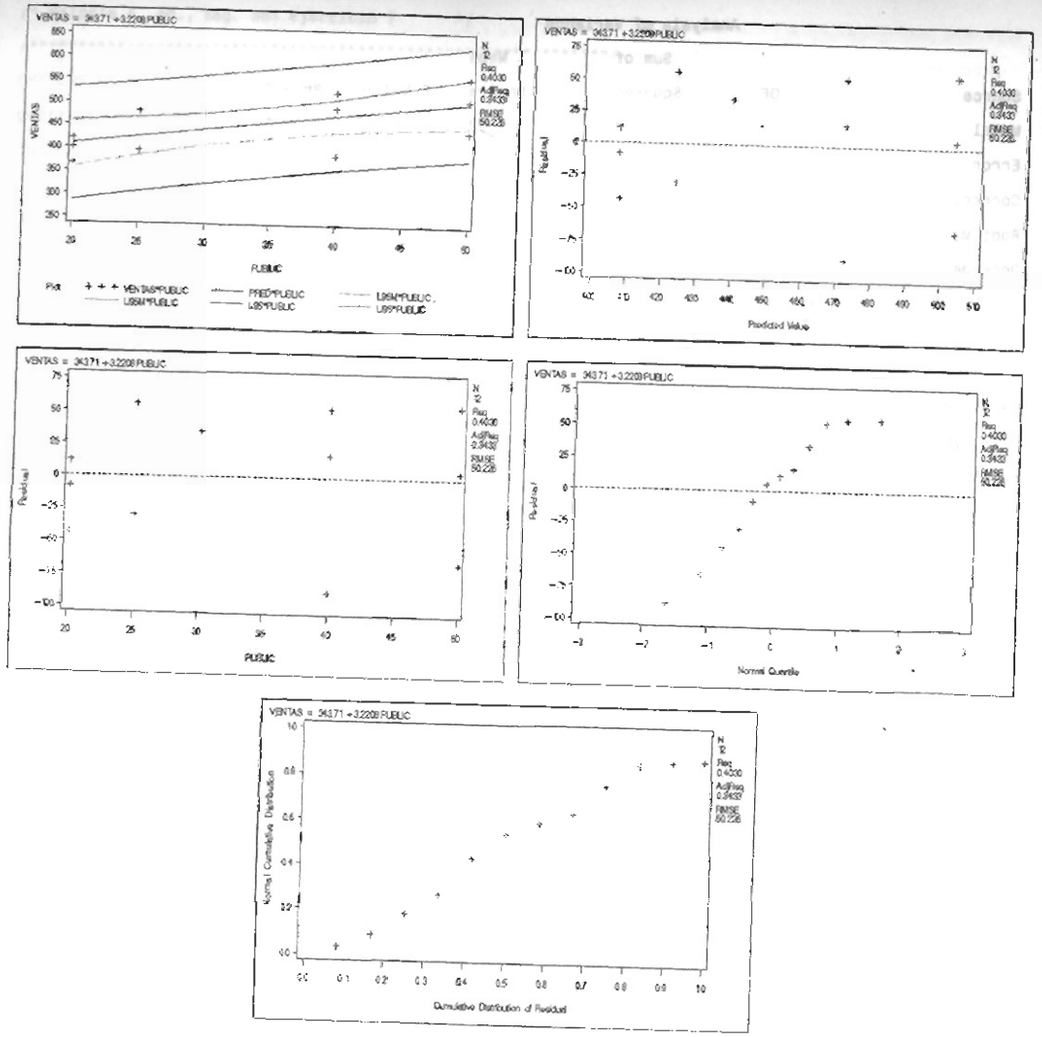


En el listado de la salida del SAS distinguimos lo siguiente:

Uncorrected Sums of squares and Crossproducts: Son la suma de cuadrados sin corregir y los productos cruzados.

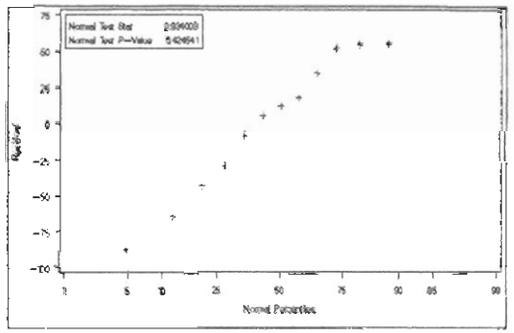
Analysis of Variance: Es el análisis de varianza correspondiente al problema planteado.

Parameter Estimates: Estimación de parámetros. En este punto se encuentra ßo y ß1 del modelo, además de otros resultados importantes para prueba de hipótesis con el valor P.

Dep Var Predict: Esta parte muestra los valores ajustados, los residuales y los valores del modelo. Note que el valor de PUBLIC al cual no le fue adjudicado valor de VENTAS muestra el valor ajustado según el modelo. Es decir, sirve para responder a la pregunta c) del ejercicio.

Por último se muestran las gráficas con los resultados pedidos en el PROC REG. Observe las gráficas de cuantil-cuantil y de probabilidad normal obtenidas en el proc reg. Otra opción es guardar los residuales y analizarlos con el proc univariate, en el cual podemos obtener tests de normalidad:

```
PROC UNIVARIATE DATA=RESID NOPRINT NORMALTEST;
VAR RES;
PROBPLOT RES;
INSET NORMALTEST PROBN;
RUN;
QUIT;
```



Ejenplo 2: Regresión múltiple en sas

```
/*Ejemplo 10.3 walpole 4ta ed.*/
options nodate, nocenter nonumber;
data uno;
input y x1 x2 x3;
cards;
25.5 1.74 5.30 10.80
31.2 6.32 5.42 9.40
25.9 6.22 8.41 7.20
38.4 10.52 4.63 8.50
18.4 1.19 11.60 9.40
26.7 1.22 5.85 9.90
26.4 4.10 6.62 8.00
25.9 6.32 8.72 9.10
32.0 4.08 4.42 8.70
25.2 4.15 7.60 9.20
39.7 10.15 4.83 9.40
35.7 1.72 3.12 7.60
26.5 1.70 5.30 8.20
;
run;
proc reg data=uno;
model y= ; /*modelo con sólo intercepto sin variables explicatorias*/
model y=x1/ss1 ss2;
```

```
model y=x1 x2/ss1 ss2;
model y=x1 x2 x3/ss1 ss2;
b1:test x1=0;/*prueba F parcial sobre la significancia de β1*/
b2:test x2=0; ;/*prueba F parcial sobre la significancia de β2*/
b3:test x3=0; ;/*prueba F parcial sobre la significancia de β3*/
b1b2:test x1,x2; ;/*prueba F parcial sobre la significancia de β1 y β2*/
b2b3:test x2,x3; ;/*prueba F parcial sobre la significancia de β2 y β3*/
b1b3:test x1,x3; ;/*prueba F parcial sobre la significancia de β1 y β3*/
run;
quit;
```

The REG Procedure
Model: MODEL1
Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	0	0	.	.	.
Error	12	438.13077	36.51090		
Corrected Total	12	438.13077			

Root MSE	6.04242	R-Square	0.0000
Dependent Mean	29.03846	Adj R-Sq	0.0000
Coeff Var	20.80835		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	29.03846	1.67587	17.33	<.0001

Model: MODEL2
Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	187.31179	187.31179	8.21	0.0153
Error	11	250.81898	22.80173		
Corrected Total	12	438.13077			

Root MSE	4.77512	R-Square	0.4275
Dependent Mean	29.03846	Adj R-Sq	0.3755
Coeff Var	16.44411		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Type I SS	Type II SS
Intercept	1	23.39788	2.37213	9.86	<.0001	10962	2218.42121
x1	1	1.23385	0.43049	2.87	0.0153	187.31179	187.31179

Model: MODEL3
Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	398.12450	199.06225	49.76	<.0001
Error	10	40.00627	4.00063		
Corrected Total	12	438.13077			

Root MSE	2.00016	R-Square	0.9087
Dependent Mean	29.03846	Adj R-Sq	0.8904
Coeff Var	6.88796		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Type I SS	Type II SS
Intercept	1	36.09468	2.01160	17.94	<.0001	10962	1288.03902
x1	1	1.03051	0.18246	5.65	0.0002	187.31179	127.58335
x2	1	-1.86964	0.25756	-7.26	<.0001	210.81271	210.81271

Model: MODEL4
Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	399.45437	133.15146	30.98	<.0001
Error	9	38.67640	4.29738		
Corrected Total	12	438.13077			

Root MSE	2.07301	R-Square	0.9117
Dependent Mean	29.03846	Adj R-Sq	0.8823
Coeff Var	7.13885		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Type I SS	Type II SS
Intercept	1	39.15735	5.88706	6.65	<.0001	10962	190.12239
x1	1	1.01610	0.19090	5.32	0.0005	187.31179	121.75501
x2	1	-1.86165	0.26733	-6.96	<.0001	210.81271	208.40997
x3	1	-0.34326	0.61705	-0.56	0.5916	1.32987	1.32987

The REG Procedure
Model: MODEL4

Test b1 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	1	121.75501	28.33	0.0005
Denominator	9	4.29738		

The REG Procedure
Model: MODEL4

Test b2 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	1	208.40997	48.50	<.0001
Denominator	9	4.29738		

