

## UN PROGRAMA GAUSS PARA SIMULAR DISTRIBUCIONES NO NORMALES MULTIVARIADAS

Juan A. Hernández Cabrera, Concepción San Luis Costas y Alfonso Sánchez Bruno  
Universidad de La Laguna

Se presenta dos programas en GAUSS, que permite generar muestras de tamaño  $n$  y  $p$  variables con características de distribución en asimetría y apuntamiento definidos previamente por el usuario, a partir de la matriz de correlaciones de las variables implicadas, en base a los algoritmos de Fleishman (1978) y Vale y Maurelli (1983).

*A GAUSS program to simulate non normal multi ariate distributions.* We present two GAUSS programs, that allow sample's generation of size  $n$  and  $p$  variables with distribution aspects in skewness and kurtosis beforehand specify by the users, from the involves variables correlation's matrix according to Fleishman (1978) and Vale & Maurelli (1983) algorithms.

Los analistas de datos de las ciencias sociales a menudo trabajan con la creencia implícita de que estos provienen de una población normal univariada o multivariada. Sin embargo, sabemos que esta premisa raramente se cumple dado que la mayoría de las variables psicológicas presentan grados de asimetría y apuntamiento que la vulneran claramente. Se necesita por tanto, una metodología de trabajo que ayude a determinar el impacto que el incumplimiento de la condición de normalidad de una determinada prueba estadística puede tener sobre el proceso de la toma de la decisión de rechazar o no la hipótesis nula con la que se trabaja. En este sentido, los procedimientos de simulación de Monte Carlo, se utilizan para modelizar y comprobar la bondad, potencia y robustez de

múltiples estadísticos con la intención de descubrir los puntos de inflexión que vulneran las condiciones de aplicación de las distintas pruebas estadísticas.

Fleishman (1978) desarrolló una transformación polinomial que él denominó «método de potencia» que permite generar muestras con valores de asimetría y apuntamiento seleccionados previamente, partiendo de una variable normal ( $N(0,1)$ ). Este método presenta una serie de ventajas entre las que destaca la posibilidad de determinar el grado de asimetría y apuntamiento deseado, tener un funcionamiento altamente fiable y ser de fácil implementación. La transformación polinomial es la siguiente:

$$Y = a + bX + cX^2 + dX^3 \quad (I)$$

Las constantes  $a$ ,  $b$ ,  $c$  y  $d$  son seleccionadas para lograr una distribución específica de  $Y$  en asimetría y apuntamiento. Con una considerable manipulación algebraica, Fleishman presenta la solución de

---

Correspondencia: Juan A. Hernández Cabrera  
Universidad de La Laguna

las constantes de la Ecuación (I) en un sistema de ecuaciones no lineales.

$$\begin{aligned} b^2 + 6bd + 2c^2 + 15d^2 &= 0 \\ 2c(b^2 + 24bd + 105d^2 + 2) - \gamma_1 &= 0 \quad (\text{II}) \\ 24 [bd + c(1 + b^2 + 28bd) + d^2 (12 + 48bd + 141c^2 + 225d^2)] - \gamma_2 &= 0 \end{aligned}$$

Donde  $\gamma_1$  y  $\gamma_2$  son respectivamente la asimetría y apuntamiento que se desea para la distribución de la variable. La obtención de  $b$ ,  $c$  y  $d$  puede realizarse resolviendo el sistema no lineal planteado en la Ecuación II mediante el procedimiento iterativo de Newton:

$$\begin{aligned} ft(b,c,d) &= b^2 + 6bd + 2c^2 + 15d^2 - 1 = 0 \quad (\text{III.1}) \\ gt(b,c,d) &= 2c(b^2 + 24bd + 105d^2 + 2) - \gamma_1 = 0 \quad (\text{III.2}) \\ ht(b,c,d) &= 24 [bd + c(1 + b^2 + 28bd) + d^2 (12 + 48bd + 141c^2 + 225d^2)] - \gamma_2 = 0 \quad (\text{III.3}) \end{aligned}$$

