Alianza Universidad Textos

Juan Ignacio Peña Sánchez de Rivera
María Milagrosa Saiz Jarabo

Aplicaciones estadísticas y matemáticas del programa GAUSS

Alianza Editorial

Para
Carmina y mis padres
Angel y mis hijas

Calle Milán, 38, 28043 Madrid; teléf. 200 00 45
ISBN: 84-206-8141-5
Depósito legal: M. 37471-1991
Fotocomposición: EFCA, S. A.
Dr. Federico Rubio y Galí, 16. 28039 Madrid
Impreso en LERKO PRINT, S. A.
Santa Engracia, 139. 28003 Madrid.
Printed in Spain

INDICE

PREFACIO

El objetivo de este libro es la aplicación de un potente lenguaje informático, el GAUSS, al planteamiento y resolución de problemas simples en Matemáticas y Estadística. Este programa de ordenador está específicamente orientado para su uso en ordenadores personales y constituye una alternativa relevante para cualquier persona interesada en la resolución de problemas numéricos de cualquier naturaleza.

Por ello, pensamos que este texto puede ser una ayuda para estudiantes de facultades de Económicas, Ciencias o Ingeniería, que sean capaces de usar el ordenador para resolver sus problemas de Matemáticas y Estadística.

El libro contiene los siguientes capítulos: el primero, de introducción y requisitos básicos para su uso. El segundo, dedicado a los primeros pasos y las operaciones más simples con GAUSS. El tercero y cuarto, dedicado a aplicaciones matemáticas y el quinto y sexto, a aplicaciones estadísticas. Se incluye un apéndice con los comandos básicos de GAUSS, su significado y ejemplos.

En todos los casos se ha tratado de ilustrar el manejo del programa con ejemplos tomados de los problemas de Matemáticas y Estadística de Primer Ciclo de la Facultad de Ciencias Económicas de la Universidad Autónoma de Madrid.

El enfoque ha sido algo diferente según se tratase de Matemáticas o Estadística. Para Matemáticas se presentan problemas sencillos, que hacen hincapié en el manejo de comandos elementales de GAUSS, mientras que en la parte de Estadística se ha optado por presentar casos que requieren programas extensos y el uso de las posibilidades gráficas de GAUSS. Tanto en Matemáticas como en Estadística, se presentan los conceptos teóricos fundamentales previos a su aplicación e implementación mediante GAUSS.

Los autores agradecen especialmente los detallados comentarios de Alvaro Escribano, así como las sugerencias de Eduardo Giménez y otros colegas de las Universidades Carlos III y Autónoma de Madrid, que han contribuido a mejorar la presentación del texto, así como la eficaz labor de mecanografía de M.ª del Carmen Muñoz y Lidia Gómez Palomar. Este trabajo ha sido financiado en parte por la DGICYT, proyecto n.º PB87-0104.

Madrid, abril 1991

Capítulo 1
INTRODUCCION

1.1. Requisitos iniciales

Este texto pretende ayudar al conocimiento de las Matemáticas y la Estadística Descriptiva usando comandos del programa GAUSS.

Para un aprovechamiento óptimo del mismo es conveniente que el lector cuente con las siguientes capacidades:

a) Un conocimiento básico del sistema operativo MS-DOS.
b) Acceso a un ordenador personal IBM compatible de las familias PC-XT-AT-PS/2.
c) Disponibilidad de uso de la versión de estudiante (SV) de GAUSS 2.0, para la cual no se requiere coprocesador matemático. Si se puede trabajar con la versión completa de GAUSS 2.0, es imprescindible el coprocesador.
d) Disponibilidad de un manual de GAUSS 2.0 en su versión completa, incluyendo las rutinas de gráficos QUICKGRAPHICS.

El programa puede adquirirse en sus diferentes versiones solicitándolo a APTECH SYSTEMS, 26250 196th Place S.E., KENT WASHINGTON 98042, USA.

1.2. ¿Qué es el GAUSS?

El sistema GAUSS para análisis estadístico y matemático está compuesto del lenguaje de programación GAUSS y de un conjunto de aplicaciones concretas, programadas en ese lenguaje.

Este es un lenguaje de ordenador especialmente diseñado para su uso en computadores personales (PC en adelante) que trabajen con el sistema operativo DOS en versiones 2.1 y superiores. Por tanto, trabaja en los equipos IBM PC-XT-AT PS/2 y compatibles. En la actualidad hay disponibles tres versiones del sistema GAUSS.

