TRABAJO DE INVESTIGACION PARA OBTENER EL GRADO DE

MAESTRIA EN ECONOMIA

EL COLEGIO DE MEXICO

CENTRO DE ESTUDIOS ECONOMICOS

EVALUACION DE PRUEBAS OMNIBUS DE NORMALIDAD MEDIANTE ESTUDIOS DE MONTE CARLO.

ANTONIO GARCIA CARREÑO

PROMOCION 1989-91

ENERO, 1992

ASESOR: Dr. Carlos Manuel Urzúa Macías

REVISOR: Dr. Angel Calderón Madrid

A mis padres

AGRADECIMIENTOS

- A mis padres y hermanos por su incondicional apoyo para la realización de mis estudios.

- A mis maestros de El Colegio de México, en especial a Carlos Urzúa, quién paciente y desinteresadamente me asesoró en este trabajo.

- A Laura Urdapilleta, quién con su compañía y apoyo me ayudó a sobrellevar la carga académica sin desistir.

- A mis compañeros y amigos, con quienes compartí momentos difíciles, pero también muy gratos, durante el programa de maestría.

- A todas aquellas personas que de una u otra manera contribuyeron positivamente para que lograra este objetivo.

RESUMEN

Mediante estudios de Monte Carlo, se evalúa el poder de tres pruebas ómnibus de normalidad: la LM de Bowman-Shenton-Jarque-Bera, la K2 de D'Agostino-Pearson, y una recientemente propuesta por Urzúa (1991) llamada aquí JBU.

La evaluación de las pruebas se realiza tanto sobre observaciones como sobre residuos, generando las primeras computacionalmente y estimando los segundos mediante mínimos cuadrados ordinarios en base a las observaciones generadas. Para el caso de observaciones se consideran 12 distribuciones de probabilidad, y valores de significancia calculados empíricamente para cada uno de los 19 tamaños de muestra; y 5 distribuciones para el caso de residuos, con valores de significancia tanto empíricos como asintóticos, también considerando 19 tamaños de muestra.

Los resultados del trabajo se resumen como sigue: i) el estadístico JBU es eficiente, teniendo mejor convergencia que el K2 y el LM, ii) el poder de cada prueba depende de la distribución de probabilidad sobre la cual se evalúa, iii) el poder relativo de las pruebas no se altera al evaluarse sobre residuos o sobre observaciones, iv) el poder absoluto de ellas es menor en el caso de residuos, confirmándonos la idea de supernormalidad de los errores de regresión, v) al considerar valores de significancia asintóticos, la prueba LM se comporta muy mal en tamaños de muestra pequeños, en comparación con la k2 y la JBU, ya que sobrestima la normalidad de la muestra y vi) de lo anterior, se sugiere la prueba JBU como una buena alternativa a la LM para muestras pequeñas, y a la K2 dado su mejor comportamiento asintótico y la facilidad de su cálculo.

INDICE

INTRODUCCION

INTRODUCCION

Las pruebas de normalidad son técnicas de inferencia estadística diseñadas para probar si la distribución de una muestra difiere de la normal o no. Existe una larga historia de estas pruebas, abarcando entre otras las basadas en el tercer y cuarto momento respecto al origen, $\sqrt{b1}$ y b2, la de análisis de varianza W de Shapiro-Francia, la K2 de D'Agostino-Pearson, y la LM de Bowman-Shenton-Jarque-Bera (buenas exposiciones de ellas aparecen en Pearson-D'Agostino-Bowman, 1977, White-Macdonald, 1980, y Jarque-Bera, 1981). Las pruebas W, K2 y LM tienen la característica de ser pruebas ómnibus, en el sentido de que son capaces de detectar desviaciones de normalidad debido tanto a asimetría como a kurtosis. Las pruebas $\sqrt{b1}$ y b2 tienen excelentes propiedades para detectar no-normalidad asociada con asimetría y con kurtosis, respectivamente.

En el presente trabajo se evalúa el poder de tres pruebas ómnibus, siendo dos de éstas ampliamente conocidas, la LM y la K2, y una tercera recientemente sugerida por Urzúa (1991), llamada aquí JBU, la cual es una modificación de la prueba LM que consiste en sustituir la estandarización del cuarto momento, por la estandarización del inverso del mismo cuarto momento. Esto se hace para acelerar la convergencia del estadístico. La razón de que no se estudien otras pruebas, tales como la W de Shapiro y Wilk, y la R de Pearson, D'Agostino y Bowman, es que Jarque y Bera (1981) han mostrado anteriormente que son inferiores a la prueba LM.

En este trabajo se evalúa el poder de las tres pruebas citadas tanto sobre observaciones como sobre residuos estimados, generando las primeras computacionalmente, y estimando los segundos mediante mínimos cuadrados ordinarios en base a las observaciones generadas. Para el caso de observaciones, se consideraron 12 distribuciones, y se evaluaron las pruebas con 19 tamaños de muestra, considerando valores de significancia calculados empíricamente para cada estadístico y para cada tamaño de muestra. Los mismos tamaños de muestra fueron tomados para la evaluación sobre residuos, sólo que en este caso se consideraron únicamente cinco distribuciones; se utilizaron valores de significancia tanto empíricos como asintóticos, esto último debido a que en aplicaciones econométricas prácticas los valores de tablas son los comunmente utilizados.

En el capítulo 2 se describen los tres estadísticos de prueba (a través del documento se usa indistintamente los términos "prueba" y "estadístico de prueba"). Luego en el capítulo 3 se exponen los principios generales de simulación computacional, así como los programas utilizados en la generación de variables seudoaleatorias con distribución de probabilidad específica. El capítulo 4 comenta los resultados del estudio de simulación, comparando el poder estimado de las distintas pruebas. En el capítulo 5 se resumen las conclusiones del trabajo. Para finalizar aparecen dos apéndices; el apéndice A presenta las tablas de resultados a las que se hace referencia en el texto, y el B incluye los programas usados en el presente trabajo.

DESCRIPCION DE LAS PRUEBAS UTILIZADAS

I.- DESCRIPCION DE LAS PRUEBAS UTILIZADAS

Los estadísticos muestrales de asimetría $\sqrt{b1}$ y kurtosis b2 se han usado en forma separada, junto con muchos otros como el D^* de D'Agostino, el estadístico R de Pearson-D'Agostino y el W de Shapiro-Wilk, para probar la hipótesis nula de normalidad de las observaciones, y muchos de ellos pueden ser modificados para probar normalidad de los errores de regresión (para un excelente resumen de estas pruebas ver White y Macdonald, 1980). A partir de la aparición del trabajo de Jarque y Bera (1981), la prueba de normalidad mediante el uso del estadístico LM, se ha convertido en la más popular, debido a su buen desempeño, así como a la facilidad de su cálculo. Extraña sin embargo, el poco interés mostrado por la prueba propuesta ya desde 1973 por D'Agostino y Pearson (D'Agostino y Pearson, 1973) los cuales sugieren una prueba ómnibus similar a la LM en que usa estandarizaciones del tercer y cuarto momento, cuya suma de cuadrados, llamada K2, se distribuye asintóticamente como una chi-cuadrada con 2 grados de libertad (por prueba ómnibus entendemos aquella capaz de detectar desviaciones de la distribución normal, debido tanto a asimetría como a kurtosis). Empíricamente se muestra en este trabajo las mejores propiedades de convergencia del estadístico K2 en comparación con el LM.

Los problemas de una lenta convergencia a la distribución chi-cuadrada con 2 grados de libertad del estadístico LM tiene su origen en los problemas que muestra la distribución del estadístico de kurtosis b2. En la tabla 1 se muestran los resultados de valores de significancia empíricos obtenidos con 10,000 repeticiones, tanto para $\sqrt{b1}$ como para b2, para distintos tamaños de muestra. De estos resultados, es fácil observar una marcada asimetría hacia la derecha de b2 para muestras chicas, no siendo ése el caso de $\sqrt{b1}$, lo cual nos corrobora la idea de que b2, aunque asintóticamente simétrico, no lo es para muestras paqueñas. Ello nos conduce a una sobrestimación de la significancia de valores altos y a una subestimación de la significancia de valores bajos, cuando en muestras chicas suponemos una distribución normal del estadístico b2.

Similarmente, la tabla 2 nos confirma la convergencia lenta y asimétrica de la estandarización de b2.

Muchos estudios se han realizado para determinar la distribución del cuarto momento estandarizado b2, y entre ellos sobresale el de Bowman y Shenton (1988), quienes sugieren un mejor comportamiento asintótico hacia la distribución normal del inverso de b2 (1/b2). Retomando esta idea, Urzúa(1991) propone una modificación al estadístico LM de Jarque-Bera-Bowman-Shenton, al sustituir la estandarización del b2 por la estandarización de 1/b2. Se muestra empíricamente una mas rápida convergencia de este estadístico (llamado en lo subsecuente JBU) en relación al LM convencional, y una convergencia más estable en relación al K2 de D'Agostino y Pearson. En la tabla 3 se observa este resultado de manera clara: concentrándonos en α=.1 y .05, los valores calculados empíricamente de los 3 estadísticos difieren considerablemente, sobre todo en tamaños de muestra chicos, siendo estos muy distantes del valor asintótico para el caso de LM, no así para el caso de los otros dos estadísticos, cuyos valores son desde muestras chicas, mucho mas cercanos al valor de tablas de una chi-cuadrada con 2 grados de libertad. Vale mencionar el comportamiento de los valores calculados de K2, los cuales para el caso de α=.1 son desde N=10 casi iguales al valor asintótico, y también para el caso de α=.05, solo que aquí la convergencia parece ser hacia abajo, con ciertos picos en N=35 y N=300. En general la mejor convergencia, considerando estabilidad y rapidez, es la del estadístico JBU.

Como se mencionó en la introducción, se consideraron tres pruebas ómnibus basadas en los estadísticos LM, K2 y JBU, todos ellos con distribución asintótica chi-cuadrada(2). Muchas pruebas no fueron incluídas ya que anteriormente se ha reportado que para una amplia gama de distribuciones, la prueba Jarque-Bera ha tenido un mejor comportamiento en la detección de no normalidad. A continuación se describen los tres estadísticos en los cuales se basan las pruebas utilizadas.

I.1.- ESTADISTICO LM DE BOWMAN-SHENTON-JARQUE-BERA

Suponiendo que las alternativas a la normal son otros miembros de la familia de Pearson, y mediante el uso del multiplicador de Lagrange, Jarque y Bera (1981,1987) derivan la expresión del estadístico LM (esta misma expresión es derivada por Urzúa (1989) para el caso cuando las alternativas son de máxima entropía). Esta expresión está dada por

$$LM = N[\sqrt{b1}/6+(b2-3)/24]+N[3(U_1)^2/2U_2-U_3U_1/(U_2)^2] \quad (1)$$

donde $U_i=\{(U_1)^i+...+(U_N)^i\}/N$.

Este estadístico se distribuye como una chi-cuadrada con 2 grados de libertad, siempre que la distribución de la muestra sobre la que se calcula sea la normal. La hipótesis nula de normalidad es rechazada cuando el valor calculado de LM excede el valor crítico de la distribución chi-cuadrada(2) para un α determinado. Para el caso de la evaluación empírica de esta prueba, se calcularon los valores de significancia de LM para distintos tamaños de muestra (tabla 8), dada la dificultad de obtener la distribución muestral del LM mediante métodos analíticos.

Suponiendo una distribución normal con media cero, la expresión (1) se reduce a

$$LM = N[(\sqrt{b1})^2/6+(b2-3)^2/24]$$

equivalente a la propuesta por Bowman-Shenton (1975).

I.2.- EL ESTADISTICO K2 DE D'AGOSTINO-PEARSON

En 1973, D'Agostino y Pearson proponen una prueba ómnibus para normalidad, mediante la previa transformación de $\sqrt{b1}$ y de b2 a nuevas variables $Z1(\sqrt{b1})$ y $Z2(b2)$, obteniendo el estadístico $K^2 = Z1^2 + Z2^2$, el cual tiene asintóticamente una distribución chi-cuadrada(2). La transformación de cada momento es la siguiente (D'Agostino et. al., 1990) (N representa el tamaño de muestra):

Para el caso de $\sqrt{b1}$

- Se calcula $\sqrt{b1}$ de la muestra, y luego se calcula
- $Y = \sqrt{b1}\{(N+1)(N+3)/6(N-2)\}^{1/2}$
- $\beta_2 = 3(N^2+27N-70)(N+1)(N+3)/(N-2)(N+5)(N+7)(N+9)$
- $W^2 = -1+\{2(\beta_2-1)\}^{1/2}$
- $\delta = 1/\sqrt{\ln(W)}$
- $\alpha = \{2/(W^2-1)\}^{1/2}$

Finalmente $Z(\sqrt{b1}) = \delta\ln(Y/\alpha+\{(Y/\alpha)^2+1\}^{1/2})$ es distribuída normalmente bajo la hipótesis nula de normalidad de la población.

Para el caso de b2:

- Se calcula b2 de la muestra, y luego se calcula
- $E(b2) = 3(N-1)/(N+1)$
- $var(b2) = 24N(N-2)(N-3)/(N+1)^2(N+3)(N+5)$
- $x = (b2-E(b2))/\sqrt{var(b2)}$
- $\beta_1=6(N^2-5N+2)\sqrt{(6(N+3)(N+5))}/((N+7)(N+9)\sqrt{(N(N-2)(N-3))})$
- $A = 6+(8/\beta_1)[2/\beta_1+\sqrt{(1+4/(\beta_1)^2)}]$

- $Z(b2)=((1-(2/9A))-[(1-2A)/(1+x\sqrt{(2/(A-4))}]^{1/3})/\sqrt{(2/9A)}$

Z(b2) se distribuye normalmente bajo la hipótesis nula de normalidad de la población.

La tabla 4 nos muestra las distribuciones empíricas de $Z(\sqrt{b1})$ y de Z(b2), siendo éstas bastante cercanas a la normal.

I.3.- ESTADISTICO JBU DE JARQUE-BERA-URZUA

Como se mencionó anteriormente, una de las principales razones de la lenta convergencia del estadístico LM es la fuerte asimetría en la distribución de la estandarización de b2, siendo ésta especialmente preocupante en muestras de tamaño inferior a 150. Esto puede llevarnos a una errónea utilización de este estadístico, cuando queremos detectar normalidad en residuos de regresiones económicas, donde el tamaño de muestra es por lo regular pequeño. Es en el afán de lograr una mejor convergencia de LM que Urzúa propone el siguiente estadístico:

$$JBU = N[(\sqrt{b1})^2/6+27(1/b2-1/3)^2/8]$$

que como se puede observar, es muy similar al LM que hemos trabajado, diferenciándose de él en que ahora se sustituyó b2 por su inverso. Para estandarizar dicho inverso se tuvo que calcular la varianza de 1/b2, la cual se obtuvo de la manera siguiente (Urzúa,1991):

Se expresa 1/b2 como $g(m_2,m_4) = (m_2)^2/m_4$, donde m_2 y m_4 son el segundo y cuarto momento respecto al origen. Posteriormente se obtienen las derivadas parciales con respecto al segundo y cuarto momento, definidas estas como g_1 y g_2, respectivamente

$$g_1 = 2m_2/m_4, \quad g_2 = -(m_2)^2/(m_4)^2$$

de esta manera la expresión de la varianza de 1/b2 es (ver Stuart y Ord, 1987 pp324 y Urzúa, 1991):

$$v(1/b2) = (2m_2/m_4)^2 v(m_2)+((-m_2)^2/(m_4)^2 v(m_4)+ 2(-2(m_2)^3/(m_4)^3 cov(m_2,m_4)+o(1/N)$$

donde el último termino de la derecha (orden de 1/N) tiende a cero cuando N crece. El desarrollo de esta expresión nos conduce a la siguiente expresión de la varianza:

$$v(1/b2) = 8/(27N)$$

la cual es usada para estandarizar el inverso de b2, pudiendo así establecer una nueva versión del LM de Bowman-Shenton-Jarque-Bera, llamado aquí JBU

$$JBU = N[(\sqrt{b1})^2/6+27(1/b2-1/3)^2/8)]$$

cuya convergencia a una chi-cuadrada(2) se ilustra en la tabla 3.