Donde las primeras derivadas parciales para cada una de las tres ecuaciones respecto a los parámetros  $b$ ,  $c$  y  $d$  son, respectivamente:

$$\begin{aligned} f_b &= \frac{\partial f_t}{\partial b} = 2b + 6b \\ f_c &= \frac{\partial f_t}{\partial c} = 4C \\ f_d &= \frac{\partial f_t}{\partial d} = 6b + 30d \end{aligned} \quad (\text{IV.1})$$

$$\begin{aligned} g_b &= \frac{\partial g_t}{\partial b} = 4bc + 24dc \\ g_c &= \frac{\partial g_t}{\partial c} = 2b^2 + 24bd + 105d^2 + 2 \\ g_d &= \frac{\partial g_t}{\partial d} = 24bc + 210cd \end{aligned} \quad (\text{IV.2})$$

$$\begin{aligned} h_b &= \frac{\partial h_t}{\partial b} = 24d + 48bc^2 + 672c^2d + 1152d^2 \\ h_c &= \frac{\partial h_t}{\partial c} = 48c + 48b^2c + 1344bcd + 6768cd^2 \\ h_d &= \frac{\partial h_t}{\partial d} = 24b + 672bc^2 + 576d + 3456bd^2 + 6768c^2d \\ &\quad + 16200d^3 \end{aligned} \quad (\text{IV.3})$$

Aplicando el procedimiento de aproximaciones sucesivas de Newton, tenemos

el siguiente sistema de ecuaciones, cuya resolución determinará los incrementos de  $b$ ,  $c$  y  $d$ .

$$f'bx + f'cy + f'dz = -ft \quad (\text{V.1})$$

$$g'bx + g'cy + g'dz = -gt \quad (\text{V.2})$$

$$h'bx + h'cy + h'dz = -ht \quad (\text{V.3})$$

**X**, **Y** y **Z**, son las constantes que sucesivamente sumadas a los valores de comienzo de las incógnitas **b**, **c** y **d** aproximarán la solución para estos parámetros, hasta obtener la convergencia y por tanto la solución del sistema planteado. Resolviendo el sistema anterior para **Z**, **X** e **Y** obtenemos:

$$\begin{aligned} Z &= \frac{(f_t h_b - f_b h_t)(f_b g_c - f_c g_b) - (f_t g_b - f_b g_t)(f_b h_c - f_c h_b)}{(f_b h_d - f_d h_b)(f_b g_c - f_c g_b) - (f_b g_d - f_d g_b)(f_b h_c - f_c h_b)} \\ Y &= \frac{(f_t g_b - f_b g_t) - Z(f_b g_d - f_d g_b)}{(f_b g_c - f_c g_b)} \quad (\text{VI.1,2,3}) \end{aligned}$$

$$X = \frac{(-f_t - f_c Y - f_d Z)}{f_b}$$

En las páginas 4 y 5, se presenta el algoritmo GAUSS que permite el cálculo de  $a$ ,  $b$ ,  $c$  y  $d$  a partir del algoritmo de Newton planteado en las Ecuaciones III, IV y V para una determinada combinación de asimetría ( $\gamma_1$ ) y apuntamiento ( $\gamma_2$ ).

Vale y Maurelli (1983), utilizaron el algoritmo de Fleishman (1978) para lograr una distribución no normal multivariante utilizando como entrada la matriz de correlaciones de las variables deseadas. A partir de ella, se determinan los componentes principales de la misma, permitiendo, una vez generadas las ecuaciones factoriales, la creación de datos brutos con distribución normal y un patrón de correlación similar al presentado por la matriz inicial. Sin embargo, para generar esta matriz de datos es necesario anticipar y tener en cuenta el efecto que la transformación de potencia tendrá sobre las características de distri-

bución deseada para los datos. Esta anticipación la llevan a cabo los autores generando la matriz de correlación de

$$\begin{aligned} r_{yj} = & \rho_{x_i x_j} (bb_j + 3bd_j + 3d_b b_j + 9d_d d_j) \\ & + \rho_{x_i x_j}^2 (2c_i c_j) + \rho_{x_i x_j}^3 (6d_i d_j) \end{aligned} \quad (\text{VII})$$

transición mediante el siguiente algoritmo:

La matriz resultante se descompondrá en sus componentes principales, permitiendo una vez determinadas las variables factoriales así generadas, la transformación de las mismas mediante el algoritmo de potencia de Fleishman(1978).

