4.

수식과 함수

DATA 단계에서 자료 처리에 활용되는 SAS 수식과 함수들에 대해 소개한다.

4.1 SAS 수식

수식(expression)은 연산자와 피연산자의 조합.
연산자는 +, -, *, / 등 특수문자 연산자, 괄호, SAS 고유의 함수(function) 등.
피연산자는 상수(constant)와 변수이름(variable name).

(예1) 1
　　아무 연산자도 사용되지 않은 듯하나 + 라는 연산자가 생략된 형태
(예2) score+10
(예3) total/n
(예4) SIN(3.1415926)
(예5) age<3 OR age>80
(예6) 1/EXP(a*b)

⇨ (예1)~(예4)는 연산자가 하나 사용된 단순수식
　(예5)~(예6)은 둘 이상의 연산자가 사용된 복합수식
[참고] 연산 우선 순위는 일반 컴퓨터 언어와 동일

(예1) 1, +1, -2, 1.234, 1.234E5, 1.234E-5
(예2) amen='N. K. Sung';
　　name="Sung's";　　(cf) title='이것은 "의자"이다';

⇨ (예1) 숫자상수의 예들
　　E는 10의 지수를 나타내는 것으로, 1.234E5는 $1.234X10^{5}$,
　　$1.234E-5$는 $1.234X10^{-5}$과 동일.
　(예2) 문자상수의 예들 (name 이라는 변수에 문자상수 값 할당)
　　문자상수는 최대 32,767자까지 사용 가능.
　　문자상수 좌우에는 홑따옴표 또는 겹따옴표를 붙이는데,
　　문자상수에 이미 **홑따옴표**가 포함되어 있으면 반드시 **겹따옴표** 사용해야 함.

4.2 SAS 연산자

4.2.1 산술 연산자 (arithmetic operator)

산술연산자	기능
+	더하기
-	빼기
*	곱하기
/	나누기
**	지수

4.2.2 비교 연산자 (comparison operator)

비교연산자		비교 기능
기호	단축문자	
=	EQ	같다 (EQual to)
^= 또는 ~=	NE	같지 않다 (Not Equal to)
>	GT	크다 (Greater Than)
<	LT	작다 (Less Than)
>=	GE	크거나 같다 (Greater than or Equal to)
<=	LE	작거나 같다 (Less than or Equal to)
	IN	어느 하나와 같다

예제참고

(예) DATA one;

```
   a=7;
   b=a=7;  /* 결과적으로 b=1 */  /* b=(a=7) */
   c=a=6;  /* 결과적으로 c=0 */  /* c=(a=6) */
RUN;
PROC PRINT DATA=one;  RUN;
```

⇒

OBS	a	b	c
1	7	1	0

a=7; 에서는 a 라는 변수에 7을 할당.

b=a=7; 에서 a=7 은 할당하는 것이 아닌 수식.

첫 번째 등호는 등호 오른쪽의 연산결과(a=7의 결과)를 등호 왼쪽의 변수(b)에 넣는 것으로, 비교연산자로 이루어진 수식인 a=7을 계산하여 그것이 참이면 진리값 1이 할당되고, 거짓이면 0이 할당됨.
즉, a=7 이 참이므로 변수 b 는 1 값이 할당됨.

c=a=6; 에서 a=6 은 할당하는 것이 아닌 수식.

위와 같은 방식으로, 비교연산자로 이루어진 수식인 a=6을 계산하여 그것이 참이면 진리값 1이 할당되고, 거짓이면 0이 할당됨.
즉, a=6 이 거짓이므로 변수 c 는 0 값이 할당됨.

(예) IF x < y THEN z = 5;
ELSE z = 9;
⇒ 만약 x 가 y 보다 작으면(즉, 'x < y'의 진리값이 1(참)이면) z에 5를 넣고, 그렇지 않으면(즉, 'x < y'의 진리값이 0(거짓)이면) z에 9를 넣음.

(예) z = 5*(x<y) + 9*(x>=y);
⇒ 만약 'x<y'가 참이면 z 에는 5가, 'x>=y'가 참이면 z 에는 9가 할당됨.

(예) IF lastname IN ('Kim', 'Lee', 'Park') THEN ... ;
⇔ IF lastname='Kim' OR lastname='Lee' OR lastname='Park' THEN ... ;
⇒ 비교연산자 IN 은 괄호 속에 주어진 여러 개의 값들 중에서 어느 하나와 일치하는 경우에 진리값이 1이 됨
즉, lastname 이란 변수값이 'Kim' 또는 'Lee' 또는 'Park'인 관측들만 골라 어떤 연산을 하고자 하는 것임.

expression IN (value1, value2, ...)

