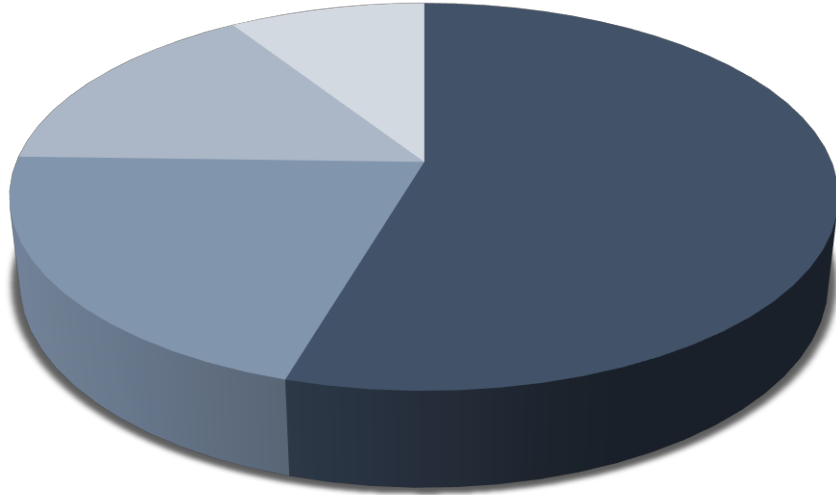




# 통계학 실습

## 2. 문법



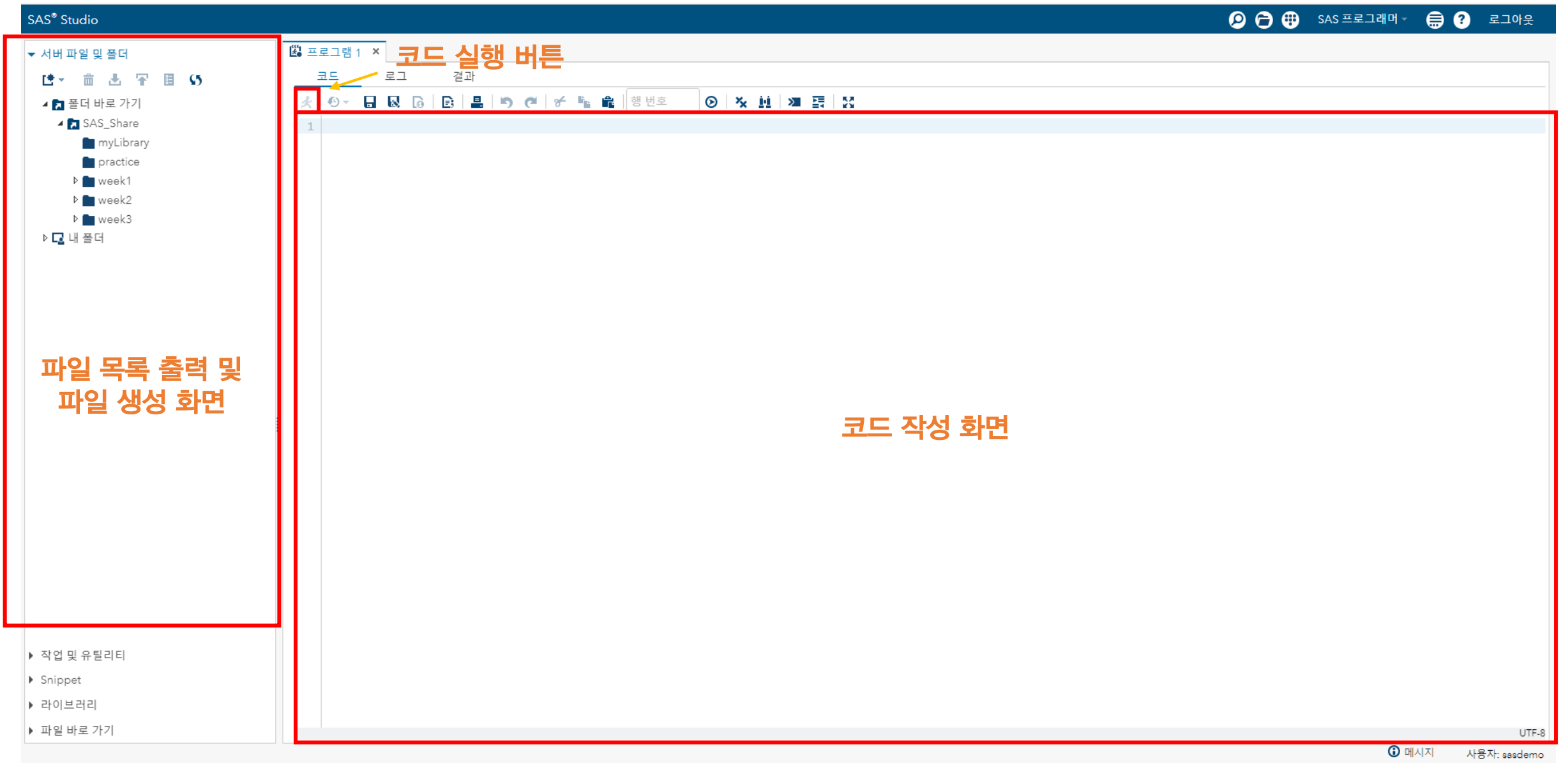
I. 기본 명령어

II. 데이터 다루기

III. 조건문, 연산, 함수

IV. 출력

# 프로그램 사용법



# 기본 명령어 – INPUT & LENGTH

- INPUT a b c d;
  - 변수명을 지정하는 일반적인 형태
- INPUT a1-a5;
  - 연속된 이름의 변수는 위와 같이 사용할 수 있다.
- INPUT a 1-2 b 3-5 c 10;
  - 원하는 index에 있는 자료를 읽도록 할 수 있다. ex) 변수 a에 1-2 index의 자료를 저장
- INPUT a 4. b 5. c 2. ;
  - 원하는 칸 만큼의 자료를 읽도록 할 수 있다. ex) 변수 a에 4칸 만큼의 자료를 저장
- INPUT str\$ num1 num2;
  - 변수가 문자인 경우 뒤에 꼭 \$를 넣어주어야 한다.
- LENGTH
  - DATA와 INPUT 사이에서 LENGTH를 이용하여 데이터 내용 길이 설정 가능
  - LENGTH 변수명 길이; 와 같이 사용한다.

## ■ 1. 일반적인 INPUT & LENGTH 사용

```
1 DATA class;  