Teniendo definidas las tres pruebas a evaluar, se procedió a realizar los estudios de Monte Carlo, para de esta manera comparar la capacidad de rechazo de las mismas.

Antes de pasar a describir el poder de las pruebas, vale la pena describir las técnicas y los programas de simulación utilizados, lo cual se hace en el siguiente capítulo.

PRINCIPIOS Y METODOS GENERALES DE GENERACION DE NUMEROS ALEATORIOS

II.1.-GENERACION DE NUMEROS ALEATORIOS UNIFORMES

Cuatro métodos alternativos se han usado hasta ahora para generar secuencias de números aleatorios uniformes (en lo subsecuente me referiré a ellos como números aleatorios). El Manual, el de Tablas (Library Tables), el de Analogía Computacional (Analog Computer), y el método de Computadoras Digitales. Los dos primeros son lentos y más bien pedagógicos; el tercero consiste en la traducción de un proceso físico aleatorio a forma numérica, siendo la computadora el intérprete. Este último método genera verdaderos números aleatorios, pero no es capaz de reproducir una secuencia de dichos números.

El método de computadoras digitales tiene 3 variantes, dependiendo de la provisión de números a ser usados: provisión externa, generación interna mediante un proceso físico, y generación interna de secuencia de dígitos mediante una relación de recurrencia.

La provisión externa consiste en grabar tablas de número aleatorios en discos, y cargarlos a la máquina. La generación mediante un proceso físico es similar al de computadoras análogas y tiene la desventaja de que sus secuencias no son reproducibles. La tercera variante de este método es la más usual, y consiste en la generación de números aleatorios mediante transformaciones indefinidamente continuas de un grupo de números arbitrariamente elegidos. Esta variante supera los principales obstáculos de las anteriores, ya que con este no existen problemas de capacidad de memoria, y además el proceso y el resultado de él es completamente reproducible.

En general, el método más aceptable de generación de números aleatorios, deberá producir una secuencia de números tal que:

1.- Estén uniformemente distribuídos
2.- Sean estadísticamente independientes
3.- Se puedan reproducir
4.- No se repitan para un tamaño de muestra dado
5.- Se generen a alta velocidad
6.- Usen un mínimo de memoria.

El primer método para generar números seudoaleatorios en una computadora digital fue propuesto por Von Neumann y Metropolis en 1946, en el cual cada número en una secuencia se obtiene usando los dígitos del centro de la raíz cuadrada del número anterior de la secuencia. Sin embargo, este método fue abandonado por su lentitud en favor de los métodos de congruencia, los cuales, con algunas variantes, se usan en la actualidad.

Los métodos de congruencia para generar números aleatorios son determinísticos, debido a que el proceso aritmético usado en el cálculo determina de manera única cada término de la secuencia. En efecto, existen fórmulas con las cuales se puede calcular a priori el valor exacto del i-ésimo número en una secuencia, antes de que ésta sea generada. Estos métodos se basan en una relación fundamental de congruencia, la cual se expresa de la manera siguiente:

$$\alpha_{i+1} = a\alpha_i + c \pmod{m} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

donde α, a, c y m son todos enteros no negativos; mod=módulo.

Expandiendo la ecuación (1) tenemos

$$\alpha_1 = a\alpha_0 + c \pmod{m}$$
$$\alpha_2 = a\alpha_1 + c = a2\alpha_0 + (a+1)c \pmod{m}$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$
$$\alpha_i = ai\alpha_0 + (c(ai-1)/(a-1)) \pmod{m}$$

Así, dado un valor inicial α_0 y las constantes a y c, con la ecuación 2 obtenemos una relación de congruencia (con modulo m) para cualquier valor i sobre la secuencia {n1, n2,...,ni...}. De los enteros de la secuencia {ni}, se pueden obtener números racionales dentro del intervalo (0,1), formando la secuencia {ri}={ni/m}.

En la práctica se ha mostrado que no es posible obtener secuencias no repetidas mediante métodos de congruencia. Sin embargo, eligiendo un módulo suficientemente grande se puede generar una secuencia satisfactoria de acuerdo a los objetivos de cada trabajo.

II.2.- GENERACION DE NUMEROS ALEATORIOS CON DISTRIBUCIÒN ESPECIFICA

La teoría de generación de números aleatorios con cierta distribución de probabilidad, descansa en los siguientes dos supuestos:

Supuesto 1: Existe un generador perfecto de números aleatorios uniformes, es decir, un generador capaz de reproducir una secuencia U1,U2,...,Un de variables, con una distribución uniforme en (0,1). Habiendo hecho este supuesto, uno puede construir una buena teoría de generación de números aleatorios.

Supuesto 2: La computadora puede guardar y manipular números reales. Este supuesto junto al anterior nos garantiza la existencia de diversos generadores de las distintas distribuciones de probabilidad.

Lo que sigue es una descripción de los principios básicos usados en la generación de números aleatorios no uniformes; estos principios se aplican a menudo, pero no siempre, tanto a variables continuas como a variables discretas.

Se define $F(x)$ como la función de distribución acumulada de x, la cual denota la probabilidad de que una variable x tome un valor x ó menos. Cuando $F(x)$ es continua sobre el dominio de x, es posible diferenciarla y definir $f(x)=dF(x)/dx$. La derivada $f(x)$ es llamada la función de densidad de probabilidad. De esta manera $F(x)$ se puede expresar como

$$F(x) = p(X \leq x) = \int_{-\infty}^{x} f(t)dx \qquad (3)$$

donde $0 \leq F(x) \leq 1$ y $f(t)$ representa el valor de la función de densidad de probabilidad, de la variable aleatoria X cuando $X=t$.

Existen 3 principios básicos (métodos) para generar números aleatorios con distribución de probabilidad específica: el de inversión, el de rechazo y el de composición. Estos métodos o algunas variantes de ellos, son la base para la simulación de distribuciones del siguiente capítulo.

METODO DE INVERSION:

Si deseamos generar números aleatorios x's de una población estadística cuya función de densidad es dada por $f(x)$, obtenemos la función acumulada $F(x)$. Dado que $F(x)$ se define sobre el rango de 0 a 1, podemos generar números aleatorios U y establecer $F(x) = u$. Es claro que x es determinado de manera única por $u = F(x)$. Se sigue de ahí, que para cualquier valor particular u, digamos u_o, es posible encontrar valores de x, digamos x_o, el cual corresponde a u_o, mediante la función inversa de F. Esto es:

$$x_o = F^{-1}(u_o) \qquad (4)$$

donde $F^{-1}(u)$ es la transformación inversa de u en el intervalo unitario, a el dominio de x. $F^{-1}(u)$ es una variable con función de densidad f(x). De esta manera, la variable x resultante tendrá función de distribución f(x).

Podemos así calcular x tan rápido como sea posible calcular la inversa de la función. Desafortunadamente, muchas veces es difícil expresar x en términos de la inversa $F^{-1}(u)$, en cuyo caso se puede obtener una aproximación a F^{-1}, ó usar alguno de los métodos que se describen enseguida.

METODO DE RECHAZO:

La condición para poder aplicar este método es que f(x) sea acotada, y que x tenga un rango finito, digamos $a \leq x \leq b$. El método consiste en primero normalizar el rango f mediante un escalar c, tal que $c.f(x) \leq 1$, $a \leq x \leq b$. Posteriormente se define x como una función lineal de u, $x = a+(b-a)u$. A continuación se generan pares de números aleatorios uniformes (u1, u2). Finalmente, cuando encontramos un par (u1, u2) tal que $u \leq c.f.(a+(b-a)u1)$, se aceptará el par y se usará $x = a+(b-a)u1$, como la variable aleatoria cuya densidad de probabilidad es f(x).

Se ha encontrado que el número esperado de intentos antes de aceptar un par (u1, u2) es igual a 1/c, lo cual implica que el método puede ser ineficiente para ciertas funciones de densidad de probabilidad.

METODO DE COMPOSICION (O DE DESCOMPOSICION EN COMBINACIONES DISCRETAS):

Si nuestra densidad objetivo f(x) se puede descomponer en combinaciones discretas, entonces:

$$f(x) = \Sigma_i P_i f_i(x) \qquad (5)$$

donde f_i's son funciones de densidad elegidas por criterios de "mejor ajuste", y que además minimizan la sumatoria $\Sigma T_i.P_i$; T_i es el tiempo de computación en la generación de números aleatorios de la función f_i; P_i's constituye un vector de probabilidad tal que $\Sigma P_i=1$.

METODO DE ACEPTACION Y COMPLEMENTO:

El método consiste en lo siguiente: sea f(x) una función de densidad la cual se puede descomponer en la suma de dos funciones no negativas $f(x) = f1(x)+f2(x)$. Suponga más aún que existe una función de densidad g(x) "fácil" tal que $f1(x) \leq g(x)$. Entonces el siguiente algoritmo se usa para generar x con densidad f(x), suponiendo que tenemos las funciones f1(x), f2(x) y g(x), además de números aleatorios U(0,1):

Si

U> f1(x)/g(x), entonces genere los números aleatorios x

con densidad $f2(x)/\int f2(x) = f(x)$.

Hasta aquí se han descrito los principios básicos utilizados en la elaboración de algoritmos para la generación de números aleatorios con distribución específica. Algunos de ellos se exponen a continuación.

II.3.- PROGRAMAS DE GENERACION DE DENSIDADES CONTINUAS

En la presente sección se plantean los algoritmos de generación de números aleatorios de 10 distribuciones continuas. La mayoría de los programas de generación toman como base los números aleatorios uniformes y normales estándar, los cuales ya están programados en GAUSS, el lenguaje que se usó para las simulaciones o experimentos de Monte Carlo.

Dado que el objetivo del trabajo es evaluar el poder de distintas pruebas de normalidad, se consideraron aquellas funciones de densidad que, con determinados valores de los parámetros, tienen cierta semejanza con la función normal. Además se evaluó la normal misma. Las distribuciones son las siguientes:

1.- Distribución Gamma(2,1)
2.- Distribución Beta(3,2)
3.- Distribución Beta(2,2)
4.- Distribución chi-cuadrada
5.- Distribución t-student
6.- Distribución F
7.- Distribución Cauchy
8.- Distribución Tukey
9.- Distribución "Mezcla de dos Normales"
10.- Distribución Lognormal
11.- Distribución Exponencial Cuártica
12.- Distribución Normal

A continuación describiré de manera breve los programas usados como generadores de las distintas distribuciones, así como algunas características de las distribuciones mismas. En todos los programas se incluye la letra s (semilla) cuando se generan números uniformes (rndus), así como cuando se generan números normales (rndns); ésto con el fin de generar la misma secuencia de números cuando sea de nuestro interés, por ejemplo evaluar normalidad de una misma secuencia de números, utilizando distintas pruebas de normalidad.

DISTRIBUCION GAMMA

Si un proceso consiste de k eventos sucesivos, y si el tiempo total del proceso se puede considerar como la suma de k variables exponenciales independientes, cada una con parámetro α, entonces la distribución de probabilidad de esta suma será la gamma, con parámetros k y α, esto es, gamma(k,α). Su función de densidad se expresa

$$f(x) = \frac{\alpha k x (k-1) e^{-\alpha x}}{(k-1)!}$$

Aquí k>0 es el parámetro de forma, y α>0 es el parámetro de escala. Cuando k es grande, la forma de la distribución se asemeja a una normal. Por ello, en la simulación se usaran valores de k grandes, digamos 20.

Para la generación de variables de una distribución gamma se supuso ésta como la suma de k variables exponenciales xi, i= 1,2,...,k, con idéntico valor esperado 1/α. Además se utilizó el hecho de que una variable exponencial exp, se genera a partir de una variable uniforme u como exp = (1/α)*ln(u).

El programa que se usó en Gauss es el siguiente:

```
proc rndgs(k,a,n,s);
  local g,u;
  u=rndus(n,k,s);
  g=-ln(prodc(u'))/a;
  retp(g);
endp;
```

donde n significa el número de variables gamma generadas, "a" equivale a α, y el resto se ha definido ya.

DISTRIBUCION BETA

La distribución beta es el cociente de dos variables gamma x1 y (x1+x2), donde x1 y x2 son variables gamma independientes con el mismo parámetro α y distinto parámetro k, k1 y k2. Así, el parámetro del numerador será k1 y el del denominador k = k1+k2. La variable beta es dada por:

$$x = \frac{x1}{(x1+x2)} \qquad 0 < x < 1$$

con una función de densidad

$$f(x) = \frac{\Gamma(k1 + k2)xk1\text{-}1(1\text{-}x)k2\text{-}1}{\Gamma(k1)\Gamma(k2)}$$

Existen diversos métodos de generación, siendo el más rápido y común el del cociente de dos gammas:

```
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'))
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

donde la variable x tiene una distribución beta(k1,k2), con k1, k2 > 1 (para la evaluación de las pruebas de normalidad, la distribución beta fue generada con parámetros (2,1) y (2,2), la primera para poder comparar los resultados con los de Jarque y Bera(1981), y la segunda para obtener una distribución simétrica mas parecida a la normal que la primera).

DISTRIBUCION CHI-CUADRADA

Esta distribución es un caso particular de la función de distribución gamma, con $k = 1$ y $\alpha = 1/2$. Otra manera alternativa de ver a esta distribución, es como la sumatoria de los cuadrados de n variables aleatorias normales estándar, siendo n-1 los grados de libertad de la función chi-cuadrada resultante.

La densidad de una chi-cuadrada(m) es la de una gamma con $k=m/2$ y $\alpha=1/2$:

$$f(x) = \frac{x^{m/2-1}\, e^{-x/2}}{2^{m/2}\,(m/2-1)!}$$

El método de generación fue en base a la suma de cuadrados de n variables normales:

```
proc rndchs(m,n,s);
  local ch;
  ch=sumc(rndns(m,n,s)^2);
  retp(ch);
endp;
```

DISTRIBUCION T-STUDENT

Esta distribución resulta de dividir una variable aleatoria normal estándar, sobre la raíz cuadrada de una chi-cuadrada previamente dividida por sus grados de libertad. Esto es, si z tiene una distribución normal estándar, y u tiene una distribución chi-cuadrada con k grados de libertad, y si además ambas son independientes, entonces

$$x = \frac{z}{\sqrt{(u/k)}}$$

tiene una distribución t-student con k grados de libertad. Su función de densidad se define como:

$$f(x) = \frac{\Gamma[(k+1)/2]}{\Gamma(k/2)\surd(k\pi)\ (1+x2/k)(k+1)/2}$$

Como caso especial de esta distribución tenemos a la Cauchy, la cual resulta cuando k = 1, y a la función F, la cual resulta de la raíz cuadrada de una variable t. Cuando k es mayor que 30, se usa directamente la distribución normal; por ello para el caso de nuestro trabajo consideraremos un valor de k de 5, con la idea de comparar los resultados con los de Jarque y Bera (1981).

Esta distribución es de las de mayor semejanza con la normal estándar, ya que también es simétrica con respecto a cero, y tiene por lo tanto el mismo valor esperado que la normal. Solo varía en una mayor varianza y en una menor kurtosis, diferencias que se atenúan cuando k crece.

Una vez que se han generado variables con distribución chi-cuadrada, y teniendo las normales estándar dentro de las subrutinas de Gauss, el algoritmo de generación resulta sencillo:

```
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(n,m+1,s);
  x=u[.,1];
  y=u[.,2:(m+1)];
  t=x./sqrt(sumc((y^2)')/m);
  retp(t);
endp;
```

DISTRIBUCION F

La distribución F resulta del cociente de dos variables chi-cuadradas independientes, dividida cada una por sus grados de libertad. Esto es, sí V y U son dos variables independientes con distribución chi-cuadrada, con m y n grados de libertad respectivamente, entonces la variable x, definida como

$$x = \frac{V/m}{U/n}$$

tiene una distribución F con m grados de libertad en el numerador y n en el denominador, o sea, F(m,n).

Su función de densidad se expresa

$$f(x) = \frac{\Gamma[(m+n)/2]}{\Gamma(m/2)\Gamma(n/2)}(m/n)m/2 \frac{x(m-2)/2}{[1+(m/n)x](m+n)/2}$$

Esta distribución se puede generar a partir de variables normales (el cociente del cuadrado de ellas, dividida cada una por sus grados de libertad). El programa con el cual se generaron las variables es el siguiente:

```
proc rndfs(m,k,n,s);
  local rn,x,y,f;
  rn=rndns(m+k,n,s);
  x=sumc(rn[1:m,.]^2)/m;
  y=sumc(rn[m+1:m+k,.]^2)/k;
  f=x./y;
  retp(f);
endp;
```

DISTRIBUCION CAUCHY

La función de densidad Cauchy

$$f(x) = \frac{1}{\pi(1+x^2)}$$

es otra de las funciones con las cuales evaluaremos las pruebas de normalidad. Esta distribución no tiene parámetros, y su media no existe. Para generar variables Cauchy el método de inversión parecería el más adecuado, ya que su función de distribución es

$$F(x) = 1/2 + 1/\pi(\arc \tan x)$$

lo cual nos sugiere un generador como $\tan(\pi u)$, donde u es una variable uniforme. Sin embargo este método se puede agilizar y generar las variables cauchy mediante el cociente de 2 normales estándar, lo cual nos conduce al siguiente algoritmo:

```
proc rndcs(n,s);
  local c,x;
  x=rndns(n,2,s);
  c=x[.,1] ./ x[.,2];
  retp(c);
endp;
```

DISTRIBUCION TUKEY

Esta distribución es muy importante considerarla, ya que con ciertos valores de los parámetros, se asemeja mucho a la normal. La función de distribución original es

$$F^{-1}(U) = 1/\sigma \ (U\sigma-(1-U)\sigma)$$

donde σ es el parámetro de forma y U una variable uniforme[0,1]. Esta distribución ha sido generalizada de manera tal que se incluye un parámetro de escala. Esta nueva distribución se define

$$F^{-1}(U) = \sigma 1+1/\sigma 2(U^{\sigma 3}-(1-U)^{\sigma 4})$$

donde $\sigma 1$ es el parámetro de localización y $\sigma 2$ es el parámetro de escala. El mérito de esta familia es su versatilidad en la modelación de datos, y su fácil generación. Las formas más comunes que puede tomar esta distribución son:

- Cuando $\sigma 3=\sigma 4=0$, la densidad de la distribución tiende a una logística.
- Cuando $\sigma 1=\sigma 3$ y $\sigma 2=\sigma 4$ -->0, la función de densidad tiende a una densidad exponencial.
- Cuando $\sigma 1=0,\sigma_2=.1975,\sigma_3=\sigma_4=.1349$, la función así obtenida difiere de la normal en solo .002.

La función de densidad no es conocida en su forma cerrada.

El programa de generación usado es el siguiente:

```
proc rndtuks(l1,l2,l3,l4,n,s);
  local u,z;
  u=rndus(n,1,s);
  z=l1+(1/l2)*((U)^l3-(1-U)^l4);
  retp(z);
endp;
```

donde se a sustituido la letra σ por la letra l, y donde, como siempre, s significa la semilla y n el tamaño de la muestra generada.

Tratando de obtener una distribución Tukey casi normal, los parámetros considerados fueron los dados arriba.

DISTRIBUCION "MEZCLA DE DOS NORMALES"

Este tipo de distribución resulta de una combinación lineal de dos variables aleatorias normales. Esta distribución tendrá dos modas, cada uno de ellas correspondiendo al valor de la media de cada normal. El programa de generación resulta sencillo, toda vez que se tiene incorporado dentro de GAUSS la rutina de generación de las variables normales.

A el programa le tendremos que dar las medias m_i de cada variable, así como sus desviaciones estándar s_i, para poder así, en base a la normal estándar que nos da GAUSS, generar las dos variables aleatorias normales.

El programa de generación es el siguiente:

```
proc rndmns(c,m1,m2,s1,s2,n,s);
  local z,mn;
  z=rndns(n,2,s);
  mn=c*(s1*z[.,1]+m1)+(1-c)*(s2*z[.,2]+m2);
  retp(mn);
endp;
```

DISTRIBUCION LOGNORMAL

Si el logaritmo de una variable aleatoria positiva tiene una distribución normal, entonces esa variable aleatoria tiene una distribución continua con asimetría positiva, conocida como distribución lognormal. Esta distribución se usa frecuentemente para describir procesos aleatorios productos de muchos eventos pequeños e independientes. Esta propiedad de la lognormal es conocida como la ley de los efectos proporcionales y provee la base sobre la cual se pude asumir que un proceso tiene distribución lognormal.

La función de densidad de una variable lognormal x, con media μ y varianza $\sigma 2$ es:

$$f(x;\mu,\sigma^2) = 1/x\surd\pi\sigma \exp[-1/2\sigma^2(\ln x-\mu)^2]$$

Para elaborar el programa generador de las variables aleatorias lognormales x, se utilizó el hecho de que tanto la media como la varianza de las variable y = ln(x), se pueden expresar en términos de los mismos momentos de la variable x. O sea que para generar variables lognormales, necesitamos darle al programa su media y su varianza. El programa es el siguiente:

```
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

donde a y b representen la media y la varianza de la variable lognormal a ser generada.

DISTRIBUCION EXPONENCIAL CUARTICA

Esta distribución es un caso especial de la familia Q-exponencial (Urzúa, 1988), la cual incluye como caso particular a la normal. La distribución exponencial cuártica es de la forma

$$f(x) = \theta(\alpha)\exp(-Q(x))$$

donde $\theta(\alpha)$ es la constante de normalización, y $Q(x)$ un polinomio de orden cuártico.

La distribución exponencial cuártica con $Q(x)=\exp(x^4/4)$ tiene gran semejanza a la normal, por lo cual resultó interesante evaluar el comportamiento de las pruebas ante este tipo de distribución, que de hecho junto con la Tukey, fue donde las pruebas mostraron menor poder. El programa de generación de variables aleatorias con distribución exponencial cuártica se basó en el método de rechazo, y es el siguiente:

```
proc rndecs(n,s);
  local i,j,x,ut,ec,s1,s2;
  i=1;
```

```
  j=0;
  ec=zeros(n,1);
  do until i>n;
    s1=s+j;
    s2=s1+1;
    x=rndns(1,1,s1);
    ut=exp((x^4)/4)*rndus(1,1,s2);
    if ut<=1;
      ec[i,1]=x;
      i=i+1;
    endif;
    j=j+1;
  endo;
  retp(ec);
endp;
```

DISTRIBUCION NORMAL

La distribución normal es la de uso mas frecuente en estadística. Esto debido a que teórica y empíricamente se ha mostrado que bajo ciertas condiciones se justifica suponer distribuciones normales. La utilidad de esta distribución se deriva del Teorema del Límite Central, el cual establece que la distribución de probabilidad de la suma de N variables aleatorias independientes e idénticamente distribuídas, se aproxima a una distribución normal.

Se dice entonces que una variable X es normalmente distribuída si su densidad es

$$f(x) = (1/\sqrt{2\pi})e^{-x^2/2}$$

Cuando X es normalmente distribuída, entonces $\mu+\sigma X$ se dice es normal(μ,σ). Por ello es que para fines de generación de variables μ y σ^2 (media y varianza) son irrelevantes.

El método de generación presentado en GAUSS es el de composición y rechazo, propuesto en el trabajo de Kinderman y Ramage, al cual referimos (Ver Devroye, 1986 pp 382).

RESULTADO DE LOS EXPERIMENTOS DE MONTE CARLO

III.- RESULTADOS DE LOS EXPERIMENTOS DE MONTE CARLO

En esta capítulo se presentan los resultados de los experimentos de Monte Carlo, realizados para comparar el poder estimado de las pruebas de normalidad basadas en los estadísticos LM, K2 Y JBU (descritos con anterioridad en el capítulo 3). El ejercicio cubrió no solo observaciones, sino también residuos de regresión obtenidos usando cuadrados mínimos ordinarios CMO.

La evaluación del poder de las tres pruebas se hizo utilizando 19 tamaños de muestra, comprendidos entre N = 10 y N = 800. Las pruebas fueron evaluadas, para el caso de observaciones, sobre 12 distribuciones, donde los valores de significancia de cada estadístico se calcularon empíricamente para cada tamaño de muestra, y donde siempre se usó un mismo nivel de significancia de 10%. Las distribuciones consideradas fueron las descritas en el capítulo anterior.

Bajo la hipótesis nula de normalidad H_O, el poder estimado de cada prueba se obtuvo dividiendo el número de veces que H_O fue rechazada, sobre el número de veces que se repitió el experimento, en nuestro caso 10,000.

Para el caso de residuos, las pruebas fueron evaluadas solo sobre las distribuciones citadas en el trabajo de Jarque y Bera(1981) y ellas fueron: la normal, gamma(2,1), beta(3,2), t-student(5), y la distribución lognormal. Para la estimación de los residuos se utilizó una matriz de diseño (N,3), cuya primera columna fueron unos, y las dos restantes números extraídos de una distribución uniforme u(0,1). Al igual que en el caso de las observaciones, los valores de significancia de cada estadístico fueron calculados empíricamente y sobre ellos fue evaluado el poder de cada prueba. Para el caso de residuos, las pruebas también se evaluaron tomando el valor de significancia asintótico de cada estadístico, el cual coincide con el de una chi-cuadrada(2), que para α=.1 tiene un valor de 4.61.

Debe también notarse que para el cálculo tanto de los valores críticos como de las estimaciones de poder de cada prueba, se realizaron 10000 repeticiones para cada tamaño de muestra; los programas fueron ejecutados en una computadora ACER 80386, con un coprocesador 80387, usando GAUSS, y una velocidad de 33MHZ.

A continuación se comentan los resultados obtenidos, los cuales se resumen en las tablas 5, 6, 7 y 8 del apéndice A.

III.1.- PRUEBAS DE NORMALIDAD DE OBSERVACIONES

En la tabla 5, bajo el título de "poder estimado sobre observaciones", se presentan las doce distribuciones trabajadas, con especificación tanto del tamaño de muestra N, como de los tres estadísticos de prueba considerados. Cada valor presentado significa la proporción de veces que se rechazó H_O=normalidad, en 10,000 repeticiones, en base a el valor de significancia calculado (ver tabla 8, y los programas 1-36 del apéndice); esto para cada distribución, tamaño de muestra y estadístico considerado. En todas las distribuciones no normales, la mejor prueba fue aquella que rechazó el mayor número de veces la hipótesis nula de normalidad (es decir, la de mayor poder estimado), excepto por supuesto cuando consideramos la distribución normal. En todos los casos, aparece subrayado el valor que corresponde a aquella prueba que mostró ser mejor, excepto cuando dos o más pruebas igualan.

En el caso de las dos distribuciones beta, es evidente la superioridad de la prueba JBU, mostrando el mayor poder para todos los tamaños de muestra. En el caso de la distribución cauchy, la LM es la prueba preferida, al igual que con la distribución chi-cuadrada(2), gamma(2,1) y la distribución $F_{(3,4)}$; siguiendo la misma tabla vemos nuevamente una superioridad clara de la prueba JBU cuando se evalúa sobre la distribución exponencial cuártica, no importando para ello el tamaño de la muestra. Evaluando la lognormal, vuelve a ser preferida la prueba LM, aunque se observa una convergencia mayor de la JBU hacia el rechazo completo de esta distribución. Cuando se consideran las últimas cuatro distribuciones, el predominio de alguna prueba sobre las otras ya no es tan claro; así por ejemplo, con la mezcla de dos normales la prueba LM se desempeña mejor en la detección de no normalidad en muestras pequeñas hasta de N=40, pero en muestras mayores es superada por la prueba JBU. Por otro lado, con la distribución normal es difícil decidir entre las pruebas K2 y JBU, sucediendo algo muy similar en el caso de la distribución Tukey. Es en la t(5) donde la prueba LM vuelve a ser preferida sobre las demás para todos los tamaños de muestra.

De lo descrito hasta aquí, se observa un mejor desempeño del estadístico de prueba JBU cuando se evalúa en distribuciones similares a la normal (beta(2,2), exponencial cuártica, tukey, beta(3,2), y la misma normal), siendo superado por el LM en distribuciones mas disímiles de la normal ($F_{(3,4)}$, gamma(2,1), lognormal).

III.2.- PRUEBAS DE NORMALIDAD DE RESIDUOS DE REGRESION

En la tabla 6, aparecen los resultados del poder estimado para cada prueba, siendo éste calculado sobre los residuos de regresión usando CMO. El poder se calcula de manera semejante a la descrita en la sección anterior, sólo que ahora se sustituyen las observaciones, por los residuos estimados en base a la expresión

$$\hat{u}=(\mathbf{I}-\mathbf{V})u,$$

la cual resulta después de estimar los coeficientes de la regresión usando mínimos cuadrados ordinarios.

$\hat{u}$, el vector estimado de residuos, es una transformación lineal de los residuos no observados u, que en este caso son vectores con distribución dada generados por simulación. La matriz de transformación $\mathbf{V}$ se calcula en base a la matriz de diseño $\mathbf{X}$ (ya definida al inicio del capítulo) siendo su expresión

$$\mathbf{V} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'.$$

En las 5 distribuciones consideradas se observa un mayor poder de la prueba K2 para N=10, lo cual se explica en parte por el valor de significancia calculado para este tamaño de muestra, 3.52, muy superior al mismo valor calculado para LM y JBU, 1.43 y 2.85, respectivamente (tabla 8). Sin considerar N=10, con la distribución beta(3,2) la prueba JBU es preferida, si bien la superioridad de esta prueba sobre las demás no es tan grande. Aplicando las pruebas sobre residuos con distribución gamma(2,1) y lognormal, la LM es preferida para tamaños de muestra menores a 65, siendo superada a partir de aquí por la prueba JBU; para el caso de la distribución t-student(5), nuevamente la prueba LM es la mejor, superada solo en una ocasión por la K2 cuando N=10. En la distribución normal no hay aparentemente ninguna prueba que domine sobre las otras, siendo quizá la JBU la mas aceptable.

La comparación de las tablas 5 y 6 nos arroja un resultado interesante, y es la confirmación de la tendencia de los residuos a ser más normales. Esto se manifiesta en que, en casi todos los casos, las pruebas evaluadas sobre residuos nos dan un menor poder estimado que las mismas pruebas aplicadas a las observaciones, tendiendo a igualarse conforme el tamaño de la muestra N crece. Algunos estadísticos como R. Gnanadesikan (ver Bera y McKenzie 1986,pp 45) llaman a este resultado "supernormalidad" de los residuos de regresión.

En general, el poder relativo de las tres pruebas es el mismo, es decir, la jerarquización de las pruebas no varía sustancialmente de cuando consideramos observaciones a cuando consideramos residuos.

Un ejercicio más que se pensó realizar fue el modificar la matriz de diseño, tanto en el número de variables independientes como en la naturaleza de las mismas. Sin embargo se desistió de este intento al consultar los resultados de Jarque y Bera (1981), los cuales demuestran que, si bien los valores estimados del poder de diferentes pruebas varían al cambiar la matriz de diseño, el poder relativo de las mismas no se altera.

III.3.- PRUEBAS DE NORMALIDAD DE RESIDUOS, CONSIDERANDO VALORES DE SIGNIFICANCIA ASINTOTICOS

Los resultados de las dos secciones anteriores, fueron obtenidos al considerar los valores de significancia (α=10%) calculados empíricamente para cada estadístico de prueba. En la práctica, las pruebas de normalidad se aplican considerando los valores asintóticos, para el nivel de significancia deseado y usando las tablas de aquella distribución a la que asintóticamente converge el estadístico de prueba en cuestión. Esto significa que en nuestro caso usaríamos las tablas de la chi-cuadrada, buscando los valores correspondientes a 2 grados de libertad. Como para α=.1 el valor crítico de la distribución chi-cuadrada(2) es 4.61, es en base a este valor de significancia que se obtuvieron los resultados de la tabla 8.