A un nivel básico existe la Student Versión (SV), compuesta por la gran mayoría de comandos básicos del lenguaje. No necesita coprocesador matemático ni disco fijo (hard disk) y contiene una serie de comandos para generar gráficos de alta resolución (QuickGraphics).

El siguiente nivel es la versión «general» de GAUSS, que en la actualidad es la versión 2.10, que necesita coprocesador matemático y disco fijo. Adicionalmente a QuickGraphics soporta gráficos de muy alta calidad y resolución (Publication Graphics). Además de todos los comandos del lenguaje GAUSS, contiene una gran cantidad de subprogramas y rutinas (procedimientos) escritas en el lenguaje básico general y que simplifican considerablemente su uso.

El nivel más potente es el GAUSS 386-VM que trabaja solamente en máquinas con procesador Intel 80386 (con coprocesador matemático) y que permite utilizar memoria virtual, es decir, usar el disco fijo como si fuera una extensión de la memoria RAM. Con esto se consigue que el espacio de trabajo salte la barrera de los 640K que impone el DOS, pudiéndose emplear como memoria todo el espacio libre del disco fijo. De hecho, al trabajar con memoria extendida (no confundir con la memoria expandida, que es como máximo de 32MB, véase Yraolagoitia (1989)), el programa puede gestionar hasta 4GB (4096MB) de memoria usable. Requiere DOS 3.3 o superior.

En este libro tomaremos como referencia la SV ya que es la más básica y las más sencilla de usar; por tanto, no se emplearan los procedimientos existentes en 2.1 ni las facilidades de extensión de memoria de GAUSS 386-VM.

GAUSS forma parte de lo que se ha dado en llamar la «Nueva Raza» de programas de uso general para PC y se perfila como una de las opciones deseables para cualquier usuario que necesite usar su PC para cálculos matemáticos de cualquier índole y en cualquier área de actividad.

Una buena revisión de las capacidades de GAUSS así como su comparación con otros programas del mismo estilo, puede verse en Dolton (1988).

1.3. Funciones, procedimientos y programas

Una de las mayores ventajas de GAUSS consiste en su facilidad para trabajar con procedimientos creados por el usuario. Los procedimientos son conjuntos de instrucciones creados mediante la sucesión de comandos de GAUSS que efectúan un cálculo o tarea concreta.

Una vez creado un procedimiento, puede usarse como si fuerse un comando más del lenguaje GAUSS, permitiendo la extensión del mismo hasta cubrir cualquier necesidad de cálculo.

Es importante diferenciar entre un «procedimiento» y un «programa». Un procedimiento funciona como si fuera un comando más de GAUSS y, en general, producirá resultados numéricos en su forma más primaria (i.e. un vector o matriz de números). Si deseamos presentar éstos de forma más cómoda (con etiquetas, gráficos, y «maquillajes» variados) debemos construir un programa específico para ello. Utilizando lenguaje familiar a los sufridos programadores en FORTRAN, un «programa» es el programa principal y un «procedimiento» es muy similar a una subrutina.

Como es esperable, los procedimientos y programas podrán trabajar con variables «locales» (sólo existentes cuando se ejecutan éstos) y variables «globales» (activas durante toda la sesión de GAUSS).

Asimismo existen «funciones» que son procedimientos de una sola línea. Ejemplos de todos ellos se verán a continuación.

1.3.1. Funciones

Empezando con el concepto más simple, éste es el de funciones. Se trata de procedimientos de una sola línea que llevan a cabo un cálculo complicado. Su sintaxis es simple:

fn nombre (variable) = expresión;

Ejemplos:

1) Cálculo del coeficiente de asimetría de una variable. Se define como:

$$asm = suma\,((x(i) - xmedia)^3)\ /\ (n*desvtipica^3)$$

Necesitamos, por tanto, comandos de GAUSS que calculen la media de un vector (xmedia), su desviación típica (desvtipica), el número de datos (n) y la suma de todos sus elementos (suma).