2     LENGTH CLASSNAME$ 15;  
3     INPUT CLASSNAME $ NUMBER;  
4     CARDS;  
5 Science 37  
6 Math 45  
7 Programming 60  
8     ;  
9 RUN;  
0
```

## ■ 2. INPUT의 자유로운 사용

```
1 DATA class;  
2     INPUT A1 $ 3. A2 $ 2. A3 9-10;  
3     CARDS;  
4 Science 37  
5 Math 45  
6 Programming 60  
7     ;  
8 RUN;  
9
```

## 파일 이름 재정의

```
DATA 새로운SAS파일명;  
  SET 존재하는SAS파일명;  
RUN;
```

```
DATA 새로운SAS파일명;  
  SET 존재하는SAS파일명1 SAS파일명2 ~;  
RUN;
```

- SET 뒤에 파일명이 1개인 경우, 기존에 존재하는 SAS 파일을 새로운 SAS 파일로 재정의한다.

- 존재하는 SAS파일에서 필요한 변수명만 따로 추출하여 새로운 데이터로 만들고 싶을 때 사용할 수 있다.

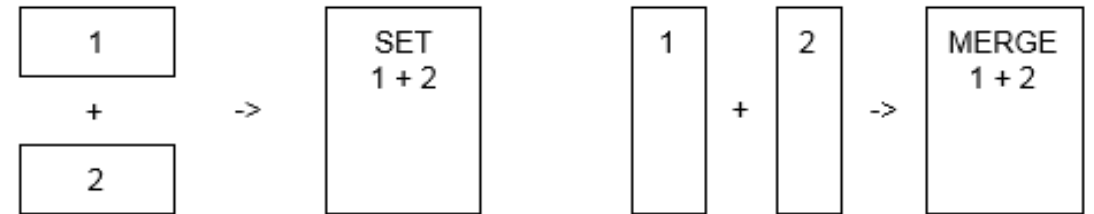
- SET 뒤에 파일명이 여러 개인 경우, 기존에 존재하던 여러 SAS 파일을 새로운 하나의 SAS 파일로 생성한다.