#### Descripción de los programas

Las rutinas han sido realizadas en lenguaje GAUSS, el cual permite escribir programas muy compactos de forma rápida, sencilla y eficiente. Seguidamente se presenta la secuencia de líneas de los programas. Los números entre paréntesis que preceden a algunas de ellas, tienen sólo un carácter de etiqueta cuyo único objetivo, es facilitar la tarea de descripción del programa.

Se describirán dos rutinas: la primera realiza el cálculo de los parámetros a b c y d de la transformación de potencia de Fleishman(1978) (Ec. 1) (Páginas 4 y 5), mientras que la segunda genera distribuciones no normales multivariadas (Páginas 7,8 y 9).

#### *Descripción del programa para estimar a b c y d*

En (1) se declara el procedimiento NEWTON, indicándose entre paréntesis los dos parámetros necesarios para su ejecución: asime (grado de asimetría deseado) y kurto (apuntamiento). Tal y como puede verse, el procedimiento devuelve 4 parámetros a saber: a, b, c y d.

```
(1) P R O C 4 = N E W T O N
(asime,kurto);
(2) LOCAL i,itera,ft,fb,fc,fd,gti,gb,
gc,gd,ht,hb,hc,hd,a,b,c,d,x,y,z,z1,z2,
bconv,converge,repetir;
(3) repetir=0;
      b=2.5;
      c=.5;
      d=.5;
(4) inicio;;
      if repetir > 1;
          CLS;
          print «NO SE LOGRO
CONVERGENCIA PARA ASIME-
TRIA= « ASIME;
          PRINT «Y APUNTA-
MIENTO= « KURTO;
          PRINT;
          print «PULSA CRTL/C Y
CTRL/F2 PARA INICIAR OTRA
COMBINACION»;
          WAIT;
      endif;
(5) if repetir == 1;
      b=.5;
      c=.5;
      d=.5;
      endif;
```

En (2) mediante la sentencia LOCAL, se declaran las variables que sólo van a participar en este procedimiento. Téngase en cuenta que un procedimiento GAUSS, es una subrutina, que puede ser llamada desde cualquier otro programa GAUSS. La sección (3) y

(5), contienen dos grupos de valores de comienzo que aseguran la convergencia para la mayoría de las combinaciones de asimetría y apuntamiento utilizadas en la literatura Boomsma, A. (1983), Babakus, E., Ferguson, C.E., y Joreskog, K.G. (1987), Chou, C.P., Bentler, P.M. y Satorra, A. (1991), Sharma, S., Durvasula, S. y Dillon, W.R. (1989). En el caso de que no se lograse la convergencia, para ninguno de los dos valores de comienzo propues-

tos, en el bloque (4) el programa presenta el mensaje con esta información.

```

(6) i=1;
    itera=200;
    do while i <= itera;
    (7) ft=b^2+6*b*d+2*c^2+15*d^2-1;
        gti=2*c*(b^2+24*b*d+105*d^2+2)—
(asime);
        ht = 24* (b*d+c^2* (1+b^2 +
28*b*d) +
d^2*(12+48*b*d+141*c^2+
225*d^2))—(kurto);
    (8) fb=2*b+6*d;
        fc=4*c;
        fd=6*b+30*d;
    (9) gb=4*b*c+24*d*c;
        gc=2*b^2+24*b*d+105*d^2+2;
        gd=24*b*c+210*c*d;
    (10) hb=24*d+48*c^2*b + 672*c^2*d
+1152*d^2;
        hc=48*c+48*c*b^2 +1344*c*b*d
+6768*d^2*c;
        hd=24*b+672*c^2*b+576*d
+3456*b*d^2+6768*c^2*d+16200*d^3;
    (11) z1=(ft*hb-fb*ht)*(fb*gc-fc*gb)-
(ft*gb-fb*gti)*(fb*hc-fc*hb);
        z2=(fb*h d-fd*h b)*(fb*g c-
fc*gb) —(fb*gd-fd*gb)*(fb*hc-fc*hb);
        z=z1/z2;
        y=((ft*gb-fb*gti) —z*(fb*gd-
fd*gb))/(fb*ge-fc*gb);
        x=(-ft-fc*y-fd*z)/fb;
        bconv=b;
    (12) b=b+x;
        c=c+y;
        d=d+z;
    (13) converge=b-bcon;
        i=i+1;
        if converge == 0;
            goto fin;
            endif;
        endo;
        if ABS(converge) > 0;
            repetir=repetir+1;
            goto inicio;;
        endif;
        fin;;

