

# Computación Estadística

## EPG3308

Profesora: María Inés Godoy,  
Ayudante: María De Los Angeles Villena

Primer Semestre 2017: 28 marzo

# Do

Cuando se desea de realizar sentencia repetidamente se usa DO.

**DO** value=*start to stop*;

Sentencias;

**END**;

La instrucción **END** marca el final de las iteraciones.

## Ejemplo :

Crear una variable que indique el número de vocales que tiene el nombre

```
DATA Nombre;  
SET EPG3308;  
contar=0;  
DO i=1 to length(APELLIDO);  
S1=substr(APELLIDO,i,1);  
IF (S1="A" or S1="E" or S1="I" or S1="O" or S1="U" or  
S1="a" or S1="e" or S1="i" or S1="o" or S1="u" or  
S1="á" or S1="é" or S1="í" or S1="ó" or S1="ú" or  
S1="Á" or S1="É" or S1="Í" or S1="Ó" or S1="Ú" )  
THEN contar=contar+1;  
END;  
DROP i S1;  
PROC PRINT;  
RUN;
```

# Algunos Comandos

A continuación se presenta algunas funciones frecuentemente usadas

- ▶ *substr*(*ARGUMENTO*, *Posicioninicial*, *longitud*): Extrae desde el argumento los caracteres, tal que comienza desde la posición que se indica como inicial y con la longitud requerida. Por Ejemplo, *substr*(CAMILO,2,3)=AMI.
- ▶ *LENGTH*(*variable*) Es la longitud de la variable especificada.
- ▶ *SCAN*(*variable*, *n*, *dlim*) Extrae la n-ésima posición del string (*variable*), separando con la clave *dlim*

# Algunos Comandos

- ▶ *TRANWRD(variable, "elemento<sub>busca</sub>", "elemento-reemplazo")*  
Encuentra *elemento<sub>busca</sub>* y lo reemplaza por *elemento<sub>rreemplazo</sub>*
- ▶ *LOWCASE/UPCASE* reemplaza todas las mayúsculas por minúsculas / la forma inversa
- ▶ *FIND(variable, valor)* Encuentra el valor en la cadena y entrega la posición.
- ▶ Más funciones en SAS help

# Ejemplo :

## Probando algunas funciones

```
DATA NOMBRE_ALIGN;  
SET NOMBRE;  
NOM_ALIGN= RIGHT (NOMBRE) ;  
NOMBRE2=TRANWRD (NOMBRE, "A", "-" ) ;  
NOMBRE3=LOWCASE (NOMBRE) ;  
NOMBRE4=FIND (NOMBRE, "A" ) ;  
RUN;
```

## Otra forma de usar DO

Cuando se desea detener la iteraciones con una determinada condición. **WHILE** y **UNTIL**.

- ▶ **WHILE** se puede utilizar para realizar las iteraciones mientras ocurre una condición determinada. La forma de usarlo es la siguiente,

**DO WHILE** (expresión);

Sentencias;

**END;**

# Otra forma de usar DO

- ▶ **UNTIL** se puede utilizar para realizar las iteraciones mientras **no** ocurra una condición determinada, la forma de usarlo es,

**DO UNTIL** (expresión);

Sentencias;

**END;**

```
Data Ejemplol;  
n=0;  
DO WHILE (n<= 5);  
output;  
n+1;  
END;  
RUN;  
proc print; run;
```



# Ejemplos

```
data ejemplo2;  
set ejemplo1;  
do while(n<3);  
y=n*2;  
n+1;  
output;  
end;  
run;  
proc print; run;
```

# Ejemplos

```
data Ejemplo3;  
input x1 x2;  
do j=1 to 10 while(j<x1);  
y1=x1**j;  
y2=(x1+x2)**j;  
output;  
end;  
cards;  
6 1  
11 1  
run;  
proc print;  
run;
```

# Ejemplo

```
data Ejemplo4;  
K = 1;  
do until (K > 1);  
k2=K*2;  
output;  
K+1;  
end;  
run;  
proc print data=ejemplo4;  
run;
```

# Do Until

La expresión es evaluada al final de cada iteración, por lo que siempre al menos una vez entra al ciclo. Si la expresión es verdadera no itera nuevamente.

```
data ejemplo5;  
input x;  
z=0;  
do i = 1 to 5 until(z > 5);  
z = z+x*i;  
output;  
end;  
cards;  
1  
2  
3  
4  
;
```