The REG Procedure
Model: MODEL4

Test b3 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	1	1.32987	0.31	0.5916
Denominator	9	4.29738		

The REG Procedure
Model: MODEL4

Test b1b2 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	2	192.12597	44.71	<.0001
Denominator	9	4.29738		

The REG Procedure
Model: MODEL4

Test b2b3 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	2	106.07129	24.68	0.0002
Denominator	9	4.29738		

The REG Procedure
Model: MODEL4

Test b1b3 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	2	64.45661	15.00	0.0014
Denominator	9	4.29738		

2.5.4.2 Prueba de carencia de ajuste:

Para obtener dicha prueba es necesario trabajar con el **PROC RSREG** (procedimiento de superficie de respuesta), con el cual además se puede hacer regresión polinomial. Veremos brevemente con un ejemplo cómo se obtienen el análisis de varianza para una prueba de carencia de ajuste:

```
OPTIONS NODATE NOCENTER NONUMBER;
goptions reset=global gunit=pct border cback=white
colors=(black blue green red)
ftitle=swissb ftext=swiss htitle=6 htext=3;
data uno;
input x y;
cards;
110 235
110 198
110 173
230 174
230 149
230 124
360 115
360 130
360 102
360 95
505 122
505 112
505 98
505 96
;
run;
proc rsreg data=uno;
model y=x/covar=1 lackfit; /*covar=1 indica que la regresión en X es lineal (polinomio de grado 1)*/
run;
quit;
```

```
The RSREG Procedure
    Response Surface for Variable y
Response Mean                  137.357143
Root MSE                        24.564563
R-Square                         0.6967
Coefficient of Variation         17.8837
```

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Covariates	1	16634	0.6967	27.57	0.0002
Linear	0	0	0.0000	.	.
Quadratic	0	0	0.0000	.	.
Crossproduct	0	0	0.0000	.	.
Total Model	1	16634	0.6967	27.57	0.0002

Residual	DF	Sum of Squares	Mean Square	F Value	Pr > F
Lack of Fit	2	2880.013219	1440.006609	3.30	0.0792
Pure Error	10	4361.000000	436.100000		
Total Error	12	7241.013219	603.417768		

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|	Parameter Estimate from Coded Data
Intercept	1	212.720931	15.784060	13.48	<.0001	212.720931
x	1	-0.235512	0.044856	-5.25	0.0002	-0.235512

2.5.4.3 Regresión polinomial en SAS:

Para realizar regresión cuadrática de una variable dependiente en una o varias variables explicatorias, se puede trabajar con el **PROC RSREG.** Para ilustrar cómo, se presenta a continuación un ejemplo donde se ajusta el polinomio de grado 2: $y = \beta_0 + \beta_1 x + \beta_2 x^2$:

```
OPTIONS NODATE NOCENTER NONUMBER;
goptions reset=global gunit=pct border cback=white
colors=(black blue green red)
ftitle=swissb ftext=swiss htitle=6 htext=3;

/*Ejercicio 10-47 Probabilidad y Estadistica, Montgomery D.C y Runger, G. C.*/
data uno;
input y x;
cards;
-1.42 0.25
-1.39 0.50
-1.55 0.75
-1.89 1.00
-2.43 1.25
-3.15 1.50
-4.05 1.75
-5.15 2.00
-6.43 2.25
-7.89 2.50
;
run;
proc rsreg;
model y=x/nooptimal;
run;
quit;
```

```
The RSREG Procedure
Coding Coefficients for the Independent Variables
Factor     Subtracted off      Divided by
x               1.375000        1.125000
    Response Surface for Variable y
Response Mean                  -3.535000
Root MSE                        0.003568
R-Square                          1.0000
Coefficient of Variation         -0.1009
```

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Linear	1	42.703225	0.9026	3355253	<.0001
Quadratic	1	4.606936	0.0974	361974	<.0001
Crossproduct	0	0	0.0000	.	.
Total Model	2	47.310161	1.0000	1858613	<.0001

Residual	DF	Sum of Squares	Mean Square
Total Error	7	0.000089091	0.000012727

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|	Parameter Estimate from Coded Data
Intercept	1	-1.633000	0.004196	-389.18	<.0001	-2.764375
x	1	1.232182	0.007010	175.78	<.0001	-3.237545
x*x	1	-1.494545	0.002484	-601.64	<.0001	-1.891534

Factor	DF	Sum of Squares	Mean Square	F Value	Pr > F
x	2	47.310161	23.655080	1858613	<.0001

En la declaración modelo pueden especificarse opciones como las siguientes:

ACTUAL, PREDICT, RESIDUAL, para presentar respectivamente, los valores originales de la respuesta, los ajustados y los residuales del ajuste.

COVAR=n declara que las primeras n variables en el lado derecho del modelo son regresores lineales simples y no factores en la superficie de respuesta cuadrática. Por defecto PROC RSREG forma para todas las variables regresoras en la declaración MODEL, los efectos cuadráticos y de productos cruzados.

D para presentar las distancias de Cook.

LACKFIT para prueba de carencia de ajuste.

L95, L95M, para desplegar respectivamente, los límites inferiores de predicción y de confianza de la respuesta media.

U95, U95M para desplegar respectivamente, los límites superiores de predicción y de confianza de la respuesta media.

NOOPTIMAL para suprimir la salida del análisis canónico de la superficie de respuesta cuadrática.

PRESS para calcular y presentar la suma de los cuadrados de los residuales de predicción.

2.5.4.4 Diagnósticos en regresión.

Con el siguiente ejemplo, se presenta algunas opciones para obtener varios estadísticos para diagnósticos en regresión. No se presentan las salidas y se sugiere al lector realizar este ejemplo como ejercicio:

```
/*Ejercicio 10-8 Probabilidad y Estadistica, Montgomery D.C y Runger, G. C.*/
options nodate nocenter nonumber;
data uno;
input y x1 x2 x3 x4 x5 x6;
```



```
cards;
8.0 5.2 19.6 29.6 94.9 2.1 2.3
8.3 5.2 19.8 32.4 89.7 2.1 1.8
8.5 5.6 19.6 31.0 96.2 2.0 2.0
8.8 6.4 19.4 32.4 95.6 2.2 2.1
9.0 5.8 18.6 28.6 86.5 2.0 1.8
9.3 5.2 18.8 30.6 84.5 2.1 2.1
9.3 5.6 20.4 32.4 88.8 2.2 1.9
9.5 6.0 19.0 32.6 85.7 2.1 1.9
9.8 5.2 20.8 32.2 93.6 2.3 2.1
10.0 5.8 19.9 31.8 86.0 2.1 1.8
10.3 6.4 18.0 32.6 87.1 2.0 1.6
10.5 6.0 20.6 33.4 93.1 2.1 2.1
10.8 6.2 20.2 31.8 83.4 2.2 2.1
11.0 6.2 20.2 32.4 94.5 2.1 1.9
11.3 6.2 19.2 31.4 83.4 1.9 1.8
11.5 5.6 17.0 33.2 85.2 2.1 2.1
11.8 6.0 19.8 35.4 84.1 2.0 1.8
12.3 5.8 18.8 34.0 86.9 2.1 1.8
12.5 5.6 18.6 34.2 83.0 1.9 2.0
. . 20 30 90 2.0 .
;
run;
data dos;
set uno;
keep y x3 x4;
run;
/*Obtencion de medidas de diagnostico de la regresion*/
proc reg data=dos noprint;
model y=x3 x4;
output out=estad p=pred r=res student=res_estudent h=hii cookd=cooks press=res_press;
run;
quit;
proc print data=estad;run;
/*Estandarizacion de residuales*/
proc standard data=estad out=estand m=0 s=1;
var res;
run;
data estand2;
set estand;
```

```
rename res=res_estand;
run;
proc print data=estand2;run;
/*Graficos de residuos comunes**/
proc plot data=estad hpercent=45 vpercent=50;
plot res*pred/vref=0;
plot res*x3/vref=0;
plot res*x4/vref 0;
run;
/*Graficos de residuos estandarizados*/
proc plot data=estand2 hpercent=45 vpercent=50;
plot res_estand*pred/vref=0;
plot res_estand*x3/vref=0;
plot res_estand*x4/vref=0;
run;
/*Graficos de residuos studentizados*/
proc plot data=estand2 hpercent=45 vpercent=50;
plot res_estudent*pred/vref=0;
plot res_estudent*x3/vref=0;
plot res_estudent*x4/vref=0;
run;
quit;
/*Analisis de normalidad residuos estandarizados*/
proc univariate data=estand2 normal;
var res_estand;
qqplot res_estand;
run;
quit;
```