- 파일들에 포함된 변수명은 서로 일치해야 한다. 일치하지 않을 경우 missing value 가 생길 수 있다.

```
DATA 새로운SAS파일명;  
  MERGE 존재하는SAS파일명1 SAS파일명2  
  ~ ;  
RUN;
```

- MERGE : 두 가지 이상의 SAS 파일을 합쳐서 새로운 SAS 파일로 생성
  - SET으로 합치는 것과 다르게 합쳐지는 대상에 존재하는 모든 변수를 포함한다.

## SET vs. MERGE



# 데이터 골라보기 (KEEP & DROP)

```
DATA 새로운SAS파일명;  
  SET 존재하는SAS파일명;  
    KEEP 변수명1 변수명2 ~ ;  
RUN;
```

위아래 둘다 같은 결과를 냄

```
DATA 새로운SAS파일명;  
  (KEEP = 변수명1 변수명2 ~);  
  SET 존재하는SAS파일명;  
RUN;
```

- KEEP : SAS파일 중 특정 변수만 포함하는 SAS파일로 재정의



# 데이터 골라보기 (KEEP & DROP)

```
DATA 새로운SAS파일명;  
  SET 존재하는SAS파일명;  
    DROP 변수명1 변수명2 ~;  
RUN;
```

```
DATA 새로운SAS파일명;  
  (DROP = 변수명1 변수명2 ~;)  
  SET 존재하는SAS파일명;  
RUN;
```

- DROP : SAS파일 중 특정 변수만 제거 후 남는 것을 SAS 파일로 재정의

```
DATA SAS파일명;  
  ~~;  
  IF 조건1 THEN 문장1;  
  ELSE 문장2;  
RUN;
```

- IF ~ THEN ~ 은 여러 번 사용할 수 있다.
- 중간에 사용되는 조건문에는 ELSE IF (혹은 ELIF)를 이용한다.
  - ex) IF 조건문1 THEN 문장1;  
 ELSE IF 조건문2 THEN 문장2;  
 ELSE 문장3;

## 조건문 예시

```
DATA cse_math;  
  SET cse;  
  IF dept= "컴퓨터공학과" THEN math1 =  
    math + 5;  
  ELSE math1=math;  
RUN;
```

- 해당 조건이 만족되면, 새로운 column 인 'math1' 이 생성되게 된다.
- 전공이 컴퓨터공학과인 경우, 기존 수학 점수에 +5점 되어 math1에 저장
- 아닌 경우, 기존 수학 점수가 그대로 math1에 저장된다.

## 조건문 (continue)

```
DATA SAS파일명;
```

```
  ~;
```

```
  IF 조건문;
```

```
RUN;
```

```
DATA SAS파일명;
```

```
  ~;
```

```
  IF 조건문 THEN OUTPUT 새로운SAS파일명;
```

```
RUN;
```

- IF만 사용할 경우, 조건문에 부합하는 결과만 데이터로 수집된다.
- OUTPUT : 해당 조건문에 부합하는 결과의 데이터를 다른 파일로 따로 저장한다.

```
DATA score;  
  SET class2;  
    total = kor + eng + math + sci;  
    average = total / 2;  
RUN;
```

- 조건문을 사용하는 것과 같이 연산자들도 똑같이 사용 가능하다.
  - +, -, \*, /, \*\* (거듭제곱)
- 새로운 column들을 생성하면서, 연산 결과를 저장한다.