Los comandos adecuados son: MEANC, STDC, ROWS y SUMC. La forma de la función con todos los comandos será:

```
fn asim(x) = sumc((x-meanc(x))^3) / (rows(x)*(stdc(x)^3));
```

Para usar ésta función bastaría teclear por ejemplo:

```
coefas = asim(datos);
```

donde «datos» es la variable que contiene los datos de interés y es de la forma de un vector columna. Se creará una nueva variable «coefas» que será un vector 1×1 cuyo elemento es el coeficiente de asimetría de la variable «datos».

El uso habitual de funciones es dentro de procedimientos más complicados para evitar cálculos repetitivos. Están diseñados para trabajar sólo con variables locales por lo que es conveniente usarlos en ese contexto.

1.3.2. Procedimientos

Los procedimientos se emplean para realizar cálculos complejos de modo recursivo. Pueden usarse como si fuera una parte más del lenguaje GAUSS y serán tan complejos o tan sencillos como el usuario desee. Dentro de un procedimiento puede manejarse cualquier comando de GAUSS, o cualquier función. Pueden usarse tanto las variables y símbolos globales como emplear variables locales.

La estructura será siempre la misma y constará de cinco secciones.

1) Nombre del procedimiento	(PROC)
2) Declaración de variables locales	(LOCAL)
3) Cálculos (sentencias GAUSS genéricas)	
4) Envío de resultados al programa principal	(RETP)
5) Fin del procedimiento	(ENDP)

De las sentencias anteriores sólo son imprescndibles PROC y ENDP ya que sin ellas no se identifica ni el comienzo ni el final del procedimiento. Por el contrario las sentencias LOCAL y RETP son opcionales y pueden omitirse o bien repetirse el número de veces que se desee, en cualquier lugar del procedimiento.

Como ejemplo de un procedimiento veamos el cálculo de los coeficientes de asimetría y curtosis de una variable.

```
PROC (2) = asicurt(x);
LOCAL as, cu;
as = sumc((x-meanc(x))^3)/(rows(x)*(stdc(x)^3));
cu = sumc((x-meanc(x))^4)/(rows(x)*(stdc(x)^4));
RETP(as, cu);
ENDP;
```

Para usar este procedimiento, basta escribir:

```
{coef1,coef2}=asicurt(datos);
```


Ahora los coeficientes de asimetría y curtosis de la variable «datos» están almacenados en las variables globales «coef1» y «coef2».

Veamos en detalle cada una de las sentencias del procedimiento.

— PROC, define el nombre del procedimiento (aquí es «asicurt»; vale cualquier nombre con 8 caracteres como máximo), el número de valores que se devuelven al programa principal (en éste caso son 2: «as» y «cu»; el máximo es 31 y por defecto es 1) y los argumentos externos que se usan en el procedimiento (en éste caso sólo hay uno: «x» que debe ser un vector columna; como máximo se admiten 31).

— LOCAL, define las variables que van a usarse en modo local para los cálculos del programa (en éste caso «as» y «cu»). Nótese que las variables que se emplean como argumentos (en éste caso «x») son siempre locales. También pueden pasarse como argumentos nombres de procedimientos o funciones ya existentes.

En ese caso el formato del comando será:

```
LOCAL var1,var2,procedi1:proc,funcion1:fn;
```

Tenemos aquí dos variables (var1, var1), un procedimeinto (procedi1) y una función (funcion1) que pueden usarse para cualquier cálculo dentro del procedimiento de referencia.

Nótese que las variables locales desaparecen al finalizar el uso del procedimiento y por tanto no aparecerán en el comando SHOW (véase comandos SHOW y LSHOW en el manual de GAUSS).

— RETP, se usa para devolver resultados al programa principal, que en el ejemplo son los coeficientes calculados de asimetría y curtosis. Se admiten no sólo nombres de variables (RETP(x)) sino también expresiones (RETP(x*y/10)) y ambas cosas a la vez (RETP(x, y, x*y/10)) con un máximo de 31. Es imprescindible que el número de elementos devueltos al programa principal coincida con los expresados en PROC. En el caso de «asicurt» se devuelven dos elementos (PROC (2) =).

— ENDP, señala el final del procedimiento. Es de uso obligatorio.

Si se construyen procedimientos que van a usarse de nuevo en el futuro, es una buena idea salvarlos en un fichero con extensión G (bien en el directorio principal de GAUSS o mejor en el subdirectorio SRC), ya que así cada vez que se usen, GAUSS los encontrará y compilará sin problemas. Además se incluyen automáticamente en el HELP general del sistema (alt-H, H y nombre del procedimiento). También pueden incluirse dentro de librerías especiales (véase comando LIBRARY).