2.5.4.5 Estandarización y diagnóstico de colinealidad.

Con el siguiente programa, se muestran los procedimientos SAS necesarios para estandarizar variables y realizar diagnósticos de multicolinealidad entre predictores:

```
options nodate nocenter nonumber;
data uno;
input x1 x2 x3 y;
cards;
19.5 43.1 29.1 11.9
24.7 49.8 28.2 22.8
30.7 51.9 37.0 18.7
```



```
29.8 54.3 31.1 20.1
19.1 42.2 30.9 12.9
25.6 53.9 23.7 21.7
31.4 58.5 27.6 27.1
27.9 52.1 30.6 25.4
22.1 49.9 23.2 21.3
25.5 53.5 24.8 19.3
31.1 56.6 30.0 25.4
30.4 56.7 28.3 27.2
18.7 46.5 23.0 11.7
19.7 44.2 28.6 17.8
14.6 42.7 21.3 12.8
29.5 54.4 30.1 23.9
27.7 55.3 25.7 22.6
30.2 58.6 24.6 25.4
22.7 48.2 27.1 14.8
25.2 51.0 27.5 21.1
;
run;
/*Regresion con datos originales*/
proc reg data=uno;
model y=x1-x3;
run;
quit;
/*Calculo de matriz de correlacion para un diagnóstico inicial de colinealidad por pares de
predictores*/
proc corr data=uno nosimple noprob;run;
/*Escalamiento de longitud unitaria. Requiere el proc means, y los datas cent, cent2 y dos que se crean
a continuación*/
proc means data=uno noprint;
output out=cent mean=m1 m2 m3 my css=scc1 scc2 scc3 sccy;/* CSS es la suma de cuadrados corregidos*/
run;
/*creamos un data con las columnas correspondientes a m1 m2 m3 my scc1 scc2 scc3 sccy, con estos
estadísticos repetidos 20 veces, o sea el tamaño de observaciones en data uno*/
data cent2;
set cent;
do i=1 to 20;
output;
end;
keep m1 m2 m3 my scc1 scc2 scc3 sccy;
```

```
run;
data dos;
merge uno cent2;
x1e=(x1-m1)/sqrt(scc1);
x2e=(x2-m2)/sqrt(scc2);
x3e=(x3-m3)/sqrt(scc3);
ye=(y-my)/sqrt(sccy);
run;
proc print data=dos;run;
/*Regresión con variables escalonadas por método de longitud unitaria y obtención de VIFs*/
/*en las salidas observe que la diagonal de la matrix (X'X)-1 relacionada con las variables*/
/*X1 a X3 son iguales a  los VIFs de dichas variables, leidas en la tabla de parámetros estimados*/
proc reg data=dos;
model ye=x1e x2e x3e/i vif;
run;
quit;
/*otro método es hacer escalamiento normal unitario, es más breve que el anterior caso*/
/*Escalamiento normal unitario. Normalizamos primero las variables en data UNO*/
proc standard data=uno out=estand m=0 s=1;
run;
/*Regresión con variables estandarizadas*/
proc reg data=estand;
model y=x1-x3/vif;
run;
quit;
```

2.5.4.6 Selección de modelos: Ajuste de todos los modelos posibles.

El siguiente ejemplo ilustra un programa SAS para generar salidas con las cuales se evalúan los posibles modelos con un conjunto de variables predictoras dadas:

```
/*Ejemplo 10.3 walpole 4ta ed.*/
options nodate nocenter nonumber;
data uno;
input y x1 x2 x3;
cards;
25.5 1.74 5.30 10.80
31.2 6.32 5.42 9.40
25.9 6.22 8.41 7.20
38.4 10.52 4.63 8.50
18.4 1.19 11.60 9.40
```



```
26.7 1.22 5.85 9.90
26.4 4.10 6.62 8.00
25.9 6.32 8.72 9.10
32.0 4.08 4.42 8.70
25.2 4.15 7.60 9.20
39.7 10.15 4.83 9.40
35.7 1.72 3.12 7.60
26.5 1.70 5.30 8.20
;
proc reg data=uno;
model y=x1 x2 x3/b selection=rsquare mse adjrsq cp;
run;
quit;
The REG Procedure
Model: MODEL1
Dependent Variable: y
R-Square Selection Method
Number in                Adjusted
  Model      R-Square    R-Square        C(p)           MSE
      1        0.6175      0.5827     29.9981       15.23542
      1        0.4275      0.3755     49.3656       22.80173
      1        0.0347      -.0531     89.4154       38.44803
-----------------------------------------------------------
      2        0.9087      0.8904      2.3095        4.00063
      2        0.6338      0.5606     30.3324       16.04314
      2        0.4360      0.3233     50.4970       24.70864
-----------------------------------------------------------
      3        0.9117      0.8823      4.0000        4.29738

Number in                ------------------Parameter Estimates-----------------
  Model      R-Square     Intercept        x1            x2            x3
      1        0.6175      42.21088         .      -2.09290             .
      1        0.4275      23.39788   1.23385             .             .
      1        0.0347      39.21692         .             .      -1.14662
-----------------------------------------------------------------------------
      2        0.9087      36.09468   1.03051      -1.86964             .
      2        0.6338      49.05408         .      -2.06735      -0.78901
      2        0.4360      28.61229   1.20828             .      -0.57425
-----------------------------------------------------------------------------
      3        0.9117      39.15735   1.01610      -1.86165      -0.34326
```

2.5.4.7 Selección secuencial de variables: fordward, backward y stepwise.

Veamos un ejemplo SAS donde en un mismo proc reg se invocan los tres métodos de selección secuencial:

```
/*Ejercicio 10-8, Probabilidad y Estadística, Montgomery D.C y Runger, G. C.*/
options nodate nocenter nonumber;
data uno;
input y x1 x2 x3 x4 x5 x6;
cards;
8.0 5.2 19.6 29.6 94.9 2.1 2.3
8.3 5.2 19.8 32.4 89.7 2.1 1.8
8.5 5.8 19.6 31.0 96.2 2.0 2.0
8.8 6.4 19.4 32.4 95.6 2.2 2.1
9.0 5.8 18.6 28.6 86.5 2.0 1.8
9.3 5.2 18.8 30.6 84.5 2.1 2.1
9.3 5.6 20.4 32.4 88.8 2.2 1.9
9.5 6.0 19.0 32.6 85.7 2.1 1.9
9.8 5.2 20.8 32.2 93.6 2.3 2.1
10.0 5.8 19.9 31.8 86.0 2.1 1.8
10.3 6.4 18.0 32.6 87.1 2.0 1.6
10.5 6.0 20.6 33.4 93.1 2.1 2.1
10.8 6.2 20.2 31.8 83.4 2.2 2.1
11.0 6.2 20.2 32.4 94.5 2.1 1.9
11.3 8.2 19.2 31.4 83.4 1.9 1.8
11.5 5.6 17.0 33.2 85.2 2.1 2.1
11.8 6.0 19.8 35.4 84.1 2.0 1.8
12.3 5.8 18.8 34.0 86.9 2.1 1.8
12.5 5.6 18.6 34.2 83.0 1.9 2.0
. . 20 30 90 2.0 .
;
run;
proc reg data=uno outest=estad;
model y=x1 x2 x3 x4 x5 x6/selection=f sle=0.05; /*método forward*/
model y=x1 x2 x3 x4 x5 x6/selection=b sls=0.05; /*método backward*/
model y=x1 x2 x3 x4 x5 x6/selection=stepwise sle=0.05 sls=0.05; /*método stepwise*/
run;
quit;
```



```
The REG Procedure
Model: MODEL1
Dependent Variable: y
Forward Selection: Step 1
Variable x3 Entered: R-Square = 0.4772 and C(p) = 6.7253
```

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	15.85455	15.85455	15.52	0.0011
Error	17	17.36650	1.02156		
Corrected Total	18	33.22105			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-8.97104	4.85449	3.48869	3.42	0.0821
x3	0.59306	0.15054	15.85455	15.52	0.0011

Bounds on condition number: 1, 1

Forward Selection: Step 2

Variable x4 Entered: R-Square = 0.6476 and C(p) = 1.6463

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	21.51330	10.75665	14.70	0.0002
Error	16	11.70775	0.73173		
Corrected Total	18	33.22105			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	4.65631	6.39482	0.38795	0.53	0.4771
x3	0.51133	0.13075	11.19035	15.29	0.0012
x4	-0.12418	0.04466	5.65875	7.73	0.0134

Bounds on condition number: 1.0532, 4.2128

No other variable met the 0.0500 significance level for entry into the model.