Aquí cambian considerablemente los resultados. Evaluados sobre residuos con distribución beta(3,2), la prueba que domina es la K2, seguida por la JBU; para el caso de la gamma(2,1) y la lognormal sucede lo mismo, siendo nuevamente mejor la prueba K2, seguida de cerca por la prueba JBU. Cuando se considera la distribución normal, el estadístico LM muestra ser el mejor (obsérvese que aquí la prueba con menor poder resulta ser la mejor), en todos los tamaños de muestra; esto es porque LM esta sesgada en favor de normalidad cuando se usa el valor asintótico. Por último, con la t-student(5) sucede que para muestras relativamente chicas, hasta de 80, la prueba preferida es la K2, siendo superada por la LM en tamaños de muestra mayores e iguales a 100.

CONCLUSIONES

CONCLUSIONES

Se ha evaluado el poder de las pruebas LM, K2 Y JBU en distintos tamaños de muestra, y sobre distintas distribuciones de probabilidad. La prueba basada en el estadístico LM modificado, o sea la JBU, se ha seguido con especial interés, ya que nunca antes su poder había sido evaluado en relación a las demás pruebas. Un primer resultado fue que el estadístico JBU es eficiente, teniendo de hecho una convergencia más rápida a la distribución chi-cuadrada(2), que el estadístico LM, y una convergencia mas estable que el estadístico K2. Lo anterior, sumado a la mayor facilidad en el cálculo del estadístico JBU, si se le compara por ejemplo con el cálculo de K2, nos hacen pensar en la posibilidad de usar la prueba JBU para el caso de tamaños de muestra relativamente chicos.

La estimación del poder de cada prueba sobre muestras de observaciones, considerando los valores de significancia calculados empíricamente, nos condujo a un segundo resultado, y es que el poder de cada prueba depende de la distribución de probabilidad de la muestra sobre la cual se evalúan dichas pruebas. Esto se refleja en el hecho de que para distribuciones muy parecidas a la normal, la prueba preferida fue la JBU, no así para otras distribuciones, digamos mas "lejanas" a la normal, donde la prueba LM mostró mejor desempeño.

La evaluación de las pruebas (modificadas) sobre residuos estimados en base al uso de cuadrados mínimos ordinarios CMO, nos arrojó un tercer resultado. Este fue que el poder relativo de las pruebas no se altera, resultando ser igual al caso de considerar muestras de observaciones. Sin embargo, al considerar el poder absoluto, se observa que éstos en general son menores para el caso de residuos, lo cual nos confirma la idea de que los residuos tienden a ser más normales que las observaciones.

Las mismas muestras de residuos se consideraron para estimar el poder de las pruebas, sólo que ahora considerando el valor de significancia asintótico de los tres estadísticos. Aquí el resultado fue interesante, siendo éste que para muestras chicas, la prueba LM se comporta muy mal, en comparación con las pruebas K2 y JBU, explicado esto claramente por la lenta convergencia que muestra el estadístico LM. Lo anterior nos lleva al resultado de que el uso de la prueba LM para detectar no-normalidad de errores de CMO, en muestras digamos inferiores a 100, sobrestima la normalidad de los mismos, recomendándose en estos casos el uso de las pruebas JBU y K2, ambas con un poder muy superior a la LM en muestras chicas. De estos dos últimos, el mejor comportamiento asintótico, así como la facilidad de su cálculo, nos sugiere a la prueba JBU como una buena alternativa de la LM para muestras de tamaño pequeño.

BIBLIOGRAFIA

BIBLIOGRAFIA

- ANSCOMBE F.J. Y GLYNN W.J. (1983).
Distribution of the kurtosis statistic b2 for norma samples. Biometrika 70,1,pp 227'34.

-BERA A. K. Y JARQUE C. M.(1981)
Efficient test for normality, homoscedasticity and serial independence of regression residual. Monte Carlo evidence. Economic Letters. 7, 4, pp. 313-318.

-BERA A. K. Y JARQUE C. M.(1981)
An efficient large-sample test for normality of observations and regression residuals. The Australian National University, Working paper No. 040.

- BERA A. Y JOHN S. (1983).
Test for multivariate normality with Pearson alternatives. Communication in Statistics. Theory and Methods. 12(1),pp 103-117.

- BERA A. K. Y MC KENZIE C. R. (1986).
Test for normality with stable alternatives. Journal of Statistical Computation and Simmulation, Vol. 25, pp 37-52.

- BOWMAN K.O. Y SHENTON L.R. (1975)
Omnibus test contours for departures from normality based on √b1 y b2. Biometrika, 62,2,pp 243-250.

- BOWMAN K.O. Y SHENTON L.R. (1988)
Properties of Estimators for the Gamma Distribution. Marcel Dekker, New York.

- D'AGOSTINO R.B. (1978)
Transformation to normality of the null distribution of g1. Biometrika, 57, 3, pp 679-681.

- D'AGOSTINO R.B. Y PEARSON E.S. (1973)
Test for departure from normality. Empirical results for the distributions of b2 and b1. Biometrika, 60, 3, pp 613-622.

- D'AGOSTINO R.B., BELANGER A. Y D'AGOSTINO R.B.(Jr.) (1990).
A Suggestion for using powerful and informative test of normality. *The American Statistician*, 44, 4, pp 316-321.

- DEVROYE LUC NON (1986).
Uniform Random Variate Generation, Springer-Verlag, New York.

- JARQUE C. M. Y BERA A. K. (1980)
Efficient test for normality, homoscedasticity and serial independence of regression residual. *Economics Letters*, 6, 3, pp. 225-259.

- JARQUE C. M. Y BERA A. K. (1987)
A test for normality of observations and regression residuals. *International Statistical Review*, 55,2, pp 163-172.

- JOINER B. L. Y ROSENBLATT J. R. (1971).
Some properties of the range in samples from Tukey's symmetric lambda distribution. *Journal of the American Statistical Association*, 66, 334, pp 394-399.

- JUDGE G. et. al. (1988)
Introduction to the Theory and Practice of Econometrics. John Wiley & Sons. New York.

- MOOD A. M., GRAYBILL F. A., BOES D. C. (1974).
Introduction to the theory of statistics. McGraw Hill.

-NAYLOR R.H., BALINTFY J.L., BURDICK D.S., CHU K. (1966).
Computer Simulation Techniques. John Wiley & Sons, New York.

- PEARSON E. S., D'AGOSTINO R. B. Y BOWMAN K. O. (1977).
Test for departure from normality; comparison of powers. *Biometrika*, 64, 2, pp 231-246.

- SHAPIRO S. S. Y FRANCIA R. S. (1972).
An approximate analysis of variance test normality. *Journal of the American Statistical Association*. 67,337, pp 215-216.

- STUART A. Y ORD J. K. (1987).
Kendall's Advanced Theory of Statistics vol. 1, Fifth edition. Oxford University Press.

- URZUA C. M. (1988)
A class of maximum-entropy multivariate distributions. *Communications in Statistics. Theory and Methods*, 17(12), pp 4039-4057.

- URZUA C. M. (1989)
Test for multivariate normality of observations and residuals. *Communications in statistics. Theory and Methods*, 17 (12), pp 4039-4057.

- URZUA C. M. (1991)
A note on the Sarque-Bera Test for normality. *El Colegio de México, manuscrito*.

- WHITE H. Y MACDONALD G. M. (1980).
Some large-sample tests for normality in the linear Regression Model. *Journal of the American Statistical Association*, 75, 369 pp 16-31.

APENDICE A

APENDICE A
RESUMEN DE RESULTADOS

TABLA 1: VALORES DE SIGNIFICANCIA DE LOS ESTADISTICOS DE ASIMETRIA Y KURTOSIS (A Y K, RESPECTIVAMENTE). 10000 REPETICIONES

SIG=SIGNIFICANCIA

SIG =	0.99		0.95		0.90		0.85		0.50	
N	A	K	A	K	A	K	A	K	A	K
10	−1.44	1.39	−0.98	1.57	−0.73	1.69	−0.59	1.77	0.00	2.29
15	−1.27	1.54	−0.85	1.72	−0.64	1.85	−0.51	1.94	0.00	2.45
20	−1.17	1.64	−0.78	1.84	−0.59	1.96	−0.47	2.04	0.00	2.55
25	−1.11	1.71	−0.72	1.92	−0.56	2.03	−0.44	2.12	0.00	2.61
30	−1.02	1.80	−0.67	1.98	−0.50	2.10	−0.40	2.19	0.01	2.67
35	−0.90	1.84	−0.62	2.02	−0.46	2.14	−0.37	2.23	0.00	2.70
40	−0.85	1.89	−0.59	2.07	−0.45	2.18	−0.35	2.27	0.00	2.73
50	−0.80	1.97	−0.54	2.16	−0.41	2.27	−0.33	2.35	0.01	2.77
65	−0.67	2.05	−0.48	2.23	−0.36	2.33	−0.29	2.41	0.00	2.81
80	−0.62	2.12	−0.42	2.29	−0.33	2.39	−0.27	2.47	0.00	2.84
100	−0.56	2.19	−0.39	2.34	−0.30	2.44	−0.24	2.51	0.00	2.87
125	−0.50	2.26	−0.35	2.41	−0.27	2.50	−0.22	2.57	0.00	2.90
150	−0.46	2.31	−0.32	2.45	−0.24	2.54	−0.20	2.60	0.00	2.91
200	−0.40	2.38	−0.28	2.52	−0.22	2.59	−0.18	2.65	0.00	2.93
250	−0.37	2.42	−0.25	2.55	−0.02	2.63	−0.16	2.68	0.00	2.94
300	−0.33	2.45	−0.23	2.59	−0.18	2.66	−0.15	2.71	0.00	2.95
400	−0.28	2.53	−0.20	2.64	−0.15	2.71	−0.12	5.75	0.00	2.96
500	−0.26	2.57	−0.18	2.67	−0.14	2.73	−0.11	2.77	0.00	2.96
800	−0.21	2.66	−0.14	2.74	−0.11	2.79	−0.09	2.82	0.00	2.98

SIG =	0.15		0.10		0.05		0.01	
N	A	K	A	K	A	K	A	K
10	0.58	3.18	0.74	3.46	0.97	3.95	1.42	5.06
15	0.53	3.32	0.66	3.62	0.84	4.16	1.23	5.37
20	0.47	3.41	0.58	3.69	0.76	4.17	1.15	5.44
25	0.43	3.41	0.54	3.71	0.73	4.18	1.08	5.37
30	0.41	3.43	0.52	3.69	0.67	4.10	98.00	5.15
35	0.39	3.43	0.48	3.66	0.62	4.05	0.92	5.02
40	0.35	3.44	0.44	3.66	0.58	4.02	0.84	5.02
50	0.32	3.41	0.39	3.62	0.51	3.96	0.77	4.88
65	0.30	3.42	0.37	3.61	0.48	3.93	0.69	4.66
80	0.26	3.40	0.33	3.56	0.43	3.86	0.64	4.55
100	0.24	3.38	0.30	3.54	0.39	3.79	0.56	4.38
125	0.22	3.34	0.27	3.47	0.35	3.69	0.51	4.33
150	0.20	3.33	0.24	3.45	0.32	3.65	0.47	4.14
200	0.17	3.29	0.21	3.40	0.27	3.57	0.40	4.03
250	0.15	3.27	0.19	3.36	0.25	3.53	0.37	3.86
300	0.14	3.26	0.18	3.35	0.23	3.48	0.34	3.76
400	0.13	3.22	0.16	3.29	0.20	3.41	0.28	3.67
500	0.11	3.21	0.14	3.27	0.18	3.37	0.25	3.59
800	0.09	3.17	0.11	3.22	0.14	3.30	0.20	3.46

TABLA 2: CONVERGENCIA A NORMALIDAD DE LOS ESTADISTICOS DE ASIMETRIA Y KURTOSIS ESTANDARIZADOS (SK Y K RESPECTIVAMENTE) INF. = VALOR ASINTOTICO

SIG =	0.99		0.95		0.90		0.85		0.50	
N	SK	K	SK	K	SK	K	SK	K	SK	K
10	-1.79	-1.04	-1.21	-0.93	-0.93	-0.85	-0.73	-0.79	-0.01	-0.46
15	-2.07	-1.15	-1.37	-1.01	-1.03	-0.91	-0.80	-0.83	0.00	-0.43
20	-2.08	-1.24	-1.39	-1.08	-1.05	-0.97	-0.83	-0.88	0.01	-0.41
25	-2.14	-1.30	-1.43	-1.09	-1.09	-0.96	-0.88	-0.88	0.01	-0.38
30	-2.25	-1.34	-1.48	-1.15	-1.13	-1.02	-0.90	-0.92	0.00	-0.38
35	-2.20	-1.39	-1.53	-1.17	-1.17	-1.03	-0.93	-0.93	0.00	-0.36
40	-2.25	-1.42	-1.53	-1.18	-1.16	-1.04	-0.93	-0.93	0.00	-0.35
50	-2.30	-1.51	-1.52	-1.24	-1.67	-1.08	-0.93	-0.95	-0.01	-0.33
65	-2.31	-1.57	-1.57	-1.28	-1.21	-1.10	-0.97	-0.98	-0.02	-0.30
80	-2.35	-1.61	-1.59	-1.30	-1.20	-1.12	-0.97	-0.98	0.01	-0.29
100	-2.29	-1.66	-1.59	-1.31	-1.23	-1.14	-0.99	-1.00	0.02	-0.27
125	-2.28	-1.70	-1.57	-1.36	-1.22	-1.15	-0.99	-1.01	0.00	-0.25
150	-2.27	-1.76	-1.60	-1.38	-1.24	-1.15	-0.99	-1.00	-0.01	-0.22
200	-2.33	-1.77	-1.61	-1.39	-1.23	-1.17	-1.00	-1.01	0.01	-0.20
250	-2.34	-1.83	-1.58	-1.44	-1.22	-1.20	-1.00	-1.03	0.00	-0.18
300	-2.31	-1.88	-1.62	-1.47	-1.26	-1.22	-1.02	-1.04	0.01	-0.19
400	2.35	-1.89	-1.65	-1.49	-1.27	-1.22	-1.02	-1.04	-0.01	-0.16
500	-2.38	-1.97	-1.63	-1.48	-1.25	-1.20	-1.02	-1.01	0.00	-0.14
800	-2.33	-2.00	-1.63	-1.50	-1.25	-1.22	-1.02	-1.02	0.02	0.00
INF.	2		1.64		-1.28		-1.03		0.00	

SIG =	0.15		0.10		0.05		0.01	
N	SK	K	SK	K	SK	K	SK	K
10	0.72	0.11	0.92	0.30	1.23	0.62	1.79	1.32
15	0.82	0.28	1.03	0.51	1.34	90.00	2.05	1.87
20	0.85	0.36	1.05	0.60	1.39	1.04	2.10	2.25
25	0.88	0.42	1.09	0.67	1.44	1.12	2.18	2.38
30	0.92	0.48	1.14	0.74	1.47	1.24	2.13	2.58
35	0.93	0.55	1.15	0.84	1.52	1.33	2.27	2.48
40	0.92	0.57	1.16	0.84	1.50	1.34	2.25	2.55
50	0.94	0.63	1.17	0.94	1.55	1.47	2.27	2.95
65	0.98	0.68	1.21	1.00	1.59	1.51	2.30	2.81
80	1.00	0.74	1.23	1.06	1.58	1.55	2.26	2.79
100	1.00	0.75	1.23	1.06	1.59	1.55	2.27	2.87
125	1.00	0.80	1.25	1.11	1.60	1.60	2.32	2.85
150	1.00	0.80	1.24	1.10	1.61	1.58	2.36	2.86
200	1.00	0.83	1.22	1.14	1.58	1.66	2.26	2.98
250	1.04	0.88	1.28	1.19	1.64	1.64	2.29	2.61
300	1.02	0.88	1.26	1.18	1.60	1.65	2.28	2.74
400	1.00	0.89	1.25	1.19	1.63	1.65	2.32	2.68
500	1.03	0.96	1.27	1.25	1.61	1.70	2.28	2.72
800	1.04	1.00	1.28	1.26	1.65	1.74	2.35	2.68
INF.	1.03		1.28		1.64		2.33	