IN 왼쪽에는 적당한 수식, 오른쪽의 괄호 안에는 수식과 같은지 비교할 상수값들을 나열하되, 상수값이 문자라면 따옴표로 둘러치고, 상수값 사이에는 쉼표를 사용

(예) exam1, exam2, exam3 의 합이 0점 또는 100점인 관측들만 고르려면?
⇒ IF exam1+exam2+exam3 IN (0, 100) THEN ..... ;

(예) age 가 20 이하면 그룹 1에, 21 이상이고 30 이하면 그룹 2에, 31 이상이고 40 이하면 그룹 3에, 그 이후는 그룹 4에 배정하려면?
⇒ age_group=1;
IF 20 < age <= 30 THEN age_group=2;
IF 30 < age <= 40 THEN age_group=3;
IF 40 < age THEN age_group=4;
⇔ age_group = 1 + (age>20) + (age>30) + (age>40);

(예) DATA;
INPUT x $;
IF x = :"R" THEN PUT _ALL_; /* PUT 은 Log 창에 출력하라는 의미 */
DATALINES;
R1
X2

R3

RUN;

⇒ 로그창에는 R1 과 R3 만 출력됨.

비교연산 x=:"R" 의 의미는 문자 변수 x의 첫 글자가 R 과 동일한지 여부를 확인하라는 의미임. 실제 데이터에는 R1, X2, R3 의 세 개의 관측치와 x 라는 한 개의 변수가 포함되어 있음.

4.2.3 논리 연산자 (logical operator)

논리연산자		기능과 진리값
기호	단축문자	
&	AND	좌우 수식이 둘 다 참이면 1, 아니면 0
|	OR	좌우 수식이 둘 다 거짓이면 0, 아니면 1
^ (또는 ~)	NOT	우측 수식이 참이면 0, 거짓이면 1

[참고] NOT 다음에 따르는 수식에는 괄호 사용

(예) a=1; b=2; c=3;

d=a<b & b<c;

e=a<b AND b>c;

f=a<b OR b>c;

g=a>b | b>c;

h=^(a<b);

⇒ a=1, b=2, c=3, d=1, e=0, f=1, g=0, h=0

4.2.4 기타 연산자

기타 연산자	기능
><	최소(minimum)
<>	최대(maximum)
||	문자결합(concatenation)

(예) a=-2<>3; b=-2><3;

⇒ a=-3, b=-2

(예) a=(-2)<>3; b=(-2)><3;

⇒ a=3, b=-2

(예) d = "(02)" || "420" || "-5304";

⇒ (02) 420-5304

(예) a="Hong"; b="KilDong"; name=a || b;

⇒ name 값은 HongKilDong

4.3 SAS 함수

: 자주 필요한 값들을 간편히 구할 수 있도록 만든 일종의 독립된 프로그램

Function (argument , argument , ...)

인수의 개수는 함수에 따라 정해져 있기도 하고 사용자가 필요한대로 나열할 수도 있음. **쉼표로 인수 사이 구분**

(예) y = SQRT(x); (Square root)

⇒ 숫자변수 x의 제곱근을 구해서 그 결과를 y에 할당.
SQRT는 함수 이름, x는 인수, SQRT 함수는 한 개의 인수만 가짐.

(예) z = MOD(10, 3);

⇒ z = 1 (= 10을 3으로 나눈 나머지)
MOD는 나머지를 구하는 함수로, 제수와 피제수에 해당하는 두 개의 인수를 쉼표(,)로 구분하여 지정.

Function (OF argument1 argument2 ...)

OF를 사용할 때는 인수 사이에 **쉼표가 아닌 빈칸으로 구분**하여 지정함

(예) average = MEAN(x1, x2, x3, x4, x5);

⇒ average = x1, x2, x3, x4, x5 의 평균값
MEAN 은 괄호 안에 지정한 인수들의 평균을 구하는 함수로 인수의 개수는 유동적이며 인수 간에는 쉼표로 구분하여 지정.

⇔ average = MEAN(OF x1 x2 x3 x4 x5);

⇔ average = MEAN(OF x1-x5); /* MEAN(x1-x5) 는 안됨 */

인수의 개수가 유동적일 때는 OF를 사용하여 변수 지정의 단축용법을 활용할 수 있는데, 인수 간에는 쉼표가 아닌 빈칸으로 구분하여 지정.