Si se crea un procedimiento que no contenga referencias globales y que no llame a otros procedimientos ni funciones del usuario, puede crearse una versión compilada con el comando SAVE que produce un programa objeto con extensión .FCP, de acceso bastante más rápido que los de extensión .G (véase punto 5.4 del manual de GAUSS).

Técnicas avanzadas para el uso de procedimientos se exponen en el manual de GAUSS, en los puntos 5.5 (cómo utilizar procedimientos dentro de otros procedimientos), 5.6 (indexado de procedimientos) y 5.7 (combinación de procedimientos con devolución de resultados múltiples).

1.3.3. Programas

Los programas son conjuntos de instrucciones de GAUSS que pueden ejecutar tareas de la complejidad que se desee.

Pueden usar como elementos las funciones y los procedimientos mencionados anteriormente.

En general, tendrán tres secciones:

a) Definición de variables y entrada de datos.
b) Cálculos numéricos.
c) Presentación de resultados (tablas y gráficos).

Ejemplos detallados de programas se presentan en los capítulos quinto y sexto de éste libro en los casos prácticos de Estadística.

Referencias

Dolton, P. J. (1988). A new breed of software? GAUSS, MATLAB and PC-ISP, a comparative review. *Journal of Economic Surveys*, 2, 77-95.

Yraolagoitia, J. de (1989). La memoria expansiva EMS 4.0. *PCWORLD*, 49, 139-154.

Capítulo 2
PRIMEROS PASOS CON GAUSS

2.1. Introducción: Ejecutando GAUSS

Este capítulo presenta los primeros pasos que cualquier usuario deberá realizar para obtener un primer contacto con las potencialidades de GAUSS. Es una versión revisada y ampliada del TUTORIAL que se encuentra en los manuales del GAUSS 2.0.

Ahora mostramos como comenzar, parar y suspender la ejecución de GAUSS.

Arranque del programa

A partir del subdirectorio donde esté instalado, GAUSS se empieza tecleando:

GAUSS

2.1.1. Salida del programa

Para salir de GAUSS, apretar la tecla Esc. Responda sí (yes) a la indicación que aparece en la pantalla.

2.1.2. *Parada temporal de GAUSS*

Para parar temporalmente GAUSS y usar el sistema operativo, manteniendo intactas las matrices y procedimientos que esté utilizando, teclee:

```
DOS
```

Lo cual permite volver temporalmente a DOS. Puede usar cualquier comando DOS, hacer funcionar otro programa (pero recuerde que GAUSS sigue residente en su memoria RAM) o cualquier cosa que necesite y después volver a GAUSS. Par volver teclee:

```
EXIT
```

Deberá estar de vuelta en la pantalla de GAUSS. Este comando puede usarse en cualquier momento (un comando equivalente es ALT+Z).

2.2. Creación de grupos de datos

Ahora veremos cómo crear matrices y conjuntos de caracteres alfabéticos.

2.2.1. *Matrices*

Hay tres formas de crear matrices de datos:

— *Sentencia LET:*

```
let x = {1 2 3, 4 5 6, 7 8 9};
```

Acaba de crear una matriz 3×3 llamada x. Para mostrarla en pantalla

```
print x;
```

o bien simplemente

```
x;
```

pruebe otra vez sin LET:

```
x = {1 2 3, 4 5 6, 7 8 9};
```


El LET es opcional si hay corchetes ({) antes y después de los números.

Ahora creamos un vector columna y lo imprimimos

```
x = {1, 2, 3, 4, 5, 6, 7, 8, 9};
print x;
```

Ahora un vector fila:

```
x = {1 2 3 4};
print x;
```

Nótese que el punto y coma (;) es optativo si se trabaja en modo interactivo (modo COMMAND), pero será obligatorio cuando se creen conjuntos de instrucciones (procedimientos) que se ejecutan en bloque.

— *Función CON:*

La función CON permite la introducción de los elementos de una matriz de forma interactiva, es decir, directamente desde el teclado. Teclee esto:

```
x = CON(2, 2);
```

El ? aparece en la pantalla. La función CON está pidiéndole 4 números. Teclee 4 números seguidos separados por un espacio.

Al finalizar, se ha creado una matriz 2×2, que puede verse inmediatamente en la pantalla.