Summary of Forward Selection

Step	Variable Entered	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	x3	1	0.4772	0.4772	6.7253	15.52	0.0011
2	x4	2	0.1703	0.6476	1.6463	7.73	0.0134

Model: MODEL2

Dependent Variable: y

Backward Elimination: Step 0

All Variables Entered: R-Square = 0.7113 and C(p) = 7.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	23.62863	3.93811	4.93	0.0092
Error	12	9.59242	0.79937		
Corrected Total	18	33.22105			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	3.13682	8.10979	0.11959	0.15	0.7057
x1	0.64443	0.58892	0.95717	1.20	0.2953
x2	-0.01042	0.26765	0.00121	0.00	0.9696
x3	0.50465	0.14234	10.04833	12.57	0.0040
x4	-0.11967	0.05623	3.61968	4.53	0.0548
x5	-2.46177	2.59776	0.71787	0.90	0.3620
x6	1.50441	1.51936	0.78371	0.98	0.3416

Bounds on condition number: 1.564, 50.783

Backward Elimination: Step 1

Variable x2 Removed: R-Square = 0.7112 and C(p) = 5.0015

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	23.62742	4.72548	6.40	0.0033
Error	13	9.59363	0.73797		
Corrected Total	18	33.22105			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	3.08042	7.66677	0.11913	0.16	0.6944
x1	0.64474	0.56580	0.95827	1.30	0.2750
x3	0.50414	0.13619	10.11226	13.70	0.0027
x4	-0.12041	0.05079	4.14790	5.62	0.0339
x5	-2.49765	2.33368	0.84532	1.15	0.3040
x6	1.50945	1.45456	0.79472	1.08	0.3183

Bounds on condition number: 1.5119, 33.062

Backward Elimination: Step 2

Variable x6 Removed: R-Square = 0.6873 and C(p) = 3.9957

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	22.83271	5.70818	7.69	0.0017
Error	14	10.38835	0.74202		
Corrected Total	18	33.22105			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	4.96439	7.46915	0.32780	0.44	0.5171
x1	0.45370	0.53648	0.53071	0.72	0.4119
x3	0.49090	0.13596	9.67275	13.04	0.0028
x4	-0.10707	0.04927	3.50405	4.72	0.0474
x5	-1.82178	2.24710	0.48772	0.66	0.4311

Bounds on condition number: 1.2643, 19.06

Backward Elimination: Step 3

Variable x5 Removed: R-Square = 0.6726 and C(p) = 2.6058

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	22.34499	7.44833	10.27	0.0006
Error	15	10.87606	0.72507		
Corrected Total	18	33.22105			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	2.41933	6.69954	0.09455	0.13	0.7230
x1	0.55297	0.51632	0.83169	1.15	0.3011
x3	0.47898	0.13361	9.31780	12.85	0.0027
x4	-0.12338	0.04446	5.58374	7.70	0.0142

Bounds on condition number: 1.1099, 9.6669

```
Backward Elimination: Step 4

Variable x1 Removed: R-Square = 0.6476 and C(p) = 1.6463

                          Analysis of Variance

                                  Sum of          Mean
Source                   DF      Squares        Square    F Value    Pr > F
Model                     2     21.51330      10.75665      14.70    0.0002
Error                    16     11.70775       0.73173
Corrected Total          18     33.22105

              Parameter     Standard
Variable       Estimate        Error   Type II SS  F Value  Pr > F
Intercept      4.65631      6.39482      0.38795     0.53  0.4771
x3             0.51133      0.13075     11.19035    15.29  0.0012
x4            -0.12418      0.04466      5.65875     7.73  0.0134

Bounds on condition number: 1.0532, 4.2128
----------------------------------------------------------------------------------------

All variables left in the model are significant at the 0.0500 level.

                       Summary of Backward Elimination

        Variable     Number     Partial     Model
Step    Removed      Vars In    R-Square    R-Square     C(p)      F Value    Pr > F
  1     x2               5      0.0000      0.7112      5.0015       0.00     0.9696
  2     x6               4      0.0239      0.6873      3.9957       1.08     0.3183
  3     x5               3      0.0147      0.6726      2.6058       0.66     0.4311
  4     x1               2      0.0250      0.6476      1.6463       1.15     0.3011

The REG Procedure

Model: MODEL3

Dependent Variable: y

Stepwise Selection: Step 1

Variable x3 Entered: R-Square = 0.4772 and C(p) = 6.7253

                          Analysis of Variance

                                  Sum of          Mean
Source                   DF      Squares        Square    F Value    Pr > F
Model                     1     15.85455      15.85455      15.52    0.0011
Error                    17     17.36650       1.02156
Corrected Total          18     33.22105

              Parameter     Standard
Variable       Estimate        Error   Type II SS  F Value  Pr > F
Intercept     -8.97104      4.85449      3.48869     3.42  0.0821
x3             0.59306      0.15054     15.85455    15.52  0.0011

Bounds on condition number: 1, 1
----------------------------------------------------------------------------------------
```



```
Stepwise Selection: Step 2

Variable x4 Entered: R-Square = 0.6476 and C(p) = 1.6463

                          Analysis of Variance

                                  Sum of          Mean
Source                   DF      Squares        Square    F Value    Pr > F
Model                     2     21.51330      10.75665      14.70    0.0002
Error                    16     11.70775       0.73173
Corrected Total          18     33.22105

              Parameter     Standard
Variable       Estimate        Error   Type II SS  F Value  Pr > F
Intercept      4.65631      6.39482      0.38795     0.53  0.4771
x3             0.51133      0.13075     11.19035    15.29  0.0012
x4            -0.12418      0.04466      5.65875     7.73  0.0134

Bounds on condition number: 1.0532, 4.2128
----------------------------------------------------------------------------------------

All variables left in the model are significant at the 0.0500 level.

No other variable met the 0.0500 significance level for entry into the model.

                       Summary of Stepwise Selection

        Variable     Variable     Number     Partial     Model
Step    Entered      Removed      Vars In    R-Square    R-Square     C(p)      F Value    Pr > F
  1     x3                            1      0.4772      0.4772      6.7253      15.52     0.0011
  2     x4                            2      0.1703      0.6476      1.6463       7.73     0.0134
```

2.5.4.8 Regresión con variables indicadoras.

Las variables indicadoras, que toman valor de 0 o 1 según se cumpla una condición, deben crearse en un paso data como se indica en el siguiente ejemplo. Recuerde que en la regresión debe omitirse una de las variables indicadoras relacionadas con el mismo factor para que el sistema sea soluble:

```
/*VARIABLES INDICADORAS, Ej. 14.9, Canavos, G. Probabilidad y Estadística. Aplicaciones y métodos*/
dm'log;clear;output;clear';
options nodate nocenter nonumber;
data uno;
input x1 y zona;
z2=(zona=2);/*crea la variable indicadora de la zona 2*/
z3=(zona=3);/*crea la variable indicadora de la zona 3*/
x12=x1*z2;
x13=x1*z3;
cards;
33.1 42.5 1
42.0 36.8 1
```

```
47.8 42.6 1
53.4 41.2 1
59.6 48.5 1
63.9 75.2 2
68.4 83.4 2
72.3 83.3 2
77.8 116.8 2
80.8 114.3 2
96.5 122.8 3
101.8 125.6 3
106.2 132.5 3
112.6 127.4 3
120.5 147.8 3
;
run;
proc print data=uno;run;
```

Obs	x1	y	zona	z2	z3	x12	x13
1	33.1	42.5	1	0	0	0.0	0.0
2	42.0	36.8	1	0	0	0.0	0.0
3	47.8	42.6	1	0	0	0.0	0.0
4	53.4	41.2	1	0	0	0.0	0.0
5	59.6	48.6	1	0	0	0.0	0.0
6	63.9	75.2	2	1	0	63.9	0.0
7	68.4	83.4	2	1	0	68.4	0.0
8	72.3	83.3	2	1	0	72.3	0.0
9	77.8	116.8	2	1	0	77.8	0.0
10	80.8	114.3	2	1	0	80.8	0.0
11	96.5	122.8	3	0	1	0.0	96.5
12	101.8	125.6	3	0	1	0.0	101.8
13	106.2	132.5	3	0	1	0.0	106.2
14	112.6	127.4	3	0	1	0.0	112.6
15	120.5	147.8	3	0	1	0.0	120.5

```
/*Considerando todas las zonas*/
proc reg data=uno;
model y=x1 z2 z3 x12 x13;
run;
quit;
```


Model: MODEL1

Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	21578	4315.60243	110.89	<.0001
Error	9	350.26520	38.91836		
Corrected Total	14	21928			

Root MSE	6.23846	R-Square	0.9840
Dependent Mean	89.38667	Adj R-Sq	0.9752
Coeff Var	6.97918		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	31.37244	14.65713	2.14	0.0610
x1	1	0.23246	0.30498	0.76	0.4654
z2	1	-128.80642	36.28669	-3.55	0.0062
z3	1	1.88863	38.82530	0.05	0.9623
x12	1	2.41118	0.54806	4.40	0.0017
x13	1	0.67861	0.45183	1.50	0.1674

```
/*REGRESIONES SEPARADAS POR ZONA. Menor precisión en la estimación de la desvación estándar del
modelo*/
proc reg data=uno;
by zona;
model y=x1;
run;quit;
zona=1
```

Model: MODEL1

Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	22.61019	22.61019	1.39	0.3228
Error	3	48.66181	16.22060		
Corrected Total	4	71.27200			

Root MSE	4.02748	R-Square	0.3172
Dependent Mean	42.34000	Adj R-Sq	0.0897
Coeff Var	9.51224		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	31.37244	9.46248	3.32	0.0452
x1	1	0.23246	0.19689	1.18	0.3228

zona=2

Model: MODEL1

Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	1311.74752	1311.74752	19.81	0.0211
Error	3	198.67248	66.22416		
Corrected Total	4	1510.42000			

Root MSE	8.13782	R-Square	0.8685
Dependent Mean	94.60000	Adj R-Sq	0.8246
Coeff Var	8.60235		

Parameter Estimates

Variaole	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-97.43398	43.30126	-2.25	0.1099
x1	1	2.64364	0.59400	4.45	0.0211

zona=3

Model: MODEL1

Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	290.67708	290.67708	8.47	0.0620
Error	3	102.93092	34.31031		
Corrected Total	4	393.60800			

Root MSE	5.85750	R-Square	0.7385
Dependent Mean	131.22000	Adj R-Sq	0.6513
Coeff Var	4.46388		

Parameter Estimates

Variable	OF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	33.26107	33.75690	0.99	0.3971
x1	1	0.91108	0.31301	2.91	0.0620