# 비교 연산자 및 논리 연산자

=	같다	&	AND
$\wedge$ 또는 $\sim$	같지 않다		OR
>	크다	$\wedge()$ 또는 $\sim()$	NOT
<	작다		
$\geq$	크거나 같다		
$\leq$	작거나 같다		

# 함수

COS(e)	코사인	MOD(e1, e2)	Modular 연산	PROBBETA(x,a,b)	베타분포
SIN(e)	사인	SIGN(e)	양수=1, 음수=-1	PROBGAM(x,a)	감마분포
TAN(e)	탄젠트	SQRT(e)	제곱근	PROBNORM(x)	표준정규분포
MEAN(e1, ..)	평균	CEIL(e)	같거나 큰 가장 작은 정수	PROBCHI(x,df)	카이제곱분포
SUM(e1, ..)	합	FLOOR(e)	같거나 작은 가장 큰 정수	PROBT(x,df)	t-분포
STD(e1, ..)	표준편차	INT(e)	e의 정수 부분	PROBF(x,df1,df2)	F-분포
VAR(e1, ..)	분산	ROUND(e, 단위)	단위만큼 반올림	RANNOR(seed)	평균=0, 분산=1인 표준정규분포 변량
CV(e1, ..)	변이계수	EXP(c)	지수의 c승	RANBIN(seed,n,p)	이항분포 변량
RANGE(e1, ..)	범위	GAMMA(c)	c의 감마함수	RANEXP(seed)	모수 $\lambda=1$ 인 지수분포 변량
STDERR(e1, ..)	표준오차	LOG(c)	c의 자연로그	RANGAM(seed,a)	감마분포 변량
ABS(e)	절댓값	LOG10(c)	c의 상용로그	RANPOI(seed, $\lambda$ )	평균= $\lambda$ 인 포아송분포 변량
MAX(e1, ..)	최댓값	PROBBNML(p,n,m)	이항분포	RANUNI(seed)	구간 (0,1)에서 균등분포 변량
MIN(e1, ..)	최솟값	POISSON( $\lambda$ ,m)	포아송분포		

```
PROC PRINT DATA=SAS파일명;  
RUN;
```

- 프로시저 : 주어진 데이터를 좀 더 편리하게 보여주거나 그래프로 나타내어 주는 등 추가적인 기능
- 작성한 프로그램의 마지막에 왼쪽과 같은 구문을 추가할 경우, 결과 화면으로 출력 결과를 확인할 수 있다.
- 출력 결과 화면을 HTML, PDF, RTF 파일로 내보낼 수 있다.



- 1. score1.txt와 score2.txt를 불러와 하나의 데이터로 합친다.
  - 위 txt들의 변수명은 gender / kor / eng / math / sci로 한다.
- 2. pass.txt는 score1.txt와 score2.txt를 합친 전체 인원의 합격 여부를 모아둔 데이터이다. pass.txt를 불러온다.
  - 위 txt의 변수명은 gender / pass로 한다.
  - pass.txt의 'pass' column에서 합격은 "T" 로, 불합격은 "F" 로 바꾸어 저장한다.
- 3. pass.txt의 데이터와 1.에서 합쳐진 데이터를 병합한다. 두 데이터는 순서에 맞게 정렬되어 있다.
- 4. 모든 과목의 점수를 더하는 새로운 변수 total을 추가해본다.

- 1. 실습에서 사용한 score1.txt와 score2.txt를 하나의 데이터로 합친다.
- 2. 국어 : 영어 : 수학 : 과학, 네 과목의 비중을 2:2:1:2로 하여 총 점을 나타내는 변수 total을 추가한다.
- 3. 새로운 column 'note' 를 추가하여 total이 400점 이상인 경우 "GREAT", 300점 이상인 경우 "GOOD", 200점 이상인 경우 "BAD", 그 이하는 "FAIL" 로 나타나도록 한다.
- 4. 변수 kor, eng, math, sci를 제외하고 데이터로 출력한다.
- 5. 출력 프로시저를 이용하여 결과 화면으로 출력한다.

- 1. 숙제를 결과 화면으로 출력하여 PDF로 내보낸다.
- 2. 코드 파일과 PDF를 압축하여 아래 서식의 이름으로 만들어 이메일로 제출한다. 이메일 제목은 파일 이름과 동일하게 한다.
- 서식
  - [2주차][학번][이름]통계학실습
- 제출 이메일
  - gtsk623@gmail.com