— *Números aleatorios:*

GAUSS permite crear fácilmente matrices de números aleatorios uniformes, U(0, 1) o bien normales estándard, N(0, 1) mediante las sentencias

```
x = RNDU(2, 2);   /* crea matriz 2×2 de U(0, 1) */
Y = RNDN(2, 2);   /* crea matriz 2×2 de N(0, 1) */
```

Si no se dice otra cosa, la semilla de generación de estos números varía cada vez que se ejecuta el comando RNDU o RNDN. Si se desea fijar la semilla (y, por tanto, obtener siempre el mismo conjunto de números) se usa el comando RNDUS o RDNDS.

2.2.2. *Conjuntos de caracteres alfabéticos*

Mostraremos dos formas de crear una cadena de caracteres (frases).

— *Sentencia de asignación simple*

Teclee:

S = "Esto es una frase";

Para visualizar en pantalla una frase, se emplea el mismo comando que para las matrices

print s;

— *La función CONS*

La función CONS le pide que introduzca a través del teclado una frase, teclee:

s = cons; print;

(el print es para generar una línea en blanco).

El cursor no está visible. Escriba algo y pulse *enter*. El cursor aparece de nuevo. Imprima s. También puede hacerse con un indicador de manera que podamos saber cuando se nos pide introducir algo.

print "Entre la frase:";; s = cons; print;

Introduzca algo. Luego imprima s. El doble punto y coma al final de la sentencia de imprimir, suprime la línea de petición subsiguiente de modo que la sentencia CONS empezará a pedir que se introduzcan datos en la misma línea.

Para unir dos frases teclee, por ejemplo:

```
b = "buenos";
d = "días";
c = b $+ d;
print c;
 buenos dias
```

El símbolo $+ es el operador de unión de frases.

2.3. Edición de ficheros

Se emplea aquí el editor básico de GAUSS pero recuerde que puede usarse cualquier otro (por ejemplo, el PE2 de IBM) siempre que se defina en el fichero STARTUP.

2.3.1. *Edición de un fichero en disco*

Creamos un programa que se almacena en un fichero en un disco. Aunque se puede usar cualquier editor que se desee, si se define previamente en el fichero STARTUP (véase sección 2.6 del manual de GAUSS), vamos a emplear en este caso el editor propio de GAUSS.

Teclee:

edit PASO3;

Aparecerá la pantalla en blanco con «New file» (nuevo fichero) en la parte superior, durante unos segundos. En la parte inferior de la pantalla está la línea de estado. En ésta, figura el nombre del fichero y la línea y la columna en la que está situado el cursor. Vamos a escribir un pequeño programa:

Teclee las líneas que van saliendo en el ordenador según van apareciendo exactamente. El error es a propósito.

```
let A[3, 3] = 6 8 −2 4 5 9 7 −6 5;
b = {1, 2, 3;
x = b/A;
print A;
print b;
print x;
```

Para que ejecute su programa, que se almacenará en un fichero de disco que se llamará *PASO3*, presione *Alt-X* (lo almacena) y después *R* (hace que se ejecute).

Le apareció un mensaje de error. Presione *Ctrl-F1* para volver al fichero y así poder hallar el error. Ahora presione *F7* para traer el mensaje de error al fichero y refrescarle la memoria. Mire al mensaje, dice que tiene un error de sintaxis (paréntesis sin cerrar) en la línea 2 del fichero PASO3.

Presione *Esc* para volver al fichero y corregir el error. Presione *Alt-X* y después *R* para hacerlo funcionar otra vez.

Acaba de resolver un sistema de ecuacones lineales de la forma Ax = b. Es un ejemplo del operador / (barra).

Para limpiar el fichero que contiene los mensajes de error, basta con presionar *Shift-F7* para conseguirlo.

Si volvemos al *modo Comando* podemos verificar si la resolución de la ecuación ha sido correcta. Si la ecuación anterior es cierta, A*x-b tiene que ser cero. Para ello, teclee:

```
print A*x;
print b;
print A*x-b;
```

El valor obtenido debe ser prácticamente cero, excepto por errores de redondeo.

Ahora presione *Ctrl-F1* otra vez para volver al fichero PASO3. Vamos a trabajar con el operador / en un tipo diferente de problema.