```

cls;

a=-c;

(14) RETP(a,b,c,d);  
endp;

En (6) se inicia el bucle iterativo que persigue conseguir convergencia en un máximo de 200 iteraciones. En (7) se especifica las tres ecuaciones de Fleishman(1978) para estimar los parámetros b c y d. En (8) (9) y (10) se especifican las tres derivadas parciales para cada ecuación respecto de los tres parámetros según las Ec. (IV.1, IV.2 y IV.3). En (11) se desarrollan los incrementos sucesivos de b c y d que serán aplicados en (12). En cada iteración se comprueba en (13) la convergencia del proceso. Una vez producida, el procedimiento devuelve en (14) los tres parámetros de la transformación de potencia. En caso de que ésta no se produzca con los valores de (3), se inicia nuevamente el proceso iterativo con los valores de comienzo de (5).

El programa GAUSS, permite la compilación de los procedimientos mediante la orden: COMPILE NOMBRE DEL PROCEDIMIENTO. Como es evidente, este compilado permite disminuir notablemente los tiempos de ejecución de las aplicaciones creadas en modo procedimiento.

A continuación, se presenta un ejemplo de llamada a esta subrutina de nombre NEWTON, para la obtención de los tres parámetros de la ecuación polinomial Ec. (I). En (1) se utiliza la orden USE que permite cargar en memoria la subrutina compilada previamente. En (2) se solicita la combinación de asimetría y apuntamiento deseadas.

```

(1) USE NEWTON;
(2) print «INTRODUCE LA ASI-
METRIA QUE DESEAS»;
    asime=con(1,1);
    print «INTRODUCE LA KUR-
TOSIS QUE DESEAS»;

```

```

kurto=con(1,1);
(3) {a,b,c,d}=newton(asime,kurto);
asi1=zeros(10,1);
kurt1=zeros(10,1);
i=1;
(4) do while i <= 10;
y=rndn(2000,1);
yt=a+b*y+c*y^2+d*y^3;
A S I = ( S U M C ( ( y t -
MEANC(yt))^3)/(ROWS(yt)*STDC(yt)^
3));
KUR=(SUMC((yt-MEANC(yt))^4)
/(ROWS(yt)*STDC(yt)^4))-3;
asi1[i]=asi;
kurt1[i]=kur;
i=i+1;
endo;
(5) asime=meanc(asi1);
kurto=meanc(kurt1);
PRINT; PRINT;
print «*****»;
print « ASIMETRIA PRO-
PUESTA/OBTENIDA=> GAM1 asime;
print « APUNTAMI. PRO-
PUESTO/OBTENIDO=> GAM2 kurto;
PRINT;
print « A=> a;
print « B=> b;
print « C=> c;
print « D=> d;
print «*****»;

```

Estos parámetros son introducidos en (3) en la llamada a la subrutina, la cual devuelve los parámetros a b c y d. En (4) se generan muestras de n= 2000 y se le aplica la transformación de potencia deseada a partir de los valores devueltos por (3). Este bucle se repite diez veces con la intención de obtener la bondad de los parámetros a b c y d para el objetivo de asimetría y apuntamiento seleccionado. En (5), se presentan las medias de asimetría y apuntamiento obtenida de las diez muestras generadas.