이것은 'x1-x5' 의 평균을 구하라는뜻(한개의수)

(예) x = ABS(-3); absolute value

⇔ a = -3; x = ABS(a);

⇒ x=3
ABS는 절대값을 구하는 함수.

4.3.1 SAS 함수 일람

: SAS 시스템에서 지원하는 함수의 범주

- 산술함수(arithmetic function)
- 비트별 논리함수(bitwise logical function)
- 문자함수(character function)
- 문자열 일치함수(character string matching function)
- 데이터 셋트함수(data set function)
- 날짜 및 시간함수(date and time function)
- DBCS 함수(DBCS function)
- 외부파일함수(external file function)
- 금융함수(financial function)
- 라이브러리 및 카탈로그함수(library and catalog function)
- 수리함수(mathematical function)
- 비중심함수(noncentrality function)
- 분위수함수(quantile function)
- 확률 및 밀도함수(probability and density function)
- 난수함수(random number function)
- 표본통계량함수(sample statistic function)
- SAS 파일 입출력함수(SAS file I/O function)
- 특수함수(special function)
- 삼각함수(trigonometric function)
- 절단함수(truncation function)
- 변수정보함수(variable information function)

4.3.2 산술 함수 (arithmetic function)

함수 이름	기능
ABS	절대값(absolute value)
MAX	최대값(maximum)
MIN	최소값(minimum)
MOD	나머지(remainder 또는 modulus)
SIGN	부호(sign) +/0/−
SQRT	제곱근(square root)

ABS(*argument*)

주어진 인수의 절대값 계산

(예) x = ABS(-3); ⇒ x=3

MAX(*argument, argument, ...*)

주어진 인수들 중 최대값 계산

(예) x = MAX(-1, 0, 1); ⇒ x=1

MIN(*argument, argument, ...*)

주어진 인수들 중 최소값 계산

(예) x = MIN(-1, 0, 1); ⇒ x=-1

MOD(*argument1, argument2*)

argument1을 argument2로 나누었을 때의 나머지 값 계산

(예) x=MOD(10, 5); y=MOD(10,3); z=MOD(10,1.6);

⇒ x=0, y=1, z=0.4

SIGN(*argument*)

인수의 부호를 확인하는 함수

인수의 값이 0보다 크면 1, 같으면 0, 작으면 -1

(예) x = SIGN(-2); ⇒ x=-1

SQRT(*argument*)

제곱근 계산하는 함수로, 인수는 0보다 커야함

(예) x = SQRT(4); ⇒ x=2

4.3.3 절단 함수 (truncation function) 소숫점자르기

함수 이름	기능
CEIL	인수 이상의 최소 정수
FLOOR	인수 이하의 최대 정수
INT	인수에서 소숫점 이하를 절단한 정수
ROUND	지정된 자릿수에서 반올림(rounding)

CEIL(*argument*)

주어진 인수 **이상**의 최소 정수 계산

(예) x=CEIL(2.2); y=CEIL(-2.2); z=CEIL(2); ⇒ x=3, y=-2, z=2

FLOOR(*argument*)

주어진 인수 **이하**의 최대 정수 계산

(예) x=FLOOR(2.2); y=FLOOR(-2.2); z=FLOOR(2); ⇒ x=2, y=-3, z=2

INT(*argument*)

주어진 인수의 값에서 소숫점 이하를 **자른** 정수 부분

(예) x=INT(1); y=INT(1.3); z=INT(-1.6); ⇒ x=1, y=1, z=-1

(예) q=INT(0.999999999999); r=INT(0.99999999999) ⇒ **q=1**, r=0

[참고] 컴퓨터 기억 용량 및 연산 처리 방식의 제약으로 인해, 임의의 정수 근방 10^{-12} 이내에 있으면 반올림 처리됨

ROUND(*argument , roundoffunit*)

주어진 인수를 가장 가까운 **반올림단위**(roundoff unit)에서 **반올림**

(예) x=ROUND(123.456, 1); **또는** x=ROUND(123.456); ⇒ x=123

y=ROUND(123.456, 100); 일의자리까지 보일수있게 ⇒ y=100

z=ROUND(123.456, 0.01); ⇒ z=123.46

4.3.4 수리 함수 (mathematical function)

함수 이름	기능
ERF	오차함수(error function)
ERFC	오차함수의 여함수. 즉 1-ERF
EXP	지수함수(exponential function)
GAMMA	감마함수(gamma function)
LOG	자연로그(natural logarithm)
LOG2	밑이 2인 로그
LOG10	상용로그(common logarithm)