2.5.4.9 PROC LOGISTIC.

Permite estimar modelos para respuesta discreta, tales como modelos logísticos, regresión Poisson, etc.

```
PROC LOGISTIC <opciones>;
MODEL respuesta=independientes </opciones>;
BY variables;
OUTPUT <out=archivo-SAS><clave=nombre . . . clave=nombre>/alpha=valor>
WEIGHT variable;
```

Ejemplo: Supongamos que un investigador social trata de explicar usando el odds-ratio el hecho de que un adolescente fuma marihuana utilizando las variables: edad, sexo, e ingreso familiar. El programa en SAS para correr la regresión logística es el siguiente:

```
options ls=70;
DATA uno;
INPUT fumo $ edad sexo $ ingreso;
IF fumo='si' THEN fumo_mar=1;
ELSE fumo_mar=0;
IF sexo='m' THEN sex=1;
ELSE sex=0;
cards;
si 14 m 1.5
no 18 f 0.8
si 15 f 2.6
si 17 m 1.7
no 14 f 0.9
no 13 m 0.6
no 15 f 0.9
si 16 m 2.7
;
RUN;
PROC PRINT;
RUN;
PROC LOGISTIC DATA=uno;
MODEL fumo_mar=edad sex ingreso;
OUTPUT out=dos c=c cbar=cbar dfbetas=dfbetas difchisq=difchisq
               difdev=difdev h=h l=l p=p reschi = reschiresdev = resdev
               stdxbeta=stdxbeta u=u xbeta=xbeta;
RUN;
```

```
PROC PRINT;
RUN;
DATA tres;
SET dos;
odds_rat=p/(1-p);output;
RUN;
PROC PRINT;
VAR odds_rat;
RUN;
```

2.5.5 PROC FORMAT: DEFINIENDO FORMATOS.

El procedimiento FORMAT permite definir sus propios formatos de lectura o impresión de variables. Es muy útil debido a que permite identificar variables de una manera más clara para evitar confusiones y pérdida de tiempo.

Los formatos de lectura determinan la forma en que se leen y guardan los valores de los datos crudos. Los formatos de impresión determinan la forma en que los valores de las variables serán impresos.

Los formatos le indican al SAS el tipo (carácter o numérico) y la forma (cuántos bytes ocupa, cuántos decimales usar para los números, cómo manejar espacios en blanco y ceros, etc.) de los datos.

Con formatos de lectura, usted puede:

- Convertir un número a una cadena de caracteres (por ejemplo, convertir 1 a SI).
- Convertir una cadena de caracteres a una cadena de caracteres diferente (por ejemplo, convertir 'YES' a 'SI').
- Convertir una cadena de caracteres a un número (por ejemplo, convertido SI a 1)
- Convertir un número a otro número (por ejemplo, convierta 0 hasta 9 a 1, 10 hasta 100 a 2, y así sucesivamente.

Con formatos de impresión, usted puede:

- Imprimir valores numéricos como una cadena de caracteres (por ejemplo, imprima 1 como VARÓN y 2 como HEMBRA).
- Imprimir una cadena de caracteres como una cadena de caracteres diferente (por ejemplo, imprimir SÍ como YES).

- Imprimir valores numéricos que usan una plantilla (por ejemplo, imprima 9458763450 como 945-876-3450).

La sintaxis de este procedimiento es la siguiente:

```
PROC FORMAT <opciones>;
EXCLUDE entradas;
INVALUE <$>nombre <(opción(es)-de-formato-de-lectura)>grupo(s)-de-rango-de-valores;
PICTURE nombre <(opción(es)-de-formato-de-impresión)>grupo(s)-de-rango-de-valores-1
<(opción(es)-de-plantilla-1)>< grupo(s)-de-rango-de-valores-n <(opción(es)-de-plantilla-n)>
>;
SELECT      entradas;
VALUE<$>nombre <(opciones-de-formato)>grupo(s)-de-rango-de-valores;
```

2.5.5.1 Opciones en la línea del PROC:

EXCLUDE Excluye entradas del catálogo de ser procesadas por las opciones FMTLIB y CNTLOUT=

INVALUE Crea un formato para la lectura y conversión de los valores en datos crudos.

PICTURE Crea a plantilla para imprimir números.

SELECT Selecciona entradas del catálogo para ser procesadas por las opciones FMTLIB y CNTLOUT=

VALUE Crea un formato de impresión que especifica cadenas de caracteres a ser usadas para imprimir valores de variables.

Antes de entrar en detalle con la sintaxis del PROC FORMAT, se describirá la especificación de valores o rangos, la cual es fundamental en la definición de los formatos de lectura e impresión de las variables.

2.5.5.2 Especificando Valores o Rangos.

Como se indica en la sintaxis de las declaraciones INVALUE, PICTURE, y VALUE, se deben especificar grupo(s)-de-rangos-de-valores. En el lado izquierdo del signo igual se especifican los valores que usted quiere convertir. En el lado derecho del signo igual, se especifican los valores en los que usted quiere que los valores del lado izquierdo se conviertan.

Las declaraciones INVALUE, PICTURE, y VALUE aceptan valores numéricos en el lado izquierdo del signo igual. INVALUE y VALUE también aceptan cadenas de caracteres en el lado izquierdo del signo igual.

Se pueden tener múltiples ocurrencias de valor-o-rango en cada grupo(s)-de-rango-de-valores, dichas ocurrencias se separan por comas. Cada ocurrencia del valor-o-rango se da en alguna de las siguientes formas:

valor un valor simple, por ejemplo, 12 o'CA'. Encierre valores de tipo caracter entre comillas simples. Se puede usar la palabra clave OTHER como un valor simple. OTHER corresponde a todos aquellos valores que no coinciden a cualquier otro valor o rango.

range una lista de valores, por ejemplo, 12-68 o 'A'-'Z'. Para rangos con cadenas de caracteres, asegúrese de encerrar cada cadena entre comillas simples. Por ejemplo, si se quiere un rango que incluya cadenas de caracteres de la A a la Z, especifique el rango como 'A'-'Z'.
Si se especifica 'A-Z', el procedimiento interpreta esto como una cadena de caracteres de tres caracteres A como el primer caracter, un guión (-) como el segundo, y una Z como el tercer caracter.
Si se omiten las comillas, el procedimiento asume comillas alrededor de cada cadena. Por ejemplo, especificando el rango abc-zzz, el procedimiento interpreta esto como 'abc'-'zzz'.
Se pueden usar las palabras clave LOW o HIGH como un valor de un rango, y se puede usar el rango LOW-HIGH para abarcar todos los valores. Por ejemplo, estos son rangos válidos:

```
low-'ZZ'
35-high
low-high
```

También se puede usar el símbolo menor que (<) para excluir valores de los rangos. Por ejemplo, el siguiente rango no incluye a 0: `0<-100`
De otro lado, el siguiente rango no incluye a 100: `0-<100`

Los siguientes rangos muestran cómo evitar superponer rangos usando la notación ya definida:

```
'AA'-<'AJ'=1 'AJ'-'AZ'=2
```

AJ es parte del segundo rango, no del primero.

Si se superponen valores en los rangos, PROC FORMAT asigna el valor al primer rango. Por ejemplo, en los siguientes rangos, el valor AJ hace parte del primer rango:

```
'AA'-'AJ'=1 'AJ'-'AZ'=2
```

Cada valor-o-rango puede ser de hasta 200 caracteres. Si un valor-o-rango tiene más de 200 caracteres, el procedimiento trunca el valor en el caracter 200.

A continuación se describen las declaraciones dentro de PROC FORMAT más usadas.

Declaración proc format

```
PROC FORMAT <opciones>;
```

Opciones

CNTLIN= Especifica un archivo de datos de SAS, del cual el PROC FORMAT construye los formatos de lectura e impresión.

CNTLOUT= Crea un archivo de datos de SAS que guarda información acerca del formato de lectura e impresión.

FMTLIB Imprime información acerca de todos los formatos de lectura e impresión.

LIBRARY= Especifica un catálogo que contendrá los formatos de lectura o impresión que usted está creando en el paso actual del PROC FORMAT.

MAXLABLEN= Especifica el número de caracteres en el valor de formato de lectura o impresión que usted quiere que aparezca en la salida del PROC FORMAT.

MAXSELEN= Especifica el número de caracteres en los valores de inicio y salida que usted quiere que aparezcan en la salida del PROC FORMAT.

NOREPLACE Previene que un nuevo formato de lectura o impresión que usted está creando reemplace un formato de lectura o impresión existente del mismo nombre.

PAGE Imprime información de los formatos de lectura e impresión en páginas separadas.

Declaración value

Crea un formato que especifica cadenas de caracteres para usar en la impresión de valores de variables.

```
VALUE <$>nombre <(opción(es)-de-formato)> <grupo(s)-de-rango-de-valores>;
```

Opciones

DEFAULT= Especifica la longitud por defecto del formato de impresión.

FUZZ= Especifica un factor para valores que correspondan a un rango.

MAX= Especifica la máxima longitud para el formato de impresión.

MIN= Especifica la mínima longitud para el formato de impresión.

MULTILABEL Especifica valores múltiples para un rango dado, o para rangos superpuestos.