TABLA 3: CONVERGENCIA A LA DISTRIBUCION CHI-CUADRADA (2) DE LOS ESTADISTICOS LM, K2, Y JBU

SIG =	0.99			0.95			0.90		
N	LM	K2	JBU	LM	K2	JBU	LM	K2	JBU
10	0.03	0.02	0.03	0.12	0.10	0.15	0.21	0.21	0.28
15	0.02	0.02	0.02	0.10	0.10	0.12	0.20	0.21	0.25
20	0.02	0.02	0.02	0.10	0.10	0.12	0.20	0.21	0.23
25	0.02	0.02	0.02	0.09	0.10	0.11	0.19	0.20	0.21
30	0.02	0.01	0.02	0.10	0.10	0.11	0.20	0.20	0.22
35	0.02	0.02	0.02	0.10	0.11	0.10	0.19	0.22	0.21
40	0.02	0.02	0.02	0.10	0.10	0.10	0.19	0.20	0.21
50	0.02	0.02	0.02	0.10	0.10	0.11	0.20	0.21	0.22
65	0.02	0.02	0.02	0.10	0.11	0.10	0.20	0.21	0.21
80	0.02	0.02	0.03	0.10	0.10	0.11	0.20	0.21	0.21
100	0.02	0.02	0.02	0.10	0.10	0.09	0.20	0.22	0.20
125	0.02	0.02	0.02	0.10	0.10	0.10	0.20	0.21	0.21
150	0.02	0.02	0.02	0.10	0.10	0.11	0.20	0.20	0.21
200	0.02	0.02	0.02	0.10	0.10	0.10	0.20	0.22	0.20
250	0.02	0.02	0.02	0.10	0.10	0.10	0.20	0.21	0.20
300	0.02	0.02	0.02	0.10	0.10	0.11	0.21	0.22	0.21
400	0.02	0.02	0.02	0.10	0.10	0.10	0.20	0.22	0.20
500	0.02	0.02	0.02	0.10	0.10	0.09	0.21	0.20	0.20
800	0.02	0.02	0.02	0.10	0.10	0.10	0.21	0.20	0.20
INF.		0.02			0.10			0.21	

SIG =	0.85			0.50			0.15		
N	LM	K2	JBU	LM	K2	JBU	LM	K2	JBU
10	0.29	0.31	0.40	0.69	1.20	1.11	1.30	3.37	2.44
15	0.29	0.32	0.36	0.86	1.25	1.12	1.61	3.45	2.62
20	0.29	0.31	0.34	0.86	1.27	1.18	1.80	3.43	2.66
25	0.30	0.30	0.32	0.91	1.30	1.17	2.02	3.68	2.78
30	0.29	0.31	0.32	0.96	1.27	1.18	2.18	3.52	2.88
35	0.28	0.34	0.31	0.97	1.31	1.20	2.32	3.60	2.91
40	0.30	0.32	0.31	1.00	1.33	1.20	2.34	3.58	2.93
50	0.29	0.32	0.32	1.04	1.33	1.23	2.52	3.59	3.05
65	0.29	0.32	0.32	1.08	1.35	1.24	2.65	3.71	3.10
80	0.29	0.31	0.31	1.13	1.30	1.23	2.86	3.64	3.28
100	0.30	0.32	0.31	1.16	1.33	1.28	2.93	3.80	3.36
125	0.30	0.32	0.31	1.17	1.33	1.27	3.04	3.63	3.28
150	0.30	0.31	0.33	1.21	1.31	1.30	3.14	3.56	3.38
200	0.30	0.33	0.31	1.24	1.35	1.30	3.29	3.74	3.42
250	0.30	0.32	0.32	1.25	1.38	1.33	3.29	3.71	3.48
300	0.31	0.34	0.33	1.27	1.38	1.30	3.38	3.84	3.46
400	0.31	0.33	0.32	1.26	1.38	1.33	3.42	3.76	3.54
500	0.32	0.31	0.30	1.32	1.36	1.32	3.51	3.78	3.57
800	0.32	0.31	0.32	1.32	1.37	1.36	3.58	3.78	3.75
INF.		0.32			1.39			3.80	

SIG =	0.10			0.05			0.01		
N	LM	K2	JBU	LM	K2	JBU	LM	K2	JBU
10	1.69	4.36	2.93	2.61	6.45	3.77	6.25	12.23	5.64
15	2.08	4.47	3.15	3.33	6.22	4.16	5.56	11.87	6.69
20	2.33	4.34	3.25	3.77	6.39	4.42	9.55	11.92	7.29
25	2.60	4.60	3.37	4.11	6.48	4.49	10.51	11.64	7.89
30	2.79	4.44	3.52	4.39	6.17	4.63	11.05	11.19	7.96
35	2.99	4.48	3.56	4.75	6.34	4.70	10.91	11.22	8.26
40	2.98	4.53	3.57	4.65	6.20	4.75	11.58	10.81	8.68
50	3.23	4.49	3.64	5.10	6.25	4.86	12.02	11.15	8.38
65	3.33	4.61	3.78	5.05	6.34	5.10	13.32	10.97	8.80
80	3.57	4.51	4.02	5.38	6.34	5.29	11.96	11.69	9.68
100	3.62	4.61	4.07	5.37	6.22	5.43	11.94	10.95	9.34
125	3.84	4.46	4.00	5.55	6.12	5.28	11.98	10.75	8.89
150	3.92	4.39	7.07	5.68	6.03	5.38	12.69	10.53	9.16
200	4.03	4.59	4.15	5.59	6.15	5.44	12.13	10.18	9.16
250	4.09	4.50	4.32	5.63	6.14	5.73	11.10	10.29	9.32
300	4.19	4.66	4.22	5.74	6.35	5.50	11.28	10.26	8.84
400	4.21	4.56	4.34	5.81	6.00	5.80	10.44	10.11	9.48
500	4.28	4.65	4.30	5.71	6.07	5.68	10.45	9.90	8.97
800	4.38	4.66	4.57	5.83	6.02	5.92	10.38	9.60	9.70
INF.		4.61			5.99			9.21	

TABLA 4: CONVERGENCIA A NORMALIDAD DE LOS ESTADISTICOS
Z1 Y Z2
INF. = VALOR ASINTOTICO

SIG =	0.99		0.95		0.90		0.85		0.50	
N	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2
10	-2.33	-2.03	-1.65	-1.53	-1.25	-1.21	-1.01	-1.02	0.00	-0.06
15	-2.36	-2.19	-1.66	-1.55	-1.28	-1.21	-1.04	-0.99	0.01	-0.02
20	-2.35	-2.23	-1.66	-1.63	-1.27	-1.27	-1.03	-1.03	0.00	-0.04
25	-2.40	-2.33	-1.67	-1.63	-1.28	-1.26	-1.04	-1.00	-0.02	0.00
30	-2.33	-2.39	-1.67	-1.64	-1.31	-1.28	-1.06	-1.04	-0.02	-0.01
35	-2.29	-2.41	-1.64	-1.63	-1.27	-1.26	-1.04	-1.03	0.00	0.00
40	-2.26	-2.46	-1.63	-1.66	-1.30	-1.29	-1.05	-1.03	-0.01	0.01
50	-2.40	-2.41	-1.69	-1.71	-1.33	-1.29	-1.08	-1.04	-0.02	0.03
65	-2.36	-2.42	-1.66	-1.67	-1.31	-1.29	-1.06	-1.04	0.00	0.02
80	-2.36	-2.43	-1.66	-1.68	-1.29	-1.31	-1.04	-1.05	0.00	0.03
100	-2.22	-2.52	-1.61	-1.73	-1.25	-1.31	-1.01	-1.03	0.01	0.00
125	-2.27	-2.48	-1.62	-1.69	-1.27	-1.31	-1.02	-1.03	0.03	0.02
150	-2.33	-2.43	-1.66	-1.67	-1.30	-1.29	-1.05	-1.05	0.00	0.01
200	-2.35	-2.44	-1.68	-1.71	-1.28	-1.29	-1.05	-1.06	-0.01	0.01
250	-2.32	-2.39	-1.63	-1.69	-1.27	-1.31	-1.03	-1.05	0.00	0.03
300	-2.32	-2.37	-1.65	-1.64	-1.27	-1.26	-1.03	-1.01	0.01	0.01
400	-2.34	-2.41	-1.65	-1.68	-1.29	-1.29	-1.03	-1.05	0.01	-0.02
500	-2.32	-2.42	-1.63	-1.68	-1.27	-1.30	-1.02	-1.03	0.00	0.02
800	-2.34	-2.34	-1.64	-1.64	-1.29	-1.26	-1.05	-1.01	0.00	0.01
INF.	-2.33		-1.64		-1.28		-1.03		0.00	

SIG =	0.15		0.10		0.05		0.01	
N	Z1	Z2	Z1	Z2	Z1	Z2	Z1	Z2
10	1.05	1.04	1.29	1.32	1.66	1.71	2.43	2.31
15	1.02	1.05	1.28	1.30	1.65	1.71	2.28	2.44
20	1.05	1.03	1.30	1.28	1.68	1.67	2.30	2.31
25	1.02	1.03	1.26	1.28	1.61	1.65	2.30	2.32
30	1.01	1.02	1.25	1.27	1.63	1.65	2.26	2.33
35	1.05	1.03	1.31	1.28	1.67	1.66	2.32	2.32
40	1.04	1.03	1.29	1.26	1.65	1.63	2.30	2.28
50	1.03	1.05	1.28	1.31	1.63	1.64	2.35	2.32
65	1.04	1.04	1.30	1.28	1.67	1.65	2.32	2.32
80	1.02	1.04	1.25	1.27	1.65	1.61	2.31	2.23
100	1.06	1.02	1.29	1.26	1.67	1.60	2.40	2.28
125	1.05	1.02	1.30	1.26	1.67	1.61	2.34	2.31
150	1.01	1.04	1.27	1.28	1.63	1.65	2.30	2.28
200	1.03	1.05	1.27	1.29	1.64	1.65	2.32	2.35
250	1.00	1.03	1.26	1.27	1.64	1.64	2.32	2.32
300	1.04	1.02	1.29	1.27	1.66	1.63	2.35	2.36
400	1.04	1.00	1.29	1.26	1.63	1.62	2.26	2.33
500	1.05	1.07	1.30	1.31	1.66	1.66	2.29	2.36
800	1.04	1.05	1.28	1.28	1.63	1.65	2.29	2.34
INF.	1.03		1.28		1.64		2.33	

TABLA 5: PODER ESTIMADO SOBRE OBSERVACIONES
VALORES DE SIGNIFICANCIA ESTIMADOS

	BETA(3,2)			BETA(2,2)		
N	LM	K2	JBU	LM	K2	JBU
10	0.076	0.074	0.131	0.049	0.055	0.044
15	0.063	0.076	0.141	0.030	0.063	0.163
20	0.068	0.097	0.162	0.023	0.090	0.196
25	0.067	0.102	0.183	0.020	0.118	0.230
30	0.083	0.130	0.201	0.027	0.165	0.267
35	0.083	0.159	0.242	0.028	0.208	0.312
40	0.099	0.185	0.280	0.037	0.262	0.364
50	0.124	0.252	0.350	0.052	0.370	0.464
65	0.227	0.338	0.457	0.154	0.501	0.581
80	0.316	0.462	0.527	0.248	0.648	0.673
100	0.482	0.593	0.674	0.427	0.771	0.793
125	0.662	0.768	0.825	0.628	0.890	0.903
150	0.798	0.877	0.906	0.777	0.947	0.949
200	0.930	0.962	0.976	0.926	0.989	0.990
250	0.986	0.993	0.994	0.983	0.998	0.998
300	0.997	0.998	0.999	0.997	1.000	1.000
400	1.000	1.000	1.000	1.000	1.000	1.000
500	1.000	1.000	1.000	1.000	1.000	1.000
800	1.000	1.000	1.000	1.000	1.000	1.000

	CAUCHY			CHI,2		
N	LM	K2	JBU	LM	K2	JBU
10	0.669	0.671	0.547	0.492	0.426	0.364
15	0.814	0.811	0.722	0.665	0.577	0.565
20	0.904	0.898	0.819	0.808	0.707	0.726
25	0.941	0.934	0.884	0.887	0.792	0.849
30	0.968	0.962	0.928	0.947	0.875	0.917
35	0.982	0.977	0.957	0.972	0.927	0.960
40	0.990	0.987	0.975	0.987	0.962	0.985
50	0.997	0.996	0.992	0.998	0.993	0.998
65	1.000	1.000	0.999	1.000	0.999	1.000
80	1.000	1.000	1.000	1.000	1.000	1.000
100	1.000	1.000	1.000	1.000	1.000	1.000
125	1.000	1.000	1.000	1.000	1.000	1.000
150	1.000	1.000	1.000	1.000	1.000	1.000
200	1.000	1.000	1.000	1.000	1.000	1.000
250	1.000	1.000	1.000	1.000	1.000	1.000
300	1.000	1.000	1.000	1.000	1.000	1.000
400	1.000	1.000	1.000	1.000	1.000	1.000
500	1.000	1.000	1.000	1.000	1.000	1.000
800	1.000	1.000	1.000	1.000	1.000	1.000

	EXP. CUARTICA			F(3,4)		
N	LM	K2	JBU	LM	K2	JBU
10	0.008	0.008	0.016	0.716	0.646	0.581
15	0.007	0.007	0.010	0.878	0.815	0.811
20	0.012	0.011	0.015	0.959	0.917	0.927
25	0.009	0.007	0.030	0.985	0.959	0.976
30	0.011	0.013	0.047	0.996	0.986	0.993
35	0.009	0.027	0.075	0.999	0.995	0.998
40	0.008	0.044	0.098	1.000	0.999	0.999
50	0.007	0.069	0.129	1.000	1.000	1.000
65	0.008	0.122	0.151	1.000	1.000	1.000
80	0.012	0.124	0.129	1.000	1.000	1.000
100	0.014	0.087	0.093	1.000	1.000	1.000
125	0.003	0.062	0.069	1.000	1.000	1.000
150	0.002	0.052	0.053	1.000	1.000	1.000
200	0.006	0.144	0.176	1.000	1.000	1.000
250	0.079	0.343	0.369	1.000	1.000	1.000
300	0.220	0.513	0.663	1.000	1.000	1.000
400	0.842	0.899	0.941	1.000	1.000	1.000
500	1.000	1.000	1.000	1.000	1.000	1.000
800	1.000	1.000	1.000	1.000	1.000	1.000

	GAMMA(2,1)			LOGNORMAL		
N	LM	K2	JBU	LM	K2	JBU
10	0.328	0.287	0.238	0.552	0.500	0.425
15	0.455	0.400	0.375	0.731	0.658	0.640
20	0.587	0.514	0.510	0.857	0.790	0.797
25	0.678	0.578	0.621	0.925	0.861	0.893
30	0.771	0.675	0.708	0.967	0.927	0.946
35	0.828	0.745	0.796	0.985	0.960	0.975
40	0.878	0.803	0.861	0.993	0.980	0.992
50	0.938	0.900	0.942	0.999	0.996	0.999
65	0.986	0.967	0.985	1.000	1.000	1.000
80	0.996	0.992	0.995	1.000	1.000	1.000
100	0.999	0.999	0.999	1.000	1.000	1.000
125	1.000	1.000	1.000	1.000	1.000	1.000
150	1.000	1.000	1.000	1.000	1.000	1.000
200	1.000	1.000	1.000	1.000	1.000	1.000
250	1.000	1.000	1.000	1.000	1.000	1.000
300	1.000	1.000	1.000	1.000	1.000	1.000
400	1.000	1.000	1.000	1.000	1.000	1.000
500	1.000	1.000	1.000	1.000	1.000	1.000
800	1.000	1.000	1.000	1.000	1.000	1.000