EXP(*argument*)

주어진 인수만큼 e의 거듭 제곱 계산 (exponential(지수함수))

(예) x=EXP(0); y=EXP(1); ⇒ x=1, y=2.71828

LOG(*argument*)

자연로그 계산, 인수는 0 보다 커야함

(예) e=EXP(1); x=LOG(e); y=LOG(1); z=LOG(10);

⇒ e=2.71828, x=1, y=0, z=2.30259

LOG10(*argument*)

상용 로그 값 계산, 인수는 0 보다 커야함

(예) x=LOG10(1); y=LOG10(10); ⇒ x=0, y=1

4.3.5 삼각 함수 (trigonometric function)

함수 이름	기능
COS	코사인(cosine)
SIN	사인(sine)
TAN	탄젠트(tangent)
ARCOS	역코사인(arc cosine)
ARSIN	역사인(arc sine)
ATAN	역탄젠트(arc tangent)
COSH	쌍곡코사인(hyperbolic cosine)
SINH	쌍곡사인(hyperbolic sine)
TANH	쌍곡탄젠트(hyperbolic tangent)

COS(*argument*)

코사인 값 계산, 인수 단위는 radian

(예) pi=3.14159265359; x=COS(pi/3); ⇒ x=0.5

SIN(*argument*)

사인 값 계산, 인수 단위는 radian

TAN(*argument*)

탄젠트 값 계산, 인수 단위는 radian

4.3.6 특수 함수 (special function)

함수 이름	기능
LAG	지체된 값(lagged value)
DIF	현재 값과 지체된 값의 차이(difference)

4.3.7 표본통계량 함수 (sample statistic function)

함수 이름	기능
CSS	수정제곱합(corrected sum of squares)
CV	변동계수(coefficient of variation)
KURTOSIS	첨도(kurtosis)
MAX	최대값(maximum)
MIN	최소값(minimum)
MEAN	산술평균(arithmetic mean)
N	비분실값(non-missing values)의 개수
NMISS	분실값(missing values)의 개수
RANGE	범위(range)
SKEWNESS	왜도(skewness)
STD	표준편차(standard deviation)
STDERR	표준오차(standard error)
SUM	합(sum)
USS	비수정제곱합(uncorrected sum of squares)
VAR	분산(variance)

- 표본통계량함수 문법은 동일하며, 인수들을 지정할 때 OF를 이용한 축약 형식 가능
- DATA 단계에서 함수를 이용한 표본통계량 계산 보다는 **PROC 단계에서 MEANS 또는 UNIVARIATE 절차**를 이용하여 한꺼번에 계산하는 것이 더 효율적임

function (*argument, argument, ...*)

(예) MEAN(*argument, argument, ...*): 주어진 인수들의 산술 평균 계산

```
average = MEAN(1,2,3);
⇔ x1=1;  x2=2;  x3=3; average=MEAN(OF x1-x3);
⇒  average=2
```

(예) SUM(*argument, argument, ...*): 주어진 인수들의 합 계산

```
total = SUM(1, 2, 3);  ⇒  total=6
```

4.3.8 분포 함수 (distribution function)

함수 이름	기능
POISSON	포아송(Poisson) 분포함수
PROBBETA	베타(beta) 분포함수
PROBBNML	이항(binomial) 분포함수
PROBCHI	카이제곱(chi-square) 분포함수
PROBF	F 분포함수
PROBGAM	감마(gamma) 분포함수
PROBHYPR	초기하(hypergeometric) 분포함수
PROBNEGB	음이항(negative binomial) 분포함수
PROBNORM	정규(normal) 분포함수
PROBT	t 분포함수

확률변수 X의

분포함수(또는 누적분포함수(cdf: cumulative distribution function))는

$$F(x) = P[X \le x]$$

X가 연속형 변수이면, 확률밀도함수(pdf: probability density function)는

$$f(x) = F'(x) = \frac{dF(x)}{dx}$$

이산형 변수이면, 확률질량함수(pmf: probability mass function)는

$$f(x) = F(x) - F(x-) \ , \quad \text{where } F(x-) = \lim_{0 < h \to 0} F(x-h)$$

(예) POISSON(*lambda, x*): 포아송 분포에 대한 cdf 값

(예) PROBBNML(*p, n, x*): 이항 분포에 대한 cdf 값

(예) PROBF(*x, ndf, ddf*): F 분포에 대한 cdf 값

(예) PROBNORM(*x*): 표준정규 분포에 대한 cdf 값

(예) PROBT(*x, df*): t 분포에 대한 cdf 값

4.3.9 분위수 함수 (quantile function)