NOTSORTED Guarda valores o rangos en el orden que usted los definió.

Las opciones: DEFAULT=, FUZZ=, MAX=, MIN=, NOTSORTED, ya descritas anteriormente son comunes en las declaraciones INVALUE, PICTURE, and VALUE. MULTILABEL permite la asignación de etiquetas múltiples a los valores de las variables. La siguiente declaración VALUE muestra el uso de la opción MULTILABEL. La primera declaración VALUE asigna etiquetas múltiples a un solo valor. También pueden asignarse etiquetas múltiples a un solo rango de valores. La segunda declaración de VALUE asigna etiquetas a rangos superpuestos de valores. La opción MULTILABEL permite la asignación de etiquetas múltiples a los valores superpuestos.

```
value uno (multilabel)
   1='ONE'
   1='UNO'
   1='UN'
value edad (multilabel)
   15-29='below 30 years'
   30-50='between 30 and 50'
   51-high='over 50 years'
   15-19='15 to 19'
   20-25='20 to 25'
   25-39='25 to 39'
   40-55='40 to 55'
   56-high='56 and above' ;
```

Sólo los procedimientos PROC MEANS, PROC SUMMARY y PROC TABULATE pueden usar etiquetas múltiples. Todos los otros procedimientos reconocen sólo la etiqueta primaria. La etiqueta primaria para una entrada dada es el valor externo que se asigna al primer valor o rango de valores que coinciden o contienen la entrada cuando los valores se ordenan secuencialmente. Por ejemplo, en la primera declaración VALUE, la etiqueta primaria para el valor 1 es ONE, porque ONE es el primer valor que se asigna a 1. Las etiquetas secundarias para 1 son UNO y UN. En la segunda declaración

VALUE, la etiqueta primaria para el valor 33 es '25 to 39' porque el rango 25-39 es secuencialmente el primer rango de valores que contiene a 33. La etiqueta secundaria para 33 es 'between 30 and 50' porque el rango 30-50 ocurre en secuencia después del rango 25-39.

grupo(s)-de-rango-de-valores

Especifica uno o más valores de la variable y una cadena de caracteres o un formato existente. El(los) grupo(s)-de-rango-de-valores pueden ser alguno de los siguientes:

```
valor-o-rango-1<...,valor-o-rango-n>=valor-formateado|[formato-de-impresión-existente]
```

El valor de la variable en el lado izquierdo del signo igual se imprime como la cadena de caracteres en el lado derecho del signo igual.

valor-formateado

Especifica un formato soportado por SAS o un formato existente definido por el usuario. El formato que se crea usa el formato existente para convertir los datos crudos que correspondan al valor-o-rango en el lado izquierdo del signo igual.

formato-de-lectura-existente

Es un formato de lectura que es proporcionado por SAS o un formato de lectura definido por el usuario. El formato de lectura que se está creando usa el formato de lectura existente para convertir los datos crudos que correspondan al valor-o-rango en el lado izquierdo del signo igual. Si usted usa un formato de lectura existente, adjunte el nombre del formato de lectura entre corchetes, por ejemplo, [date9.] o entre paréntesis y barras verticales, por ejemplo, (|date9.|).

valor-o-rango (ver especificando valores o rangos) tiene los siguientes argumentos requeridos:

nombre nombre del formato que se está creando. El nombre debe ser un nombre SAS válido de ocho caracteres máximo y no acabar en un número. Los nombres de formato de carácter deben tener una señal del dólar ($) como el primer carácter y ninguno más de siete caracteres adicionales. Un formato usuario-definido no puede ser el nombre de un formato proporcionado por SAS. Refiérase después al formato usando el nombre seguido por un punto. Sin embargo, no pone un punto después del nombre del formato en la declaración de VALOR.

Ejemplo 1. Crea un formato que imprime el índice de natalidad de 36 países de América con los valores BAJO, MEDIO y ALTO.

El siguiente programa lee el archivo de datos países (definido en la sección 2.5) y crea una variable INAT que asigna un valor de 1 a valores de la variable x1 (natalidad) entre 10 y 20, un valor de 2 a valores de la variable x1 (natalidad) entre 20 y 30, un valor de 3 a valores de la variable x1 (natalidad) entre 30 y 40.

```
DATA PAISES;
SET PAISES;
IF (X1>=10 & X1<20) THEN INAT=1;
IF (X1>=20 & X1<30) THEN INAT=2;
IF (X1>=30 & X2<40) THEN INAT=3;
RUN;
```

A continuación se crea un formato IN que asigna a los valores de la variable INAT valores cualitativos que indican el nivel del índice de natalidad como BAJO, MEDIO y ALTO.

```
PROC FORMAT;
VALUE IN 1='BAJO' 2='MEDIO' 3='ALTO';
```

Por último se imprimen los países con su respectivo índice de natalidad.

```
PROC PRINT DATA=PAISES noobs;
FORMAT INAT IN.;
VAR PAIS INAT;
TITLE 'Indice de Natalidad de 36 Paises de América';
RUN;
```

A continuación se muestran los resultados de la ejecución de este programa:

Indice de Natilidad de 36 Paises de América

PAIS	INAT
Antigua	BAJO
Argentina	MEDIO
Bahamas	BAJO
Barbados	BAJO

Belice	ALTO
Bolivia	ALTO
Brasil	MEDIO
Canada	BAJO
Colombia	MEDIO
Costarica	MEDIO
Cuba	BAJO
Chile	MEDIO
Dominica	MEDIO
Ecuador	ALTO
Salvador	ALTO
USA	BAJO
PuertoRico	BAJO
Granada	ALTO
Guatemala	ALTO
Guyana	MEDIO
Haiti	ALTO
Honduras	ALTO
Jamaica	MEDIO
Mexico	MEDIO
Nicaragua	ALTO
Panama	MEDIO
Paraguay	ALTO
Peru	ALTO
R Dominicana	MEDIO
S Cristobal y Nevis	MEDIO
S Vicente y la Granadinas	MEDIO
Santa Lucia	MEDIO
Suriname	MEDIO
Trinidad y Tobago	MEDIO
Uruguay	BAJO
Venezuela	MEDIO

Ejemplo 2. Usa la declaración VALUE para crear un formato tipo caracter que imprime un valor de una variable tipo caracter como una cadena de caracteres diferente.

El archivo EMPRESA contiene información acerca de un pequeño subgrupo de empleados quiénes trabajan para una compañía que tiene sucursales en Colombia y Venezuela. Los datos incluyen el nombre, el número de identificación, el salario (en dólares) y el sitio de trabajo. A continuación se muestra el programa que genera e imprime el archivo EMPRESA.

```
data Empresa;
input Nombre & $16. IdNumero $ Salario Sucursal $;
cards;
Castañeda, Jaime  2355 2116 COL1
Centeno, Luis     5889 2098 COL1
Durango, Beatriz  3878 1957 COL2
Lopez, Brenda     4409 3432 COL2
Martinez, Maria   3985 4906 VEN2
Ortiz, Felipe     0740 5009 VEN2
Patiño, Maria     2398 3518 COL3
Sierra, Josefina  4421 3876 VEN1
Suarez, Roberto   5162 4010 COL5
Zuñiga, Carlos    7385 2299 COL3
;
run;

proc print data Empresa noobs;
title 'Archivo Empresa sin formatos';
run;
```

A continuación se muestra la salida del anterior programa.

Archivo Empresa sin formatos

Nombre	Id Numero	Salario	Sucursal
Castañeda, Jaime	2355	2116	COL1
Centeno, Luis	5889	2098	COL1
Durango, Beatriz	3878	1957	COL2
Lopez, Brenda	4409	3432	COL2
Martinez, Maria	3985	4906	VEN2
Ortiz, Felipe	0740	5009	VEN2
Patiño, Maria	2398	3518	COL3
Sierra, Josefina	4421	3876	VEN1
Suarez, Roberto	5162	4010	COL5
Zuñiga, Carlos	7385	2299	COL3

Ahora se usará la declaración VALUE para crear un formato tipo caracter que imprime la ciudad en donde se encuentra la sucursal de donde pertenece el empleado, además verifica que el código de la sucursal no haya sido mal digitado.


```
proc format;
value  Sciudad 'COL1'="Bogotá D.C. (COL)'
               'COL2'="Medellin (COL)'
               'COL3'='Cali (COL)'
               'VEN1'='Caracas (VEN)'
               'VEN2'='Maracaibo (VEN)'
               other='CODIGO INCORRECTO';
run;

proc print data=Empresa noobs;
format Sucursal Sciudad.;
title 'Archivo Empresa con un Formato para la Variable Sucursal';
run;
```

Lo anterior produce el siguiente resultado:

Archivo Empresa con un Formato para la Variable Sucursal

Nombre	Id Numero	Salario	Sucursal
Castañeda, Jaime	2355	2116	Bogotá D.C. (COL)
Centeno, Luis	5889	2098	Bogotá D.C. (COL)
Durango, Beatriz	3878	1957	Medellin (COL)
Lopez, Brenda	4409	3432	Medellin (COL)
Martinez, Maria	3985	4906	Maracaibo (VEN)
Ortiz, Felipe	0740	5009	Maracaibo (VEN)
Patiño, Maria	2398	3518	Cali (COL)
Sierra, Josefina	4421	3676	Caracas (VEN)
Suarez, Roberto	5162	4010	CODIGO INCORRECTO
Zuñiga, Carlos	7385	2299	Cali (COL)

El formato SCIUDAD. convierte cada uno de los códigos COL1, COL2, etc., en el nombre de la correspondiente ciudad. La palabra clave OTHER formatea valores en los archivos de datos que no corresponden a cualquiera de los valores del lado izquierdo de los signos igual (COL1, COL2, ..., VEN2) como CODIGO INCORRECTO.