## Ejemplo

```
data ejemplo6;  
DO x = 1, 2, 3, 4, 5, 6, 7, 8;  
y = x**2;  
output;  
end;  
run;  
  
proc print data=ejemplo6 noobs;  
run;
```

# Ejemplo

```
data ejemplo7;  
    DO x = 1, 2, 3, 4, 5, 6, 7, 8;  
        y = x**2;  
        output;  
    end;  
run;  
  
proc print data=ejemplo7 noobs;  
run;
```

# PROCEDIMIENTO MEANS

**PROC MEANS** <option(s)> <statistic-keyword(s)>;

**BY**<DESCENDING> variable-1 <... <DESCENDING>

variable-n><NOTSORTED>;

**CLASS** variable(s) </ option(s)>;

**FREQ** variable;

**ID** variable(s);

**OUTPUT** <OUT=SAS-data-set> <output-statistic-specification(s)>

<id-group-specification(s)> <maximum-id-specification(s)>

<minimum-id-specification(s)> </ option(s)> ;

**TYPES** request(s);

**VAR** variable(s) < / WEIGHT=weight-variable>;

**WAYS** list;

**WEIGHT** variable;

# PROCEDIMIENTO MEANS

- ▶ **PROC MEANS** Calcular los estadísticos descriptivos de las variables.
- ▶ **BY** Calcula estadísticas separadas para cada uno de los grupo.
- ▶ **CLASS** Identifica las variables cuyos valores definen subgrupos para el análisis.
- ▶ **FREQ** Identifica una variable cuyos valores representan la frecuencia de cada observación.
- ▶ **ID** Incluye variables de identificación adicionales en el conjunto de datos de salida.
- ▶ **OUTPUT** Crea un conjunto de datos de salida que contiene estadísticas y variables de identificación.



# PROCEDIMIENTO MEANS

- ▶ **TYPES** Identifica las combinaciones específicas de las variables **CLASS** que se usa para subdividir los datos.
- ▶ **VAR** Identifica las variables de análisis y su orden en los resultados.
- ▶ **WAYS** Especifica maneras de hacer combinaciones únicas de las variables **CLASS**.
- ▶ **WEIGHT** Identifica una variable cuyos valores son los peso de cada observación en los cálculos estadísticos.

# PROCEDIMIENTO MEANS

```
Title 'Ejemplo 1';  
data cake;  
input LastName $ 1-12 Age 13-14 PresentScore 16-17  
TasteScore 19-20 Flavor $ 23-32 Layers 34 ;  
datalines;  
Orlando      27 93 80  Vanilla      1  
Ramey        32 84 72  Rum          2  
Goldston     46 68 75  Vanilla      1  
Roe          38 79 73  Vanilla      2  
.  
.  
;  
proc means data=cake n mean max min range std;  
run;
```

# PROCEDIMIENTO MEANS

```
Title 'Ejemplo 2';  
Data grade;  
input Name $ 1-8 Gender $ 11 Status $13 Year $ 15-16  
Section $ 18 Score 20-21 FinalGrade 23-24;  
datalines;  
Abbott      F 2 97 A 90 87  
Branford    M 1 98 A 92 97  
Crandell    M 2 98 B 81 71  
.  
.  
;  
proc means data=grade maxdec=3;  
var Score;  
class Status Year;  
types () status*year;  
run;
```

# PROCEDIMIENTO MEANS

```
title 'Ejemplo 3';  
proc sort data=Grade out=GradeBySection;  
by section;  
run;  
proc means data=GradeBySection min max median;  
by Section;  
var Score;  
class Status Year;  
run;  
  
proc means data=Grade noprint;  
class Status Year;  
var FinalGrade;  
output out=sumstat mean=AverageGrade  
idgroup (max(score) obs out (name)=BestScore) / ways levels;  
run;
```

# Ejercicios

**Ejercicios:** Con lo aprendido en la clase de hoy, realice lo siguiente.

- ▶ Cree una base de datos con las siguientes características.
  - ▶ X1= Una variable numérica que toma valores desde el 1 hasta el 10.
  - ▶ X2= Una variable numérica que toma 10 números pares correlativos, comenzando desde el 2.
  - ▶ X3= Una variable numérica que toma 10 números impares correlativos, comenzando desde el 1.
- ▶ Usando las sentencias **DO WHILE** o **DO UNTIL**, escriba las tablas numéricas del 2,3,4,5 en una misma data.

# Referencia

SAS Statistics Institute Inc. 2012. BASE SAS 9.3 Prodedures Guide.  
Cary, NC: Sas Institute Inc.