### *Descripción del programa para generar distribuciones no normales multivariadas*

En (1), se declara el procedimiento FLEISHMAN, indicándose entre paréntesis los cinco parámetros necesarios para su ejecución: variable alfanumérica que contendrá el nombre del fichero de la matriz de correlaciones de las variables que se desean simular, tamaño muestral deseado, número de variables de la matriz de correlaciones, asimetría y apuntamiento deseados. Si quisiésemos obtener una muestra de 500 sujetos en 16 variables con 0.75 de asimetría y 1 de apuntamiento a partir de la matriz de las correlaciones de las 16 variables situada en el fichero de nombre SIGN\_1.MAT, llamaríamos al procedimiento de la siguiente forma:

```
c=fleis(«sign_1»,500,16,0.75,1);
```

En (2) mediante la sentencia LOCAL, se declaran las variables que sólo van a participar en este procedimiento. En (3) se determina el tipo de formato de presentación que se desea para los datos.

El bloque (4), a través de las variables a1, a2 y a3, indica el nombre del fichero que contiene la matriz de correlaciones. Dicho fichero, ha de poseer la extensión «MAT», y en la llamada a la subrutina FLEISHMAN se pondrá sólo el nombre entre comillas sin la extensión.

En (5) tiene lugar la carga de la matriz de correlaciones de orden pxp y la posterior comprobación de si se trata de una matriz triangular (los elementos de la diagonal superior, han de ser ceros) o completa.

En (6) se produce la carga de los vectores a b c y d correspondientes a las constantes multiplicativas de la transformación de potencia de Fleishman (1978) (Ec. I). Dichas constantes, han sido calculadas previamente mediante la subrutina NEW-

TON En el caso de que se desee la misma distribución para las p variables, los p elementos de a b c y d serán iguales.

El apartado (7) genera la matriz de transición en base al algoritmo de la Ecuación (VII) partiendo de la matriz de correlaciones original y de las constantes multiplicativas de Fleishman definidas por NEWTON.

```
(1) proc l=FLEISMAN (cor,n,v,as,k u);
(2) LOCAL de, check, x, i, j, y, va, ve,
   a1, a2, a3, f, rl, yt, to, yf,v2, za, z1a,
   corre, a, b, c, d, ASI1, KU1, asime,
   kurto,
   distri, SKEW,KURT, UMBRAL1,
   UMBRAL2;
(3) FORMAT /Rd 5.4;
   outwidth 132;
(4) a1=cor;
   a2=>.mat|,
   a3=cor $+ a2;
   de=eye(V);
(5) LOAD y[V,V]=^a3;
   check=y[1,v];
   if check eq 0;
      y=y+y'-de;
   else;
   endif;
(6) load a[v,1]=a.dat;
   load b[v,1]=b.dat;
   load c[v,1]=c.dat;
   load d[v,1]=d.dat;
(7) x=ones(v,v);
   i=1;
   do while i <= V;
      j=1;
      do while j <= V;
         if i==j;
            y[i,j]=1;
         elseif i > j;
            y[i,j]=0;
         else;
            x[i,j]=y[i,j]^(b[i]*b[j]
               +3*b[i]*d[j] +3*d[i]*b[j]
               +9*d[i]*d[j]);
            y[i,j]=x[i,j] +y[i,j] ^2*(2*
               c[i]* c[j]) +y[i,j]^3*(6*d[i]*d[j]);
```

```
      endif;
      j=j+1;
   endo;
   i=i+1;
endo;
```

En (9) se calculan los vectores y valores propios de la matriz de transición y en (10), se determina la matriz de carga factorial no rotada ve en base a:

$$A = V_p \sqrt{L} \quad (\text{VIII})$$

Donde A es la matriz de carga factorial que contiene las correlaciones de los componentes principales con las variables, Vp son los vectores propios y L los valores propios.

La etiqueta (11) r: indica el lugar de comienzo para el caso de que no se cumpla la condicional en (17).