Ejemplo 3. Crea un formato y muestra como usar rangos con cadenas de caracteres.

El siguiente programa crea e imprime el archivo ENTREN a partir del archivo EMPRESA creado anteriormente.

```
data entren;
set Empresa(keep=nombre idnumero);
apellido=SUBSTR(nombre,1,6);
run;

proc print data=entren noobs;
title 'Archivo ENTREN sin Formatos';
run;
```

A continuación se muestra los resultados de correr el programa anterior. La función SUBSTR extrae de la variable NOMBRE los primeros 6 caracteres.

Archivo ENTREN sin Formatos

Nombre	Id Numero	apellido
Castañeda, Jaime	2355	Castan
Centeno, Luis	5889	Centen
Durango, Beatriz	3878	Durang
Lopez, Brenda	4409	Lopez,
Martinez, Maria	3985	Martin
Ortiz, Felipe	0740	Ortiz,
Patiño, Maria	2398	Patiño
Sierra, Josefina	4421	Sierra
Suarez, Roberto	5162	Suarez
Zuñiga, Carlos	7385	Zuñiga

Ahora se creará un formato que asigna una prueba de habilidad a los empleados de la empresa (como parte de un entrenamiento), esto dependiendo del apellido de los mismos. Lo anterior utilizando los rangos para cadenas de caracteres.

```
proc format;
value $habil  'a'-<'e','A'-<'E'='Prueba A'
              'e'-<'m','E'-<'M'="Prueba B'
              'm'-'z-','M'-'Z-'='Prueba C';
run;
proc print data=entren noobs label;
var nombre idnumero apellido;
format apellido $habil.;
label apellido='Prueba';
title "Asignación de Pruebas para Cada Empleado';
run;
```


La siguiente es la salida del procedimiento anterior.

Asignación de Pruebas para Cada Empleado

Nombre	Id Numero	Prueba
Castañeda, Jaime	2355	Prueba A
Centeno, Luis	5889	Prueba A
Durango, Beatriz	3878	Prueba A
Lopez, Brenda	4409	Prueba B
Martinez, Maria	3985	Prueba C
Ortiz, Felipe	0740	Prueba C
Patiño, Maria	2398	Prueba C
Sierra, Josefina	4421	Prueba C
Suarez, Roberto	5162	Prueba C
Zuñiga, Carlos	7385	Prueba C

La declaración VALUE crea el formato SHABIL., el cual imprime para cada empleado el número de identificación y la prueba que le ha sido asignada. Los empleados deben tomar alguna prueba entre PRUEBA A, PRUEBA B, o PRUEBA C, dependiendo de su apellido. El operador de exclusión (<) excluye el último valor del rango. Así, el primer rango incluye empleados cuyo apellido comience con cualquier letra entre A y D, y el segundo rango incluye empleados cuyo apellido comience con cualquier letra entre E y M. La tilde (~) en el último rango es necesaria para incluir una cadena entera que comience con la letra Z.

2.5.6 PROC TABULATE: TABLAS Y REPORTES.

El procedimiento TABULATE muestra estadísticas descriptivas en formato de tabla usando algunas o todas las variables en un archivo de datos. Se puede crear una gran variedad de tablas que pueden ser poco o ampliamente personalizadas. PROC TABULATE calcula muchas de las estadísticas que también son calculadas por otros procedimientos estadísticos descriptivos como MEANS, FREQ, y REPORT.

El PROC TABULATE proporciona:

- Métodos simples pero poderosos para crear informes en formato de tabla.
- Flexibilidad clasificando valores de variables y estableciendo relaciones jerárquicas entre las mismas.

- Mecanismos para etiquetar y formatear variables y estadísticas generadas por algún procedimiento.

Por qué usarlo?

El software SAS posibilita el análisis de datos de centenares de formas. Se puede usar el paso DATA para dividir y organizar sus datos, y hay docenas de procedimientos que procesarán y producirán todo tipo de estadísticas. Pero es probable que sin importar cómo se organicen y analicen sus datos, se terminará produciendo un reporte en forma de tabla. El PROC TABULATE sirve entonces para la preparación de reportes. Este procedimiento no hace nada que usted no puede hacer con otros procedimientos, la diferencia radica en la salida. PROC TABULATE calcula una gran variedad de estadísticas, y genera de una manera sencilla los resultados agrupados en una sola tabla. Así que en lugar de ejecutar dos o tres procedimientos y generar una gran cantidad de resultados, en los cuales, se pierden los estadísticos que verdaderamente importan en el informe, o lo que es peor, tener que volver a digitar parte de los resultados en una tabla, el PROC TABULATE se puede acondicionar para crear una sola salida lista para la impresión.

Sintaxis

```
PROC TABULATE <opciones>;  BY <DESCENDING> variable-1 <...<DESCENDING> variable-n>
<NOTSDRTED>;
CLASS variable(s) </ opciones>;
CLASSLEV variable(s) / style =<style-element-name | <PARENT>> <[ especificaciones-atributo-
estilo]>;
FREQ variable;
KEYLABEL palabra_clave-1='descripcion-1' <...palabra_clave-n='descripcion-n'>;
PALABRA_CLAVE palabra_clave(s) / style =<nombre-elemento-estilo | <PARENT>> <[
especificaciones-atributo-estilo]>;
TABLE <<expression-página,> expression-fila,> expression-columna </ tabla-opciones>;
VAR variable(s)-análisis</ opciones>;
WEIGHT variable;
```

Declaraciones

BY Crea una tabla separada para cada grupo BY.

CLASS Identifica las variables en el archivo de entrada como variables de clase.

CLASSLEV Especifica un estilo para los encabezados de los niveles de las variables clase.

FREQ Identifica una variable en el archivo de entrada cuyos valores representan la frecuencia de cada observación.

KEYLABEL Especifica una etiqueta para una palabra clave.

KEYWORD Especifica un estilo para los encabezados de la palabra clave.

TABLE Describe la tabla que será creada.

VAR Identifica las variables en el archivo de entrada como variables a ser analizadas.

WEIGHT Identifica una variable en el archivo de entrada cuyos valores ponderan cada observación en los cálculos estadísticos.

Declaración PROC TABULATE

```
PROC TABULATE DATA= <opcion(es)>;
```

Opciones

CONTENTS= Personaliza el vínculo a la salida de contenidos HTML.

NOTRAP Deshabilita la recuperación de la excepción de punto flotante.

OUT= Especifica el archivo de salida.

TRAP Habilita la recuperación de la excepción de punto flotante.

Identifica las categorías de los datos que son de interés

CLASSDATA= Especifica un archivo de datos secundario que contiene las combinaciones de valores de las variables de clase a ser incluidas en las tablas y archivos de salida.

EXCLUSIVE Excluye de las tablas y archivos de salida todas las combinaciones de valores de las variables de clase que no están en el archivo CLASSDATA=

MISSING Considera que los valores faltantes son valores válidos para las variables de clase

Controles para el Análisis Estadístico

EXCLNPWGTS Excluye observaciones con ponderaciones no positivas.

QMARKERS= Especifica el tamaño de muestra a ser usado para el método de estimación P2 del cuantil.

QMETHOD= Especifica el método de estimación del cuantil.

QNTLDEF= Especifica la definición matemática para calcular cuantiles.

VARDEF= Especifica el divisor de la varianza.

Personaliza la Apariencia de la Tabla

FORMAT= Especifica un formato por defecto para cada celda en la tabla

FORMCHAR= Define los caracteres a ser usados para construir los bordes y divisores de la tabla

NOSEPS Elimina las líneas separadoras horizontales de los títulos de fila y el cuerpo de la tabla

ORDER= Ordena los valores de una variable de clase de acuerdo a un orden especificado

STYLE= Especifica el(los) elemento(s) de estilo por defecto (para el sistema de entrega de salida) a ser usado por cada celda en la tabla

Detalles

`ORDER=DATA | FORMATTED | FREQ | UNFORMATTED`

Especifica el orden en que se crean las combinaciones de los valores de las variables de clase, los cuales forman los encabezados de la tabla de acuerdo al orden especificado, donde:

DATA Ordena los valores de acuerdo a su orden en el archivo de entrada.

FORMATTED Ordena los valores por sus valores formateados. Por defecto, el orden es ascendente.

FREQ Ordena los valores por frecuencia de conteo descendente.

INTERNAL Ordena los valores por sus valores no formateados, el cual coincide con el mismo orden del PROC SORT.

UNFORMATTED Ordena los valores por sus valores sin formatos, el cual coincide con el mismo orden del PROC SORT. Esta secuencia de orden es útil para mostrar fechas cronológicamente.

Por defecto `ORDER=UNFORMATTED`

QMETHOD=OS|P2 Especifica el método que el PROC TABULATE usa para procesar el archivo de entrada cuando calcula cuantiles. Si el número de observaciones es menor o igual a el valor en QMARKERS= y QNTLDEF=5, ambos métodos producen los mismos resultados.