	MEZCLA DE NORMALES			NORMAL		
N	LM	K2	JBU	LM	K2	JBU
10	0.104	0.103	0.096	0.101	0.099	0.097
15	0.099	0.098	0.099	0.098	0.099	0.102
20	0.106	0.105	0.102	0.106	0.106	0.104
25	0.107	0.101	0.104	0.106	0.099	0.107
30	0.108	0.105	0.100	0.103	0.104	0.100
35	0.102	0.102	0.100	0.102	0.102	0.103
40	0.104	0.101	0.102	0.105	0.100	0.105
50	0.100	0.102	0.105	0.100	0.105	0.108
65	0.105	0.100	0.105	0.108	0.102	0.106
80	0.101	0.101	0.092	0.101	0.104	0.096
100	0.101	0.097	0.095	0.099	0.096	0.097
125	0.106	0.106	0.104	0.103	0.108	0.101
150	0.104	0.109	0.101	0.105	0.111	0.100
200	0.095	0.099	0.101	0.098	0.098	0.104
250	0.096	0.101	0.093	0.101	0.107	0.097
300	0.099	0.095	0.103	0.094	0.095	0.106
400	0.097	0.097	0.096	0.103	0.103	0.106
500	0.100	0.095	0.103	0.104	0.099	0.110
800	0.098	0.097	0.096	0.099	0.098	0.096

	T-STUDENT(5)			TUKEY		
N	LM	K2	JBU	LM	K2	JBU
10	0.201	0.201	0.138	0.102	0.104	0.099
15	0.260	0.253	0.195	0.092	0.093	0.095
20	0.316	0.301	0.234	0.100	0.099	0.099
25	0.352	0.337	0.277	0.096	0.089	0.099
30	0.396	0.375	0.305	0.103	0.098	0.091
35	0.428	0.401	0.337	0.098	0.096	0.092
40	0.462	0.426	0.367	0.100	0.098	0.099
50	0.514	0.487	0.429	0.096	0.099	0.099
65	0.588	0.543	0.495	0.096	0.089	0.092
80	0.644	0.605	0.543	0.093	0.095	0.086
100	0.713	0.673	0.629	0.091	0.087	0.084
125	0.778	0.747	0.708	0.090	0.094	0.091
150	0.836	0.800	0.768	0.092	0.097	0.090
200	0.902	0.877	0.863	0.084	0.083	0.087
250	0.945	0.924	0.911	0.084	0.087	0.080
300	0.970	0.955	0.953	0.084	0.081	0.087
400	0.989	0.984	0.983	0.080	0.081	0.082
500	0.997	0.995	0.995	0.078	0.073	0.081
800	1.000	1.000	1.000	0.079	0.077	0.076

TABLA 6: PODER ESTIMADO SOBRE RESIDUOS
VALORES DE SIGNIFICANCIA ESTIMADOS

	BETA(3,2)			GAMMA(2,1)		
N	LM	K2	JBU	LM	K2	JBU
10	0.103	0.128	0.125	0.188	0.286	0.150
15	0.081	0.086	0.127	0.362	0.328	0.282
20	0.068	0.091	0.145	0.469	0.428	0.396
25	0.067	0.095	0.153	0.578	0.515	0.506
30	0.084	0.116	0.178	0.699	0.606	0.620
35	0.073	0.128	0.199	0.750	0.670	0.710
40	0.082	0.152	0.226	0.811	0.735	0.784
50	0.119	0.205	0.306	0.905	0.844	0.898
65	0.165	0.309	0.392	0.968	0.948	0.969
80	0.257	0.394	0.490	0.991	0.983	0.992
100	0.446	0.548	0.652	0.999	0.997	0.999
125	0.607	0.715	0.775	1.000	1.000	1.000
150	0.740	0.818	0.869	1.000	1.000	1.000
200	0.917	0.947	0.963	1.000	1.000	1.000
250	0.982	0.990	0.993	1.000	1.000	1.000
300	0.996	0.998	0.999	1.000	1.000	1.000
400	1.000	1.000	1.000	1.000	1.000	1.000
500	1.000	1.000	1.000	1.000	1.000	1.000
800	1.000	1.000	1.000	1.000	1.000	1.000

	LOGNORMAL			NORMAL		
N	LM	K2	JBU	LM	K2	JBU
10	0.346	0.375	0.234	0.109	0.131	0.106
15	0.597	0.544	0.477	0.107	0.105	0.104
20	0.754	0.696	0.674	0.102	0.104	0.102
25	0.860	0.803	0.807	0.104	0.105	0.102
30	0.931	0.883	0.897	0.113	0.111	0.103
35	0.959	0.925	0.943	0.104	0.102	0.102
40	0.978	0.960	0.974	0.104	0.103	0.102
50	0.995	0.989	0.995	0.102	0.102	0.110
65	0.999	0.999	1.000	0.099	0.106	0.103
80	1.000	1.000	1.000	0.100	0.100	0.102
100	1.000	1.000	1.000	0.105	0.100	0.109
125	1.000	1.000	1.000	0.103	0.103	0.101
150	1.000	1.000	1.000	0.099	0.096	0.096
200	1.000	1.000	1.000	0.103	0.100	0.100
250	1.000	1.000	1.000	0.101	0.104	0.102
300	1.000	1.000	1.000	0.102	0.105	0.104
400	1.000	1.000	1.000	0.109	0.098	0.104
500	1.000	1.000	1.000	0.098	0.109	0.101
800	1.000	1.000	1.000	0.101	0.098	0.101

T-STUDENT(5)

N	LM	K2	JBU
10	0.145	0.232	0.103
15	0.225	0.211	0.159
20	0.273	0.263	0.199
25	0.324	0.304	0.243
30	0.377	0.341	0.281
35	0.393	0.365	0.310
40	0.424	0.392	0.334
50	0.490	0.451	0.404
65	0.567	0.532	0.475
80	0.626	0.581	0.532
100	0.708	0.657	0.626
125	0.768	0.728	0.693
150	0.819	0.777	0.751
200	0.896	0.867	0.854
250	0.936	0.919	0.907
300	0.966	0.954	0.950
400	0.988	0.982	0.981
500	0.996	0.995	0.994
800	1.000	1.000	1.000

TABLA 7: PODER ESTIMADO SOBRE RESIDUOS
VALOR DE SIGNIFICANCIA ASINTOTICO
SIG = 10%

	BETA(3,2)			GAMMA(2,1)		
N	LM	K2	JBU	LM	K2	JBU
10	0.010	0.066	0.035	0.105	0.272	0.104
15	0.012	0.071	0.047	0.237	0.391	0.235
20	0.013	0.083	0.063	0.354	0.494	0.359
25	0.013	0.101	0.080	0.466	0.577	0.474
30	0.014	0.121	0.095	0.568	0.659	0.579
35	0.016	0.148	0.119	0.658	0.734	0.675
40	0.017	0.178	0.147	0.734	0.795	0.756
50	0.025	0.239	0.202	0.852	0.892	0.873
65	0.049	0.338	0.303	0.952	0.967	0.964
80	0.102	0.446	0.417	0.988	0.991	0.992
100	0.221	0.593	0.570	0.998	0.999	0.999
125	0.428	0.751	0.739	1.000	1.000	1.000
150	0.628	0.856	0.851	1.000	1.000	1.000
200	0.878	0.961	0.961	1.000	1.000	1.000
250	0.973	0.992	0.992	1.000	1.000	1.000
300	0.994	0.999	0.998	1.000	1.000	1.000
400	1.000	1.000	1.000	1.000	1.000	1.000
500	1.000	1.000	1.000	1.000	1.000	1.000
800	1.000	1.000	1.000	1.000	1.000	1.000

	LOGNORMAL			NORMAL		
N	LM	K2	JBU	LM	K2	JBU
10	0.268	0.483	0.258	0.015	0.096	0.026
15	0.506	0.650	0.512	0.032	0.096	0.038
20	0.684	0.776	0.685	0.037	0.098	0.042
25	0.805	0.860	0.808	0.045	0.102	0.049
30	0.889	0.922	0.893	0.052	0.102	0.054
35	0.935	0.956	0.942	0.055	0.101	0.056
40	0.966	0.979	0.972	0.056	0.103	0.060
50	0.993	0.996	0.995	0.058	0.102	0.061
65	1.000	1.000	1.000	0.061	0.101	0.065
80	1.000	1.000	1.000	0.064	0.100	0.070
100	1.000	1.000	1.000	0.068	0.101	0.073
125	1.000	1.000	1.000	0.069	0.095	0.074
150	1.000	1.000	1.000	0.072	0.096	0.076
200	1.000	1.000	1.000	0.080	0.102	0.085
250	1.000	1.000	1.000	0.082	0.099	0.086
300	1.000	1.000	1.000	0.084	0.101	0.087
400	1.000	1.000	1.000	0.091	0.104	0.094
500	1.000	1.000	1.000	0.091	0.103	0.093
800	1.000	1.000	1.000	0.091	0.101	0.095

T-STUDENT(5)

N	LM	K2	JBU
10	0.064	0.188	0.056
15	0.137	0.246	0.121
20	0.231	0.288	0.169
25	0.256	0.328	0.213
30	0.297	0.357	0.244
35	0.343	0.396	0.282
40	0.381	0.425	0.311
50	0.448	0.478	0.367
65	0.523	0.540	0.440
80	0.597	0.601	0.508
100	0.671	0.667	0.590
125	0.747	0.736	0.672
150	0.808	0.793	0.741
200	0.889	0.877	0.847
250	0.938	0.928	0.911
300	0.965	0.957	0.947
400	0.988	0.984	0.981
500	0.996	0.995	0.994
800	1.000	1.000	1.000

TABLA 8: VALORES DE SIGNIFICANCIA CALCULADOS EMPIRICAMENTE CON 10000 REPETICIONES. SIG=10%

	OBSERVACIONES			RESIDUOS		
N	LM	K2	JBU	LM	K2	JBU
10	1.610	4.360	2.930	1.430	3.530	2.850
15	2.080	4.470	3.150	1.880	4.190	3.110
20	2.310	4.350	3.250	2.260	4.260	3.250
25	2.550	4.590	3.370	2.510	4.420	3.410
30	2.660	4.440	3.520	2.560	4.420	3.460
35	2.850	4.480	3.560	2.860	4.590	3.540
40	2.960	4.530	3.570	3.010	4.570	3.630
50	3.240	4.490	3.640	3.160	4.620	3.610
65	3.330	4.610	3.780	3.460	4.490	3.830
80	3.530	4.510	4.020	3.590	4.630	3.930
100	3.640	4.610	4.070	3.610	4.610	3.910
125	3.760	4.460	4.010	3.760	4.570	4.070
150	3.840	4.390	4.070	3.920	4.670	4.180
200	4.120	4.590	4.150	4.030	4.620	4.250
250	4.160	4.520	4.320	4.130	4.530	4.270
300	4.190	4.660	4.220	4.160	4.510	4.250
400	4.250	4.560	4.340	4.180	4.740	4.410
500	4.320	4.650	4.310	4.460	4.510	4.470
800	4.430	4.660	4.570	4.440	4.660	4.480

APENDICE B

APENDICE B
PROGRAMAS UTILIZADOS

PROGRAMA PARA PRUEBA DE PODER DEL ESTADISTICO LM,CONSIDERANDO DISTINTOS TAMAÑOS DE MUESTRA, DE OBSERVACIONES CON DISTRIBUCIONES DADAS:

1.- BETA(3,2) (SIG ES EL VECTOR DE VALORES CRITICOS DE LM PARA DISTINTOS TAMAÑOS DE MUESTRA, CALCULADOS EMPIRIRICAMENTE REALIZANDO 10000 REPETICIONES, SUPONIENDO DISTRIBUCION NORMAL DE LA MUESTRA SOBRE LA CUAL SE CALCULAN DICHOS VALORES. EL NIVEL DE SIGNIFICANCIA CONSIDERADO ES DE 10%).

```
new ,65520;
output file = b:rprjbb.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(1,5,nsize[jj],j+jj+2);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndbs(k1,k2,a,n,s);
  local g1,x,g2,g3,u,u1,u2;
  u=rndus(k1+k2,n,s);
```

```
  u1=u[1:k1,.];
  u2=u[(k1+1):(k1+k2),.];
  g1=-ln(prodc(u1))/a;
  g2=-ln(prodc(u2))/a;
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

2.-BETA(2,2)

```
new ,65520;
output file = b:rprjbb2.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(2,2,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
```

```
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

3.-CAUCHY

```
new ,65520;
output file = b:rprjbc.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndcs(nsize[jj],j+jj+2);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndcs(n,s);
  local c,x;
  x=rndns(n,2,s);
  c=x[.,1] ./ x[.,2];
  retp(c);
endp;
```

4.-CHI-CUADRADA, 2 g.l.

```
new ,65520;
output file = b:rprjbchi.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndchs(2,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndchs(m,n,s);
  local ch;
  ch=sumc(rndns(m,n,s)^2);
  retp(ch);
endp;
```

5.-EXPONENCIAL CUARTICA

```
new ,65520;
output file = b:rprjbec.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
```

```
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndecs(nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndecs(n,s);
 local i,j,x,ut,ec,s1,s2;
 i=1;
 j=0;
 ec=zeros(n,1);
 do until i>n;
  s1=s+j;
  s2=s1+1;
  x=rndns(1,1,s1);
  ut=exp((x^4)/4)*rndus(1,1,s2);
  if ut<=1;
   ec[i,1]=x;
   i=i+1;
  endif;
  j=j+1;
 endo;
 retp(ec);
endp;
```

6.- F(3,4)

```
new ,65520;
output file = b:rprjbf.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
```

```
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndfs(3,4,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndfs(m,k,n,s);
  local rn,x,y,f;
  rn=rndns(m+k,n,s);
  x=sumc(rn[1:m,.]^2)/m;
  y=sumc(rn[m+1:m+k,.]^2)/k;
  f=x./y;
  retp(f);
endp;
```

7.-GAMMA(2,1)

```
new ,65520;
output file = b:rprjbg.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
```

```
 j=1;
 do while j<=10000;
  x=rndgs(2,1,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndgs(k,a,n,s);
  local g,u;
  u=rndus(n,k,s);
  g=-ln(prodc(u'))/a;
  retp(g);
endp;
```

8.-LOGNORMAL

```
new ,65520;
output file = c:rprjblog.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndlogs(1,1,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
```

```
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

9.-MEZCLA DE DOS NORMALES

```
new ,65520;
output file = c:rprjbmn.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndmns(.5,0,1,1,1,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
```

```
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndmns(c,m1,m2,s1,s2,n,s);
  local z,mn;
  z=rndns(n,2,s);
  mn=c*(s1*z[.,1]+m1)+(1-c)*(s2*z[.,2]+m2);
  retp(mn);
endp;
```

10.-NORMAL

```
new ,65520;
output file = c:rprjbn.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 s=j+jj;
 do while j<=10000;
  x=rndns(nsize[jj],1,s);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
```

```
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
```

11.-T-STUDENT (5)

```
new ,65520;
output file = b:rprjbt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndts(5,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(m+1,n,s);
  x=u[1,.]';
  y=u[2:(m+1),.];
  t=x./sqrt(sumc(y^2)/m);
  retp(t);
endp;
```

12.-TUKEY

```
new ,65520;
output file = b:rprjbtuk.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.61,2.08,2.30,2.55,2.66,2.85,2.96,3.24,3.33,3.53,3.64,3.76,
3.84,4.12,4.16,4.19,4.25,4.32,4.43};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndtuks(0,.1975,.1349,.1349,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndtuks(l1,l2,l3,l4,n,s);
  local u,z;
  u=rndus(n,1,s);
  z=l1+(1/l2)*((u^l3)-(1-u)^l4);
  retp(z);
endp;
```

PROGRAMAS PARA LA EVALUACION DEL ESTADISTICO K2.AQUI SE HA SUSTITUIDO LOS VALORES CRITICOS DE LOS PROGRAMAS ANTERIORES POR LOS ESTIMADOS EMPIRICAMENTE PARA K2, PARA EL MISMO NIVEL DE SIGNIFICANCIA DE 10%.

13.-BETA(3,2)

```
new ,65520;
output file = b:rprdpb.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(3,2,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
```

```
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

14.-BETA(2,2)

```
new ,65520;
output file = b:rprdpbb.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(2,2,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
```

```
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

15.-CAUCHY

```
new ,65520;
output file = b:rprdpc.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndcs(nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
```

```
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndcs(n,s);
  local c,x;
  x=rndns(n,2,s);
  c=x[.,1] ./ x[.,2];
  retp(c);
endp;
```

16.-CHI-CUADRADA, 2 g.l.