En (12) se genera una matriz de datos f con distribución normal de media 0 y desviación típica 1 así como las matrices y vectores que se necesitan en (13) con valores iniciales nulos. En este bucle se generan los datos brutos a partir de las ecuaciones factoriales y de la matriz f generada en (11). Una vez calculados los datos brutos para cada variable, en (14) se le aplica la transformación de potencia (Ec. I) y en (15) se determina su asimetría y apuntamiento. En (16) se calcula la media de asimetría y apuntamiento de las p variables generadas y en (17) se comprueba que esta media no es superior o inferior en 0.2 a los momentos definidos en (1). Es importante destacar que las instrucciones (16) y (17) sólo tienen sentido si todas las variables poseen la misma asimetría y apuntamiento. En el caso de que las p variables, tuviesen distribuciones distintas, dichas líneas de programas habrían de ser eliminadas.

```

(8)   y=y+y'-de;
(9)   {va,ve}=eigrs2(y);
va=rev(va);
ve=(rev(ve'))';
(10)  ve=(ve' .* sqrt(va))';
(11)  r:
(12)  f=rndn(n.v);
r1=zeros(n,1);
yt=zeros(n,1);
to=zeros(n.v);
yf=zeros(n,v);
asime=zeros(v,1);
kurto=zeros(v,1);
(13)  i=1;
      DO WHILE i <= V;
      v2=ve[i,.];
      za=f.*v2;
      j=1;
      DO WHILE j <= V;
      z1a=za[.j];
      r1=r1+z1a;
      j=j+1;
      ENDO;
(14)  yt=a[i]+(b[i].*r1)+(c[i].*r1^2)
+(d[i].*r1^3);
      yf[.,i]=yt;
(15)  ASI1 = ( SUMC((yt - MEANC(yt))^3) / (ROWS(yt)* STDC(yt)^3));
      KU1 = (SUMC((yt - MEANC(yt))^4)/(ROWS(yt)*STDC(yt)^4))-3;
      asime[i]=ASI1;

```

```

kurto[i]=KU1;
r1=zeros(n,1);
yt=zeros(n,1);
ASI1=0;
KU1=0;
i=i+1;
ENDO;
(16) SKEW=meanc(asime);
UMBRAL1=abs(SKEW-AS);
KURT=meanc(kurto);
UMBRAL2=abs(KURT-KU);
(17) IF UMBRAL1 > 0.2;
      goto r;;
ENDIF;
IF UMBRAL2 > 0.2;
      goto r;;
ENDIF;
(18) corre=corrx(yf);
(19) retp(corre);
endp;
END;

```

En (18) se determina la matriz de correlaciones de los datos generados y transformados a la distribución deseada y en (19) el procedimiento devuelve esta matriz de correlaciones al programa que la solicitó, pudiendo ser utilizada posteriormente para ser analizadas por ejemplo en procedimientos de simulación que pretenden analizar la bondad de los procedimientos de análisis factorial tanto en su vertiente exploratoria como confirmatoria.

## Referencias

- Boomsma, A. (1983). On the robustness of LISREL (maximum likelihood estimation) against small sample size and non-normality. Unpublished PhD dissertation, University of Groningen, Groningen. The Netherlands.
- Babakus, E., Ferguson, C.E., & Joreskog, K.G. (1987). The sensitivity of Confirmatory Maximum likelihood factor analysis to vio-

- lations of measurement scale and distributional assumptions. *Journal of Marketing Research*, 24, 222-228.
- Chou, C.P., Bentler, P.M., & Satorra, A. (1991). Scaled test statistics and robust standard errors for non-normal data in covariance structure analysis: A Monte Carlo study. *British Journal of Mathematical and Statistical Psychology*, 44, 47-357.

- Fleishman, A.(1978). *A method for simulating non-normal distributions.* Psychometrika, 43, 4, 521-531.
- Gauss Sistem Version 3.0. Aptech Systems, Inc.
- Sharma, S., Durvasula, S., & Dillon, W.R. (1989). Some results on the behavior of alternate covariance structure estimation procedures in the presence on non-normal data. Journal of Marketing Research, 26, 214-221.
- Vale, D., & Maurelli, V. (1983). *Simulating multi ariate nonnormal distributions.* Psychometrika, 48, 3, 465-471.