OS Usa estadísticos de orden. Igual que lo hace el PROC UNIVARIATE.

P2 Usa el método P2 para aproximar el cuantil.

Por defecto `QMETHOD=OS`

VARDEF=divisor Especifica el divisor a ser usado en el cálculo de la varianza y desviación estándar. A continuación se muestran posibles valores para VARDEF= y los divisores asociados. Valores Posibles para VARDEF=

Valor	Divisor	Formula par el Divisor
DF	Grados de libertad	n - 1
N	Número de observaciones	n
WDF	Suma de pesos menos uno	SUM(wi) - 1
WEIGHT|WGT	Suma de pesos	SUM(wi)

Por defecto `VARDEF=DF`

A continuación se describirán algunas de las declaraciones más utilizadas.

DECLARACIÓN BY

Crea una tabla separada sobre una página separada para cada grupo BY

```
BY <DESCENDING> variable-1 <...<DESCENDING> variable-n> <NOTSORTED>;
```

Argumentos Requeridos

variable especifica la variable que el procedimiento usa para formar los grupos BY. Usted puede especificar más de una variable. Si no se usa la opción NOTSORTED en la declaración BY, las observaciones en la base de datos deben estar ordenadas por todas las variables que usted especificó, o se deben indexar apropiadamente.

Opciones

DESCENDING Especifica que las observaciones están ordenadas en forma descendiente por la variable que sigue inmediatamente de la palabra DESCENDING en la declaración BY.

NOTSORTED Especifica que las observaciones no están necesariamente dadas en orden alfabético o numérico. Las observaciones son agrupadas en otra forma, por ejemplo, en orden cronológico. El requerimiento de ordenar o indexar las observaciones según los valores de las variables es suspendido cuando usted usa la opción NOTSORTED. De hecho, el procedimiento no usa un índice si usted especifica NOTSORTED. El procedimiento define un grupo como un conjunto de observaciones consecutivas que tienen los mismos valores, esto para todas las variables. Si las observaciones con los mismos valores no son consecutivas, el procedimiento toma cada conjunto consecutivo como un grupo BY diferente.

Declaración CLASS

Identifica la variable de clase para la tabla. Las variables de clase determinan las categorías que el PROC TABULATE usa para calcular estadísticos.

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

Argumentos Requeridos

variable(s) Especifica una o más variables que el procedimiento usa para agrupar los datos. Las variables en la declaración CLASS son llamadas como variables de clase. Las variables de clase pueden ser numéricas o tipo caracter. Las variables de clase pueden tener valores contiguos, pero ellos generalmente tienen pocos valores discretos que definen la clasificación de la variable. No es necesario ordenar los datos por las variables de clase.

Declaración FREQ

Especifica una variable numérica que contiene la frecuencia de cada observación.

```
FREQ variable;
```

Argumentos requeridos

variable Especifica una variable numérica cuyos valores representan la frecuencia de cada observación. Si se usa la declaración FREQ, el procedimiento asume que cada observación representa n observaciones, donde n es el valor entero de la variable. Si n no es un entero, SAS lo trunca. Si n es menor que 1 o es un valor faltante, el procedimiento no usa esa observación para calcular estadísticos. La suma de la variable frecuencia representa el número total de observaciones.

Declaración KEYLABEL

Etiqueta una palabra clave durante la ejecución del PROC TABULATE. PROC TABULATE usa la etiqueta en cualquier parte donde la correspondiente palabra clave debería aparecer.

```
KEYLABEL palabra-clave-1='descripción-1' <...palabra-clave-n='descripción-n'>;
```

Argumentos Requeridos

palabra-clave Es una de las palabras clave para los estadísticos que se muestran a continuación.

COLPCTN	PCTSUM
COLPCTSUM	RANGE
CSS	REPPCTN

CV	REPPCTSUM
MAX	ROWPCTN
MEAN	ROWPCTSUM
MIN	STDDEV|STD
N	STDERR
NMISS	SUM
PAGEPCTN	SUMWGT
PAGEPCTSUM	USS
PCTN	VAR

Palabras clave de estadísticos de cuantiles

MEDIAN|P50	Q3|P75
P1	P90
P5	P95
P10	P99
Q1|P25	ORANGE

Palabras clave de pruebas de hipótesis

PROBT	T

Descripción Es una cadena de hasta 256 caracteres usada como etiqueta. Como lo muestra la sintaxis, se debe encerrar la descripción entre comillas simples.

Declaración TABLE

```
TABLE <<expresión-de-página,> expresión-de-fila,> expresión-de-columna </ opción(es)-table>;
```

Argumentos Requeridos

expresión-de-columna Define las columnas en la tabla.

Opciones

Para adicionar dimensiones:

expresión-de-página Define las paginas en la tabla.

expresión-de-fila Define las filas en la tabla.

CONTENTS= Personaliza los vínculos de entrada a contenidos HTML en la salida.

STYLE= Especifica un elemento de estilo para varias partes de la tabla.

Para personalizar texto en la tabla:

BOX= Especifica el texto a ser colocado en la caja vacía encima de los títulos de fila.

MISSTEXT= Suministra hasta 256 caracteres para imprimir en las celdas de tabla que contiene valores faltantes.

NOCONTINUED Suprime el mensaje de continuación para tablas que se dividen en múltiples páginas físicas.

Para modificar el esquema de la tabla:

CONDENSE Imprime tantas páginas lógicas completas como es posible sobre un página simple impresa o, si es posible, imprime múltiples páginas de tablas que son muy amplias en una sóla página, una debajo de la otra sobre una página simple, en lugar de en páginas separadas.

PRINTMISS Crea los mismos encabezados de fila y columna para todas las páginas de la tabla

Personaliza encabezados de fila:

INDENT= Especifica el número de espacios de sangría para encabezados de filas anidadas.

ROW= Controla la asignación de espacio para los títulos de la fila dentro del espacio disponible.

RTSPACE= Especifica el número de posiciones de impresión disponibles para los títulos de fila.

Declaración VAR

```
VAR variable(s)-de-análisis </ opción(es)>;
```

Argumentos Requeridos

variable(s)-de-análisis Identifica las variables de análisis en la tabla. Las variables de análisis son numéricas para las cuales PROC TABULATE calcula estadísticos. Los valores de una variable de análisis pueden ser continuos o discretos. Si una observación contiene un valor faltante para una variable de análisis, PROC TABULATE omite ese valor de los cálculos de todos los estadísticos excepto N (el número de observaciones con valores "no faltantes" de la variable) y NMISS (el número de observaciones con valores faltantes de la variable). Por ejemplo, el valor faltante no incrementa el total (SUM), y no es contado cuando se calculan estadísticos como la media (MEAN).

Opciones

```
STYLE=<style-element-name | <PARENT>><[style-attribute-specification(s)>]
```

Especifica un elemento de estilo para el encabezado del nombre de la variable de análisis.

WEIGHT=variable-peso Especifica una variable numérica cuyos valores ponderan los valores de las variables que están especificadas en la declaración VAR. La variable no tiene que ser un entero. Si el valor de la variable weight es:

0 toma en cuenta la observación en el número total de observaciones.

menor que 0 convierte el valor a cero y toma en cuenta la observación en el número total de observaciones.

valor faltante excluye la observación.

Declaración WEIGHT

```
WEIGHT variable;
```

Argumentos Requeridos

variable Especifica una variable numérica cuyos valores ponderan los valores de las variables de análisis. La variable no tiene que ser un entero.

Ejemplo 1.

El siguiente ejemplo ilustra las similitudes y diferencias entre PROC TABULATE y otros procedimientos de SAS, realizando el mismo análisis usando primero el PROC MEANS y luego usando el PROC TABULATE.

Enfoque usando PROC MEANS. Se tienen datos acerca de los gastos de energía por dos tipos de clientes, residencial y comercial, en los estados de las regiones Nordeste (1) y Oeste (4) de los Estados Unidos. El objetivo del analista es establecer el gasto promedio: (1) para las 2 tipos de cliente y (2) para las tipos de cliente dentro de cada una de las regiones analizadas. Para resolver el punto (1), donde se quiere obtener el gasto promedio para los tipos de cliente, se usa el siguiente código:

```
data energia;
length State $2;
input Region Division Estado $ Cliente Gastos @@;
cards;
1 1 ME 1 708 1 1 ME 2 379 1 1 NH 1 597 1 1 NH 2 301 1 1 VT 1 353 1 1 VT 2 188 1 1 MA 1 3264 1 1 MA 2
2498 1 1 RI 1 531 1 1 RI 2 358 1 1 CT 1 2024 1 1 CT 2 1405 1 2 NY 1 8786 1 2 NY 2 7825 1 2 NJ 1 4115 1
2 NJ 2 3558 1 2 PA 1 6478 1 2 PA 2 3695 4 3 MT 1 322 4 3 MT 2 232 4 3 ID 1 392 4 3 ID 2 298 4 3 WY 1
194 4 3 WY 2 184 4 3 CO 1 1215 4 3 CO 2 1173 4 3 NM 1 545 4 3 NM 2 578 4 3 AZ 1 1694 4 3 AZ 2 1448 4 3
```