Presione *Alt-K* para suprimir la primera línea. Teclee esto:

```
x = { 1 2 3,
      4 5 6,
      7 8 9 };
```

Localice el cursor al principio de la línea que empieza con b = . Presione *Alt-I*. ¿Ve el cursor grande? Quiere decir que está en un modo de sobreimpresión.

```
y = 2;
```

Puede usar *Alt-K* otra vez si quiere o usar el espaciador para deshacerse de lo que le queda de la antigua línea.

Edite el resto del programa que le aparecerá como éste:

```
x = { 1 2 3,
      4 5 6,
      7 8 9 };
y = 2;
z = x / y;
print x;
print;
print y;
print z;
```

Si necesita ayuda para recordar los comandos de edición, presione *Alt-H*. *Esc* le sacará de la pantalla de ayuda y le devolverá al fichero.

Ejecute el nuevo programa. Recuerde, presione *Alt-X* y después *R*.

El operador / está haciendo ahora una división normal. Obtendrá:

```
print x;
1 2 3
4 5 6
7 8 9
print;
print Y;
2
print z;
0.5   1   1.5
 2   2.5   3
3.5   4   4.5
```

2.3.2. *Operaciones elemento a elemento*

Ahora vuelva a su programa y cambie las líneas segunda y tercera de manera que el programa quede:

```
x = { 1 2 3,
      4 5 6,
      7 8 9 };
y = { 1 2 3 };                    /* vector fila 3×1 */
z = x./ y;                     /* operador punto barra */
print x;
print;
print y;
print z;
```

recuerde que «/* vector fila 3×1 */» es un comentario. Los comentarios no se ejecutan, pero son útiles para documentar el programa. Pueden ponerse en cualquier lugar con el formato /* comentario */.

Ejecute su nuevo programa. Debería obtener:

```
1 2 3
4 5 6
7 8 9

1 2 3

1  2  1
4 2.5 2
7  4  3
```

Esto es una división elemento a elemento (primera fila dividida por primer elemento, etc.). La barra inclinada con punto siempre hace división elemento a elemento.

Vuelva el fichero. Recuerde *Ctrl-F1*.

Modifíquelo del modo siguiente:

```
x = { 1 2 3,
      4 5 6,
      7 8 9 };
y = { 1, 2, 3 };      /* vector columna 3×1 */
z = x./ y;            /* operador punto barra */
print x;
print y;
print z;
```

Use *Alt-D* para hacer desaparecer la línea print;

Ejecútelo (ALT-X, R). Obtendrá:

```
1 2 3
4 5 6
7 8 9

1
2
3

1            2            3
2            2.5          3
2.3333333    2.6666667    3
```

Aquí el vector columna se traslada a través de la matriz.

Ahora salga de GAUSS presionando la tecla *Esc* y contestando sí. Vuelva a GAUSS y haga funcionar el programa al mismo tiempo tecleando:

```
GAUSS PASO3
```

Presione *Alt-F2* desde el modo de Comandos para volver a ejecutar el programa. También puede ejecutar el programa desde el modo de Comandos tecleando RUN PASO3.

2.4. Ficheros y procedimientos

Ahora crearemos un pequeño fichero de datos.

2.4.1. Lectura de un fichero ASCII

```
edit DATOS
```

Crearemos un fichero ASCII de datos. Teclee los siguientes números:

```
1 2 3 1
3 2 7 4
9 2 6 3
5 1 2 7
4 8 5 2
1 7 5 7
```

Presione *Alt-X*, después *W* para salvar éstos en el fichero *datos*. Podemos entonces cargarlos en una matriz.

```
load x[6, 4] = datos;
print x;
```

Este es el modo rápido y sencillo de usar *LOAD* para pequeños ficheros de datos. Vea LOAD en la sección COMMAND REFERENCE para usarlo de una forma que tiene en cuenta un control de errores. Para cargar ficheros grandes de datos, existe el programa ATOG.EXE que necesita para funcionar el coprocesador matemático. Véase el capítulo 14 de utilidades del manual de GAUSS.

Para crear un grupo de datos GAUSS usando la matriz *x*, usaremos *SAVED*.

```
call saved(x, "mydata", 0);
```

Esta sentencia crea un fichero de datos llamado *mydata.dat*. Para comprobar si se ha creado, teclee:

```
dos dir mydata;
```

Un grupo de datos GAUSS consiste en dos ficheros: uno con una extensión .DAT que contiene los datos y otro con una extensión .DHT que contiene otra información sobre fichero, tal como tamaño, tipo de datos y los nombres que usan para las columnas del fichero.

