 세미콜론(;)으로 끝난다.
- 한 SAS문은 여러 줄에 걸쳐 작성할 수 있으며 한 줄에 두 개 이상의 SAS문이 나올 수 있고 문장 사이는 반드시 세미콜론(;)으로 구분한다.

- 주석은 별표(*)로 시작하고 세미콜론(;)으로 끝난다. 여러 문장으로 구성된 주석은 /* 로 시작하고 */로 끝난다.

```
PROC contents data = Sasuser.Iris_dat;
RUN;
/* 데이터의 요약정보를 보여준다 */
```

- 변수명 단축용법

  - 숫자범위로 지정 : variable을 생성하거나 지정하고자 할 때, 필요한 변수이름을 일일이 나열하기가 번거로울 때 번호 붙은 변수를 사용하여 변수들을 통칭하는 단축용법.

    qtr1 qtr2 qtr3 qtr3 -> qtr1-qtr4

  - 변수 범위로 지정 : X – A는 변수 X와 A를 포함한 두 변수 사이의 모든 변수를 통칭하여 언급하고자 할 때 사용

```
PROC means data = Sasuser.Iris_dat;
var Sepal_Length -- Petal_Length;
RUN;
```

- 데이터 입력

  - DATA step에서 데이터를 직접 입력하여 SAS 데이터셋으로 만들 수 있다.

```
data one; /* 생성할 데이터셋의 이름 */
input ph        Time        Temperature; /* 입력할 데이터의 변수 */
cards;
        4.5         20                  125
        4.1         22                  133
        2.8         18                  149
        4.0         26                  120
        5.0         25                  120
        6.0         21                  138
;
run;
```

- Examples 1

```
data package;
input pack pack2@;
cards;
15 8 10  21
17 13  19 14
12  14 7 18
;
run;

data package2;
input pack pack2@@;
cards;
15 8 10  21
17 13  19 14
12  14 7 18
;
run;
```

- Examples 2

```
data one;
input name $ mid final hw1-hw6 atten;
if hw2 = . then hw2 = 0;
if hw6= . then hw6=0;
score = mid*0.4 + final*0.45  + mean(of hw1-hw6)*0.1 + atten*0.05;
if score >= 80 then total = "pass";
else total = "fail";
cards;
김부산        83 92 10  8  9  8 10 10 20
이전주        66 93 10  8 10  8  9  . 20
오대구        88 94 10 10 10  8 10 10 20
남궁전        62 94  8  . 10  8 10 10 19
백김천        82 98 10 10  8  8 10 10 20
신제주        42 43 10  8  9  8  6 .  15
;
run;
```

- 데이터 읽기

  - 외부 데이터를 읽기 위해서는 PROC IMPORT문을 사용한다.

  - csv 파일 읽기

```
PROC IMPORT datafile = "파일명.csv" dbms = csv replace
out = csvfile;
      getnames = yes;
      guessingrows = 1000;
RUN;
/* dbms는 읽는 파일의 형식을 입력함(ex:csv, xls 등)
out으로 저장할 파일명을 설정할 수 있음
repalce는 이미 생성된 파일에 덮어 쓸 수 있도록 하는 옵션
getnames를 통해 열 이름을 저장할 수 있음
guessingrows를 통해 입력된 숫자의 행까지 각 열이 어떤 형식으로
작성되었는지 파악하여 저장 */
```

  – xls 파일 읽기

```
PROC IMPORT datafile = "파일명.xls" dbms = xls replace
out = xlsfile;
      namerow = 1;
      startrow = 2;
      getnames = yes;
      guessingrows = 1000;
RUN;
/* namerow 옵션을 통해 열이름이 작성된 행의 번호를 입력할 수 있음
startrow 옵션을 통해 데이터의 시작 행번호를 입력할 수 있음
dbms의 종류에 따라 설정할 수 있는 옵션이 다르기 때문에 주의해야함.*/
```

  - EX1 : iris.xlsx 파일 읽기

```
PROC IMPORT datafile = "C:\Users\User\Desktop\19학년도 회귀분석 및 실습\
iris.xlsx" dbms = xlsx replace
out = iris_dat ;
getnames = yes;
RUN;
```

- 데이터 정렬하기

  - 데이터를 특정 기준을 통해 정렬하기 위해서는 PROC SORT를 사용한다.
  - 정렬 변수가 문자형일 때

```
PROC sort data = iris1 out = sorted_iris;
by Sepal_Length;
RUN; QUIT;

PROC sort data = iris1 out = sorted_iris;
by descending Sepal_Length;
RUN; QUIT;

PROC sort data = sorted_iris out = sorted_iris;
by Species;
RUN; QUIT;
```

  - 정렬 변수가 두개 이상일 때

```
PROC sort data = iris1 out = sorted_iris;
by descending Sepal_Length Sepal_Width;
RUN;
```

- 조건문을 통한 변수 생성
  - 데이터에서 조건문을 통해 새로운 변수를 생성할 수 있다.

```
DATA iris_dummy;
set iris1;
if Species = "setosa" then dummy = 1;
else dummy = 0;
RUN;
```

- 데이터 병합
  - 두개의 data를 열 이름 기준으로 행으로 결합할 때

```
PROC IMPORT datafile = "C:\Users\User\Desktop\19학년도 회귀분석 및 실습
\Data\iris1.csv" dbms = csv replace
out = example.iris1;
getnames = yes;

PROC IMPORT datafile = "C:\Users\User\Desktop\19학년도 회귀분석 및 실습
```

```
\Data\iris2.csv" dbms = csv replace
out = iris2;
getnames = yes;

DATA iris_rbind;
set iris1 iris2;
RUN;
```

– 두개의 data를 특정 변수를 기준으로 결합할 때

```
PROC IMPORT datafile = "C:\Users\User\Desktop\19학년도 회귀분석 및 실습\
Data\iris3.csv" dbms = csv replace
out = example.iris3;
getnames = yes;

PROC IMPORT datafile = "C:\Users\User\Desktop\19학년도 회귀분석 및 실습\
Data\iris4.csv" dbms = csv replace
out = example.iris4;
getnames = yes;

PROC sort data = iris3 out = sorted_iris3;
by Species;

PROC sort data = iris4 out = sorted_iris4;
by Species;
RUN;

data iris_merge;
merge sorted_iris3 sorted_iris4;
by Species;
RUN;

DATA iris_merge2;
merge iris3 iris4;
RUN;
```

- 데이터 쓰기

– 데이터는 PROC EXPORT를 통해 다른 형식으로 저장 할 수 있다.

```
PROC EXPORT data = iris_merge dbms = xlsx replace
outfile = "C:\Users\User\Desktop\merged_iris.xlsx";
RUN;

PROC EXPORT data = iris_merge dbms = csv replace
outfile = "C:\Users\User\Desktop\merged_iris.csv";
RUN;
```