함수 이름	기능
BETAINV	베타 분포의 분위수
CINV	카이제곱 분포의 분위수
FINV	F 분포의 분위수
GAMINV	감마 분포의 분위수
PROBIT	정규 분포의 분위수
TINV	t 분포의 분위수

확률변수 X, 또는 분포함수 F의 p-번째 분위수 q는 다음 조건을 만족하는 가장 작은 x 값 (단, $0 \le p \le 1$)

$$F(x) \ge p$$

FINV(*p, ndf, ddf*)

F 분포의 p-번째 분위수 계산

ndf는 분자의 자유도, ddf는 분모의 자유도

(예) q = FINV(0.95, 2, 10); ⇒ q=4.1028

PROBIT(*p*)

표준정규분포의 p-번째 분위수 계산

(예) q = PROBIT(0.975); ⇒ q=1.96

TINV(*p, df*)

t 분포의 p-번째 분위수 계산

df는 자유도

(예) q = TINV(.95, 2); ⇒ q=2.92

4.3.10 문자 함수 (character function)

함수 이름	기능
COMPRESS	빈칸 등 삭제
SUBSTR	

[참고] 결과를 받을 문자변수 길이가 길더라도 미리 포맷을 지정하지 않으면 문자변수의 디폴트 길이인 8 바이트가 자동 할당되므로 DATA 단계에서 FORMAT을 선언해야 함

FORMAT *variable* $*w.*;

(예) FORMAT a $40.; /* 5장, 6장 참고 */

COMPRESS(*source*)

압축, 예를 들어, 문자열에 존재하는 빈칸 제거

(예) a="S u n g"; b=COMPRESS(a); ⇒ b=Sung

(예) b=COMPRESS("S u n g"); ⇒ b=Sung

SUBSTR(*variable, position, length*) = *characters*

등호(=) 좌측에 사용하여 **문자열의 일부를 다른 문자로 대체**

variable = SUBSTR(*variable, position, length*)

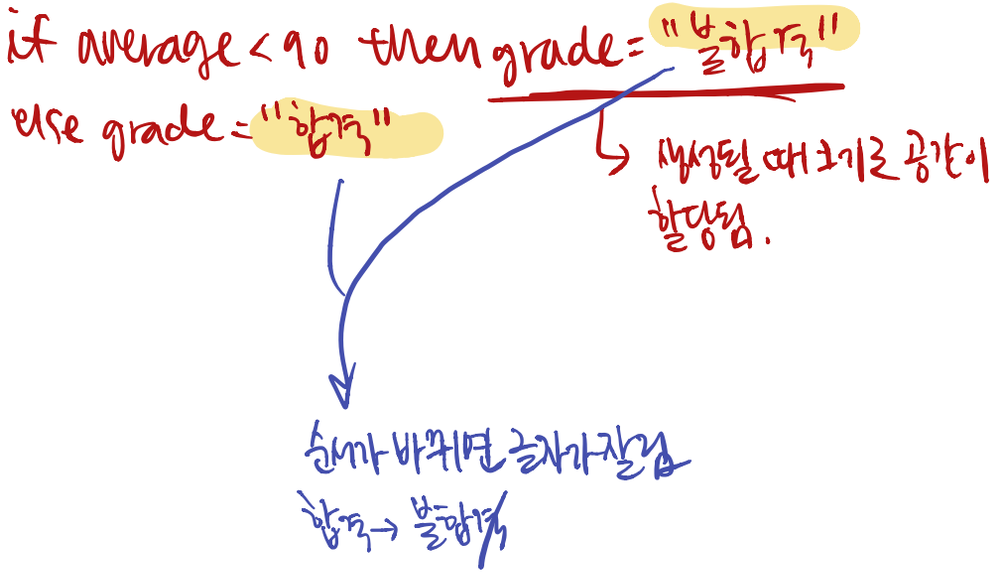
등호(=) 우측에 사용하여 **문자열 중 일부를 추출**

(예) a="MINJI"; SUBSTR(a, 4, 1)="Z"; ⇒ a=MINZI

(예) a="MINJI"; SUBSTR(a, 4)="Z"; ⇒ a=MINZ

[참고] **길이 명시하지 않으면** 지정 위치 이후의 모든 문자열이 대체됨

(예) name="SUNG MINJI";
surname=SUBSTR(name, 1, 4);
givenname=SUBSTR(name, 6, 5);
⇒ surname=SUNG, givenname=MINJI



libname주의 (경로).