```
new ,65520;
output file = b:rprdpchi.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndchs(2,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
```

```
    b[jj,1]=b[jj,1]+1;
   endif;
   j=j+1;
  endo;
  b[jj,1]=b[jj,1]/10000;
  output on;
  print "PARA N=";;nsize[jj];
  print "FRACCION =";;b[jj];
  output off;
 jj=jj+1;
 endo;
 proc pruebadp(x);
  local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
 zb2,k2;
   n=rows(x);
   m1=meanc(x);
   m2=sumc((x-m1)^2)/n;
   sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
   b2=(sumc((x-m1)^4)/n)/(m2^2);
   y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
   b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
   w2=-1+sqrt(2*(b2sb1-1));
   d=1/sqrt(.5*ln(w2));
   a=sqrt(2/(w2-1));
   zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
   eb2=3*(n-1)/(n+1);
   vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
   sb2=(b2-eb2)/(sqrt(vb2));
   sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
          3)))/((n+7)*(n+9));
   aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
   zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
        4))))^(1/3))/sqrt(2/(9*aa));
   k2=((zsb1)^2+(zb2)^2);
   retp(k2);
 endp;
 proc rndchs(m,n,s);
   local ch;
   ch=sumc(rndns(m,n,s)^2);
   retp(ch);
 endp;
```

17.-EXPONENCIAL CUARTICA

```
new ,65520;
output file = b:rprdpec.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
```

```
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndecs(nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndecs(n,s);
 local i,j,x,ut,ec,s1,s2;
 i=1;
 j=0;
 ec=zeros(n,1);
 do until i>n;
  s1=s+j;
  s2=s1+1;
  x=rndns(1,1,s1);
  ut=exp((x^4)/4)*rndus(1,1,s2);
  if ut<=1;
   ec[i,1]=x;
```

```
    i=i+1;
   endif;
   j=j+1;
  endo;
  retp(ec);
 endp;
```

18.-F(3,4)

```
new ,65520;
output file = b:rprdpf.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndfs(3,4,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
```

```
          3)))/((n+7)*(n+9));
   aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
   zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
        4))))^(1/3))/sqrt(2/(9*aa));
   k2=((zsb1)^2+(zb2)^2);
   retp(k2);
endp;
proc rndfs(m,k,n,s);
   local rn,x,y,f;
   rn=rndns(m+k,n,s);
   x=sumc(rn[1:m,.]^2)/m;
   y=sumc(rn[m+1:m+k,.]^2)/k;
   f=x./y;
   retp(f);
endp;
```

19.-GAMMA(2,1)

```
new ,65520;
output file = b:rprdpg.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndgs(2,1,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
```

```
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndgs(k,a,n,s);
  local g,u;
  u=rndus(n,k,s);
  g=-ln(prodc(u'))/a;
  retp(g);
endp;
```

20.-LOGNORMAL

```
new ,65520;
output file = b:rprdplog.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndlogs(1,1,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
```

```
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

21.-MEZCLA DE DOS NORMALES

```
new ,65520;
output file = b:rprdpmn.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndmns(.5,0,1,1,1,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
```

```
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndmns(c,m1,m2,s1,s2,n,s);
  local z,mn;
  z=rndns(n,2,s);
  mn=c*(s1*z[.,1]+m1)+(1-c)*(s2*z[.,2]+m2);
  retp(mn);
endp;
```

22.-NORMAL

```
new ,65520;
output file = c:rprdpn.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 s=j+jj;
```

```
 do while j<=10000;
  x=rndns(nsize[jj],1,s);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
```

23.-T-STUDENT (5)

```
new ,65520;
output file = b:rprdpt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
```

```
 j=1;
 do while j<=10000;
  x=rndts(5,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(m+1,n,s);
  x=u[1,.]';
  y=u[2:(m+1),.];
  t=x./sqrt(sumc(y^2)/m);
  retp(t);
endp;
```

24.-TUKEY

```
new ,65520;
```

```
output file = b:rprdptuk.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={4.36,4.47,4.35,4.59,4.44,4.48,4.53,4.49,4.61,4.51,4.61,4.46,
4.39,4.59,4.50,4.66,4.56,4.65,4.66};
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndtuks(0,.1945,.1349,.1349,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndtuks(l1,l2,l3,l4,n,s);
  local u,z;
  u=rndus(n,1,s);
  z=l1+(1/l2)*((u^l3)-(1-u)^l4);
```

```
  retp(z);
endp;
```

PROGRAMAS PARA EVALUACION DEL ESTADISTICO JBU, CON SUS RESPECTIVOS VALORES CRITICOS. SIGNIFICANCIA DE 10%.

25.-BETA(3,2)

```
new ,65520;
output file = b:rprurb.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(3,2,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
```

```
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

26.-BETA(2,2)

```
new ,65520;
output file = b:rprurbb.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(2,2,nsize[jj],j+jj),
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
```

```
endp;
```

27.-CAUCHY

```
new ,65520;
output file = c:rprurc.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndcs(nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndcs(n,s);
  local c,x;
  x=rndns(n,2,s);
  c=x[.,1] ./ x[.,2];
  retp(c);
endp;
```

28.-CHI-CUADRADA, 2

```
new ,65520;
output file = b:rprurchi.doc reset;
```

```
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndchs(2,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndchs(m,n,s);
  local ch;
  ch=sumc(rndns(m,n,s)^2);
  retp(ch);
endp;
```

29.-EXPONENCIAL CUARTICA

```
new ,65520;
output file = c:rprurec1.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
```

```
 do while j<=10000;
  x=rndecs(nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndecs(n,s);
 local i,j,x,ut,ec,s1,s2;
 i=1;
 j=0;
 ec=zeros(n,1);
 do until i>n;
  s1=s+j;
  s2=s1+1;
  x=rndns(1,1,s1);
  ut=exp((x^4)/4)*rndus(1,1,s2);
  if ut<=1;
   ec[i,1]=x;
   i=i+1;
  endif;
  j=j+1;
 endo;
 retp(ec);
endp;
```

30.-F(3,4)

```
new ,65520;
output file = b:rprurf.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
```

```
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndfs(3,4,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndfs(m,k,n,s);
  local rn,x,y,f;
  rn=rndns(m+k,n,s);
  x=sumc(rn[1:m,.]^2)/m;
  y=sumc(rn[m+1:m+k,.]^2)/k;
  f=x./y;
  retp(f);
endp;
```

31.-GAMMA(2,1)

```
new ,65520;
output file = b:rprurg.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndgs(2,1,nsize[jj],j+jj);
```

```
   a[j,1]=pruebaur(x);
   if a[j,1] >= sig[jj,1];
    b[jj,1]=b[jj,1]+1;
   endif;
   j=j+1;
  endo;
  b[jj,1]=b[jj,1]/10000;
  output on;
  print "PARA N=";;nsize[jj];
  print "FRACCION =";;b[jj];
  output off;
 jj=jj+1;
 endo;
 proc pruebaur(x);
   local m1,m2,n,sb1,b2,jbur;
   n=rows(x);
   m1=meanc(x);
   m2=sumc((x-m1)^2)/n;
   sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
   b2=(sumc((x-m1)^4)/n)/(m2^2);
   jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
   retp(jbur);
 endp;
 proc rndts(m,n,s);
   local t,x,y,u;
   u=rndns(m+1,n,s);
   x=u[1,.]';
   y=u[2:(m+1),.];
   t=x./sqrt(sumc(y^2)/m);
   retp(t);
 endp;
 proc rndgs(k,a,n,s);
   local g,u;
   u=rndus(n,k,s);
   g=-ln(prodc(u'))/a;
   retp(g);
 endp;
```

32.-LOGNORMAL

```
new ,65520;
output file = c:rprurlog.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndlogs(1,1,nsize[jj],j+jj);
```

```
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

33.-MEZCLA DE DOS NORMALES

```
new ,65520;
output file = c:rprurmn.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndmns(.5,0,1,1,1,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
```

```
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndmns(c,m1,m2,s1,s2,n,s);
  local z,mn;
  z=rndns(n,2,s);
  mn=c*(s1*z[.,1]+m1)+(1-c)*(s2*z[.,2]+m2);
  retp(mn);
endp;
```

34.-NORMAL

```
new ,65520;
output file = b:rprurn.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  x=rndns(nsize[jj],1,s);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
```

```
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
```

35.-T-STUDENT (5)

```
new ,65520;
output file = b:rprurt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndts(5,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(m+1,n,s);
  x=u[1,.]';
```

```
  y=u[2:(m+1),.];
  t=x./sqrt(sumc(y^2)/m);
  retp(t);
endp;
```

36.-TUKEY

```
new ,65520;
output file = b:rprurtuk.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.93 3.15 3.25 3.37 3.52 3.56 3.57 3.64 3.78 4.02 4.07
4.00 4.07 4.15 4.32 4.22 4.34 4.30 4.57;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndtuks(0,.1975,.1349,.1349,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndtuks(l1,l2,l3,l4,n,s);
  local u,z;
  u=rndus(n,1,s);
  z=l1+(1/l2)*((u^l3)-(1-u)^l4);
  retp(z);
endp;
```

PROGRAMAS UTILIZADOS PARA EVALUAR LOS ESTADISTICOS LM, K2 Y JBU, CONSIDERANDO AHORA LOS RESIDUOS DE CMO (LOS VALORES CRITICOS SON LOS CALCULADOS EMPIRICAMENTE PARA EL CASO DE RESIDUOS)

PROGRAMAS PARA LM

37.-BETA(3,2)

```
new ,65520;
output file = c:rprjbbr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.43,1.88,2.26,2.50,2.56,2.86,3.01,3.16,3.46,3.59,
     3.60,3.76,3.92,4.03,4.13,4.16,4.18,4.46,4.44};
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndbs(3,2,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebajb(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndbs(k1,k2,n,s);
```

```
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

38.-GAMMA(2,1)

```
new ,65520;
output file = c:rprjbgr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.43,1.88,2.26,2.50,2.56,2.86,3.01,3.16,3.46,3.59,
     3.60,3.76,3.92,4.03,4.13,4.16,4.18,4.46,4.44};
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndgs(2,1,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebajb(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
```

```
endp;
proc rndgs(k,a,n,s);
  local g,u;
  u=rndus(n,k,s);
  g=-ln(prodc(u'))/a
  retp(g);
endp;
```

39.-LOGNORMAL

```
new ,65520;
output file = c:rprjblogr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.43,1.88,2.26,2.50,2.56,2.86,3.01,3.16,3.46,3.59,
     3.60,3.76,3.92,4.03,4.13,4.16,4.18,4.46,4.44};
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndlogs(1,1,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebajb(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
x=rndlogs(a,b,10000,200);
proc rndlogs(a,b,n,s);
```

```
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

40.-NORMAL

```
new ,65520;
output file = b:rprjbnr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.43,1.88,2.26,2.50,2.56,2.86,3.01,3.16,3.46,3.59,
     3.60,3.76,3.92,4.03,4.13,4.16,4.18,4.46,4.44};
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  er=rndns(nsize[jj],1,s);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebajb(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
```

41.-T-STUDENT

```
new ,65520;
output file = b:rprjbtr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
sig={1.43,1.88,2.26,2.50,2.56,2.86,3.01,3.16,3.46,3.59,
     3.60,3.76,3.92,4.03,4.13,4.16,4.18,4.46,4.44};
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndts(5,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebajb(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(m+1,n,s);
  x=u[1,.]';
  y=u[2:(m+1),.];
  t=x./sqrt(sumc(y^2)/m);
  retp(t);
endp;
```

PROGRAMAS PARA K2

42.-BETA(3,2)

```
new ,65520;
output file = c:rprdpbr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=3.52 4.19 4.26 4.42 4.42 4.59 4.58 4.62 4.49 4.63 4.60
4.57 4.67 4.62 4.53 4.51 4.74 4.50 4.66;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndbs(3,2,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebadp(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
```

```
        3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

43.-GAMMA(2,1)

```
new ,65520;
output file = c:rprdpgr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=3.52 4.19 4.26 4.42 4.42 4.59 4.58 4.62 4.49 4.63 4.60
4.57 4.67 4.62 4.53 4.51 4.74 4.50 4.66;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndgs(2,1,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebadp(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebadp(x);
```

```
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndgs(k,a,n,s);
  local g,u;
  u=rndus(n,k,s);
  g=-ln(prodc(u'))/a;
  retp(g);
endp;
```

44.-LOGNORMAL

```
new ,65520;
output file = c:rprdplogr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=3.52 4.19 4.26 4.42 4.42 4.59 4.58 4.62 4.49 4.63 4.60
4.57 4.67 4.62 4.53 4.51 4.74 4.50 4.66;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndlogs(1,1,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebadp(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
```

```
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
x=rndlogs(a,b,10000,200);
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

45.-NORMAL

```
new ,65520;
output file = b:rprdpnr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=3.52 4.19 4.26 4.42 4.42 4.59 4.58 4.62 4.49 4.63 4.60
```

```
4.57 4.67 4.62 4.53 4.51 4.74 4.50 4.66;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  er=rndns(nsize[jj],1,s);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebajb(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
        3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
```

46.-T-STUDENT

```
new ,65520;
output file = b:rprdptr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=3.52 4.19 4.26 4.42 4.42 4.59 4.58 4.62 4.49 4.63 4.60
4.57 4.67 4.62 4.53 4.51 4.74 4.50 4.66;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndts(5,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebadp(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
```

```
        4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(m+1,n,s);
  x=u[1,.]';
  y=u[2:(m+1),.];
  t=x./sqrt(sumc(y^2)/m);
  retp(t);
endp;
```

PROGRAMAS PARA JBU

47.-BETA(2,1)

```
new ,65520;
output file = b:rprurbr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.85 3.10 3.25 3.41 3.46 3.54 3.63 3.61 3.83 3.93 3.91
4.07 4.18 4.25 4.27 4.25 4.41 4.47 4.48;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndbs(3,2,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebaur(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
```

```
   m1=meanc(x);
   m2=sumc((x-m1)^2)/n;
   sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
   b2=(sumc((x-m1)^4)/n)/(m2^2);
   jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
   retp(jbur);
 endp;
 proc rndbs(k1,k2,n,s);
   local g1,g2,g3,x,u,u1,u2;
   u=rndus(n,k1+k2,s);
   u1=u[.,1:k1];
   u2=u[.,(k1+1):(k1+k2)];
   g1=-ln(prodc(u1'));
   g2=-ln(prodc(u2'));
   g3=g1+g2;
   x=g1./g3;
   retp(x);
 endp;
```

48.-GAMMA(2,1)

```
new ,65520;
output file = b:rprurgr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.85 3.10 3.25 3.41 3.46 3.54 3.63 3.61 3.83 3.93 3.91
4.07 4.18 4.25 4.27 4.25 4.41 4.47 4.48;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndgs(2,1,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebaur(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebaur(x);
```

```
   local m1,m2,n,sb1,b2,jbur;
   n=rows(x);
   m1=meanc(x);
   m2=sumc((x-m1)^2)/n;
   sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
   b2=(sumc((x-m1)^4)/n)/(m2^2);
   jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
   retp(jbur);
endp;
proc rndgs(k,a,n,s);
   local g,u;
   u=rndus(n,k,s);
   g=-ln(prodc(u'))/a;
   retp(g);
endp;
```

49.-LOGNORMAL

```
new ,65520;
output file = b:rprurlor.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.85 3.10 3.25 3.41 3.46 3.54 3.63 3.61 3.83 3.93 3.91
4.07 4.18 4.25 4.27 4.25 4.41 4.47 4.48;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndlogs(1,1,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebaur(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebaur(x);
   local m1,m2,n,sb1,b2,jbur;
   n=rows(x);
   m1=meanc(x);
```

```
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

50.-NORMAL

```
new ,65520;
output file = b:rprurnr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.85 3.10 3.25 3.41 3.46 3.54 3.63 3.61 3.83 3.93 3.91
4.07 4.18 4.25 4.27 4.25 4.41 4.47 4.48;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  er=rndns(nsize[jj],1,s);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebaur(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
```

```
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
```

51.-T-STUDENT (5)

```
new ,65520;
output file = c:rprurtr.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
let sig=2.85 3.10 3.25 3.41 3.46 3.54 3.63 3.61 3.83 3.93 3.91
4.07 4.18 4.25 4.27 4.25 4.41 4.47 4.48;
jj=1;
b=zeros(rows(nsize),1);
z=rndu(nsize[jj],2);
x=ones(nsize[jj],1)~z;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  er=rndts(5,nsize[jj],j+jj);
  esig=er-x*(invpd(x'x))*(x'*er);
  a[j,1]=pruebaur(esig);
  if a[j,1] >= sig[jj,1];
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
z=z|rndu((nsize[jj]-nsize[jj-1]),2);
x=ones(nsize[jj],1)~z;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(m+1,n,s);
```

```
  x=u[1,.]';
  y=u[2:(m+1),.];
  t=x./sqrt(sumc(y^2)/m);
  retp(t);
endp;
```

PROGRAMAS UTILIZADOS PARA EVALUAR LOS ESTADISTICOS LM, K2 Y JBU, CONSIDERANDO COMO VALOR CRITICO EL PRESENTADO EN LAS TABLAS DE CHI-CUADRADA PARA 2 g.l., O SEA, EL VALOR ASINTOTICO 4.61, CORRESPONDIENTE UN NIVEL DE SIGNIFICANCIA DE 10%.

PROGRAMAS PARA LM

52.-BETA(3,2)

```
new ,65520;
output file = c:rprjbbt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(3,2,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndbs(k1,k2,n,s);
```

```
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

53.-GAMMA(2,1)

```
new ,65520;
output file = c:rprjbgt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndgs(2,1,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndgs(k,a,n,s);
  local g,u;
  u=rndus(n,k,s);
  g=-ln(prodc(u'))/a;
  retp(g);
endp;
```

54.-LOGNORMAL

```
new ,65520;
output file = c:rprjblot.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndlogs(1,1,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

55.-NORMAL

```
new ,65520;
output file = c:rprjbnt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
```

```
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  x=rndns(nsize[jj],1,s);
  a[j,1]=pruebajb(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
```

56.-T-STUDENT (5)

```
new ,65520;
output file = c:rprjbtt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  x=rndts(5,nsize[jj],j+jj);
  a[j,1]=pruebajb(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
```

```
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(n,m+1,s);
  x=u[.,1];
  y=u[.,2:(m+1)];
  t=x./sqrt(sumc((y^2)')/m);
  retp(t);
endp;
```

PROGRAMAS PARA K2

57.-BETA(3,2)

```
new ,65520;
output file = c:rprdpbt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(3,2,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
```

```
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
   n=rows(x);
   m1=meanc(x);
   m2=sumc((x-m1)^2)/n;
   sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
   b2=(sumc((x-m1)^4)/n)/(m2^2);
   y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
   b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
   w2=-1+sqrt(2*(b2sb1-1));
   d=1/sqrt(.5*ln(w2));
   a=sqrt(2/(w2-1));
   zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
   eb2=3*(n-1)/(n+1);
   vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
   sb2=(b2-eb2)/(sqrt(vb2));
   sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
          3)))/((n+7)*(n+9));
   aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
   zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
        4))))^(1/3))/sqrt(2/(9*aa));
   k2=((zsb1)^2+(zb2)^2);
   retp(k2);
endp;
proc rndbs(k1,k2,n,s);
   local g1,g2,g3,x,u,u1,u2;
   u=rndus(n,k1+k2,s);
   u1=u[.,1:k1];
   u2=u[.,(k1+1):(k1+k2)];
   g1=-ln(prodc(u1'));
   g2=-ln(prodc(u2'));
   g3=g1+g2;
   x=g1./g3;
   retp(x);
endp;
```

58.-GAMMA(2,1)

```
new ,65520;
output file = c:rprdpgt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndgs(2,1,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
```

```
   if a[j,1] >= 4.61;
    b[jj,1]=b[jj,1]+1;
   endif;
   j=j+1;
  endo;
  b[jj,1]=b[jj,1]/10000;
  output on;
  print "PARA N=";;nsize[jj];
  print "FRACCION =";;b[jj];
  output off;
 jj=jj+1;
 endo;
 proc pruebadp(x);
  local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
 zb2,k2;
   n=rows(x);
   m1=meanc(x);
   m2=sumc((x-m1)^2)/n;
   sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
   b2=(sumc((x-m1)^4)/n)/(m2^2);
   y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
   b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
   w2=-1+sqrt(2*(b2sb1-1));
   d=1/sqrt(.5*ln(w2));
   a=sqrt(2/(w2-1));
   zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
   eb2=3*(n-1)/(n+1);
   vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
   sb2=(b2-eb2)/(sqrt(vb2));
   sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
          3)))/((n+7)*(n+9));
   aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
   zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
        4))))^(1/3))/sqrt(2/(9*aa));
   k2=((zsb1)^2+(zb2)^2);
   retp(k2);
 endp;
 proc rndgs(k,a,n,s);
   local g,u;
   u=rndus(n,k,s);
   g=-ln(prodc(u'))/a;
   retp(g);
 endp;
```

59.-LOGNORMAL

```
new ,65520;
output file = c:rprdplot.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
```

```
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndlogs(1,1,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
x=rndlogs(a,b,10000,200);
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

60.-NORMAL

```
new ,65520;
output file = c:rprdpnt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  x=rndns(nsize[jj],1,s);
  a[j,1]=pruebadp(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
```

61.-T-STUDENT (5)

```
new ,65520;
output file = c:rprdptt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndts(5,nsize[jj],j+jj);
  a[j,1]=pruebadp(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
```

```
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(n,m+1,s);
  x=u[.,1];
  y=u[.,2:(m+1)];
  t=x./sqrt(sumc((y^2)')/m);
  retp(t);
endp;
```

PROGRAMAS PARA JBU

62.-BETA(3,2)

```
new ,65520;
output file = c:rprurbt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndbs(3,2,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
```

```
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
  retp(x);
endp;
```

63.-GAMMA(2,1)

```
new ,65520;
output file = c:rprurgt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndgs(2,1,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj].
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndbs(k1,k2,n,s);
  local g1,g2,g3,x,u,u1,u2;
  u=rndus(n,k1+k2,s);
  u1=u[.,1:k1];
  u2=u[.,(k1+1):(k1+k2)];
  g1=-ln(prodc(u1'));
  g2=-ln(prodc(u2'));
  g3=g1+g2;
  x=g1./g3;
```

```
  retp(x);
endp;
proc rndgs(k,a,n,s);
  local g,u;
  u=rndus(n,k,s);
  g=-ln(prodc(u'))/a;
  retp(g);
endp;
```

64.-LOGNORMAL

```
new ,65520;
output file = c:rprurlot.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndlogs(1,1,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndlogs(a,b,n,s);
  local x,c,d,z;
  d=ln(1+b/(a^2));
  c=ln(a)-.5*d;
  z=rndns(n,1,s);
  x=exp(c+sqrt(d)*z);
  retp(x);
endp;
```

65.-NORMAL

```
new ,65520;
output file = c:rprurnt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  s=j+jj;
  x=rndns(nsize[jj],1,s);
  a[j,1]=pruebaur(x);
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
```

66.-T-STUDENT (5)

```
new ,65520;
output file = c:rprurtt.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
b=zeros(rows(nsize),1);
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndts(5,nsize[jj],j+jj);
  a[j,1]=pruebaur(x);
```

```
  if a[j,1] >= 4.61;
   b[jj,1]=b[jj,1]+1;
  endif;
  j=j+1;
 endo;
 b[jj,1]=b[jj,1]/10000;
 output on;
 print "PARA N=";;nsize[jj];
 print "FRACCION =";;b[jj];
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
proc rndts(m,n,s);
  local t,x,y,u;
  u=rndns(n,m+1,s);
  x=u[.,1];
  y=u[.,2:(m+1)];
  t=x./sqrt(sumc((y^2)')/m);
  retp(t);
endp;
```

PROGRAMAS UTILIZADOS PARA ESTIMAR LOS VALORES CRITICOS DE OBSERVACIONES Y RESIDUOS DE LOS TRES ESTADISTICOS.

PROGRAMA PARA LM, OBSERVACIONES

```
new ,65520;
output file = b:rjbsiaut.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndn(nsize[jj],1);
  a[j,1]=pruebajb(x);
  j=j+1;
 endo;
 b=sortc(a,1);
 output on;
```

```
 print "PARA N=";;nsize[jj];
 print "99% =";;b[100,1];
 print "95% =";;b[500,1];
 print "90% =";;b[1000,1];
 print "85% =";;b[1500,1];
 print "50% =";;b[5000,1];
 print "15% =";;b[8500,1];
 print "10% =";;b[9000,1];
 print "5% =";;b[9500,1];
 print "1% =";;b[9900,1];
 print " ";
 output off;
jj=jj+1;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jb=n*(((sb1)^2/6)+((b2-3)^2)/24);
  retp(jb);
endp;
```

PROGRAMA PARA K2, OBSERVACIONES

```
new ,65520;
output file = b:rdpsiaut.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndn(nsize[jj],1);
  a[j,1]=pruebadp(x);
  j=j+1;
 endo;
 b=sortc(a,1);
 output on;
 print "PARA N=";;nsize[jj];
 print "99% =";;b[100,1];
 print "95% =";;b[500,1];
 print "90% =";;b[1000,1];
 print "85% =";;b[1500,1];
 print "50% =";;b[5000,1];
 print "15% =";;b[8500,1];
 print "10% =";;b[9000,1];
 print "5% =";;b[9500,1];
 print "1% =";;b[9900,1];
 print " ";
```

```
 output off;
jj=jj+1;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
```

PROGRAMA PARA JBU, OBSERVACIONES

```
new ,65520;
output file = b:rursiaut.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
do while jj <= rows(nsize);
 a=zeros(10000,1);
 j=1;
 do while j<=10000;
  x=rndn(nsize[jj],1);
  a[j,1]=pruebaur(x);
  j=j+1;
 endo;
 b=sortc(a,1);
 output on;
 print "PARA N=";;nsize[jj];
 print "99% =";;b[100,1];
 print "95% =";;b[500,1];
 print "90% =";;b[1000,1];
 print "85% =";;b[1500,1];
 print "50% =";;b[5000,1];
 print "15% =";;b[8500,1];
```

```
 print "10% =";;b[9000,1];
 print "5% =";;b[9500,1];
 print "1% =";;b[9900,1];
 print " ";
 output off;
jj=jj+1;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
```

PROGRAMA PARA LM, RESIDUOS

```
new ,65520;
output file = b:rjbsiuni.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
 z=rndu(nsize[jj],2);
 x=ones(nsize[jj],1)~z;
 do while jj <= rows(x);
 a=zeros(10000,1);
 ixx=invpd(x'x);
 j=1;
 do while j<=10000;
  er=rndn(nsize[jj],1);
  esig=er-(x*ixx)*(x'*er);
  a[j,1]=pruebajb(esig);
  j=j+1;
 endo;
 b=sortc(a,1);
 output on;
 print "PARA N=";;nsize[jj];
 print "99% =";;b[100,1];
 print "95% =";;b[500,1];
 print "90% =";;b[1000,1];
 print "85% =";;b[1500,1];
 print "50% =";;b[5000,1];
 print "15% =";;b[8500,1];
 print "10% =";;b[9000,1];
 print "5% =";;b[9500,1];
 print "1% =";;b[9900,1];
 print " ";
 output off;
 jj=jj+1;
 z=z|rndu((nsize[jj]-nsize[jj-1]),2);
```

```
 x=ones(nsize[jj],1)~z;
endo;
proc pruebajb(x);
  local m1,m2,n,sb1,b2,jb;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  jb=n*(((sb1^2)/6)+((b2-3)^2)/24);
  retp(jb);
endp;
```

PROGRAMA PARA K2, RESIDUOS

```
NEW , 65520;
output file = b:rdpsiuni.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
 z=rndu(nsize[jj],2);
 x=ones(nsize[jj],1)~z;
 do while jj <= rows(x);
 a=zeros(10000,1);
 ixx=invpd(x'x);
 j=1;
 do while j<=10000;
  er=rndn(nsize[jj],1);
  esig=er-(x*ixx)*(x'*er);
  a[j,1]=pruebadp(esig);
  j=j+1;
 endo;
 b=sortc(a,1);
 output on;
 print "PARA N=";;nsize[jj];
 print "99% =";;b[100,1];
 print "95% =";;b[500,1];
 print "90% =";;b[1000,1];
 print "85% =";;b[1500,1];
 print "50% =";;b[5000,1];
 print "15% =";;b[8500,1];
 print "10% =";;b[9000,1];
 print "5% =";;b[9500,1];
 print "1% =";;b[9900,1];
 print " ";
 output off;
 jj=jj+1;
 z=z|rndu((nsize[jj]-nsize[jj-1]),2);
 x=ones(nsize[jj],1)~z;
endo;
proc pruebadp(x);
 local m1,m2,n,sb1,b2,d,y,b2sb1,w2,d,a,zsb1,eb2,vb2,sb2,sb1b2,aa,
zb2,k2;
```

```
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  y=sb1*sqrt((n+1)*(n+3)/(6*(n-2)));
  b2sb1=(3*(n^2+27*n-70)*(n+1)*(n+3))/((n-2)*(n+5)*(n+7)*(n+9));
  w2=-1+sqrt(2*(b2sb1-1));
  d=1/sqrt(.5*ln(w2));
  a=sqrt(2/(w2-1));
  zsb1=d*ln(y/a+sqrt(1+(y/a)^2));
  eb2=3*(n-1)/(n+1);
  vb2=24*n*(n-2)*(n-3)/(((n+1)^2)*(n+3)*(n+5));
  sb2=(b2-eb2)/(sqrt(vb2));
  sb1b2=(6*(n^2-(5*n)+2))*sqrt(6*(n+3)*(n+5)/(n*(n-2)*(n-
         3)))/((n+7)*(n+9));
  aa=6+16/(sb1b2^2)+8*sqrt(1+4/(sb1b2^2))/sb1b2;
  zb2=(1-2/(9*aa)-((1-2/aa)/(1+sb2*sqrt(2/(aa-
       4))))^(1/3))/sqrt(2/(9*aa));
  k2=((zsb1)^2+(zb2)^2);
  retp(k2);
endp;
```

PROGRAMA PARA JBU, RESIDUOS

```
new ,65520;
output file = b:rursiuni.doc reset;
let nsize=10 15 20 25 30 35 40 50 65 80 100 125 150 200 250 300
400 500 800;
jj=1;
 z=rndu(nsize[jj],2);
 x=ones(nsize[jj],1)~z;
 do while jj <= rows(x);
 a=zeros(10000,1);
 ixx=invpd(x'x);
 j=1;
 do while j<=10000;
  er=rndn(nsize[jj],1);
  esig=er-(x*ixx)*(x'*er);
  a[j,1]=pruebaur(esig);
  j=j+1;
 endo;
 b=sortc(a,1);
 output on;
 print "PARA N=";;nsize[jj];
 print "99% =";;b[100,1];
 print "95% =";;b[500,1];
 print "90% =";;b[1000,1];
 print "85% =";;b[1500,1];
 print "50% =";;b[5000,1];
 print "15% =";;b[8500,1];
 print "10% =";;b[9000,1];
 print "5% =";;b[9500,1];
```

```
 print "1% =";;b[9900,1];
 print " ";
 output off;
 jj=jj+1;
 z=z|rndu((nsize[jj]-nsize[jj-1]),2);
 x=ones(nsize[jj],1)~z;
endo;
proc pruebaur(x);
  local m1,m2,n,sb1,b2,jbur;
  n=rows(x);
  m1=meanc(x);
  m2=sumc((x-m1)^2)/n;
  sb1=(sumc((x-m1)^3)/n)/(m2^1.5);
  b2=(sumc((x-m1)^4)/n)/(m2^2);
  jbur=n*(((sb1^2)/6)+(27/8)*(((1/b2)-(1/3))^2));
  retp(jbur);
endp;
```