Si el tercer parámetro en SAVED es 0, los nombres para las columnas del grupo de datos empezarán con una X. Al usar GAUSS para análisis estadístico se usarán éstos para los nombres de sus variables. Cada fila del grupo de datos es entonces un caso u observación.

Ahora podemos conseguir algunas estadísticas sobre este grupo de datos.

```
call dstat("mydata", 0); call ols("mydata",0,0);
```

Para saber qué significan los parámetros *OLS* o *DSTAT*, presione *Alt-H* y después *H*. Le preguntará sobre qué quiere información, conteste OLS o DSTAT. Presione *Esc* para salir de la pantalla de ayuda (en este caso, DSTAT calcula estadísticas simples de los datos y OLS efectúa una regresión de la primera columna sobre las restantes).

Este tipo de ayuda sobre la marcha es también posible para sus propias funciones, como se ve a continuación.

2.4.2. *Creación de procedimientos*

Crearemos un fichero que contiene un procedimiento. Para ello editamos un fichero

```
edit sumd.g
```

Crearemos un proceso que sume las columnas del grupo de datos y que obtenga un vector de las sumas.

Las sentencias son:

```
proc sumd(name);
```

Esto es el principio de la definición de un procedimiento que actúa como una función definida por el usuario. Si no se indica otra cosa, retornará un resultado de las operaciones que se efectúen. En el ejemplo, se proporciona al procedimiento el nombre (name) de una variable que contiene los datos a analizar.

```
local fp, sum;
```

Estas variables *fp* y *sum* son variables locales. Sólo existen mientras este procedimiento actúa.

```
open fp = ^name;
```

El comando *OPEN* abre un fichero, *name* será una variable alfabética. La otra variable, *fp* va a ser usada como un apodo para manipular el contenido del fichero «name». El operador ^ en el contexto de la sentencia *OPEN*, significa que lo que sigue es una variable de caracteres y que deseamos usar su contenido.


```
if fp == -1;
errorlog "NO SE PUEDE ABRIR EL FICHERO";
end;
endif;
```

IF y *ENDIF* son operadores condicionales (si fp toma el valor menos uno, etc.). El operador de comparación lógica es == en vez de simplemente un signo de igual. Esto es un control de la accesibilidad del fichero de datos con el que queremos trabajar. Si no se puede acceder a él (porque, por ejemplo, no existe), GAUSS asigna el valor −1 a ese nombre lógico (fp en este caso). La sentencia ERRORLOG hace que se imprima en pantalla un mensaje de error (que puede visualizarse en cada momento con la tecla F7). Si el fichero no puede abrirse, el programa finaliza.

```
sum = 0;
```

Esto tiene como objetivo inicializar a cero la variable que vamos a emplear como acumulador para las sumas.

```
do until eof(fp);
```

Vamos a empezar un bucle. Nos mantendremos haciendo este bucle hasta que lleguemos al final del fichero.

```
sum = sum + sumc(readr(fp, 50));
```

La operación es la siguiente: el comando READR lee del fichero fp las primeras 50 filas de *todas* las columnas que contenga. Por ejemplo, si el fichero fp contiene una matriz de 200×200, el resultado de la anterior operación será una matriz 50×200.

A esa matriz se le aplica el operador SUMC que calcula las *sumas* de cada columna, y, por tanto, da lugar a un vector 1×200. A este vector se le suma la variable SUM (que vale cero) y obtenemos el primer valor de SUM.

A continuación, se extrae la siguiente matriz 50×200, se suman sus columnas (vector 1×200) y se le añade el valor anterior de SUM (que es el primer vector 1×200 ya calculado). Es decir, ahora tenemos un nuevo vector 1×200 que es la suma del primero y del segundo. Así se va procediendo sucesivamente hasta que al llegar al final del fichero y acabar el bucle, SUMC contiene ahora un vector 1×200 cuyos elementos son las sumas *totales* de cada columna, efectuadas de 50 en 50 elementos.

```
endo;
```

Esto es el final del bucle.

```
fp = close(fp);
```

Con esto se cierra el fichero que hemos manejado con el apodo de fp.

```
retp(sum);
```

La sentencia *RETP* se usa para salir del proceso. El vector de suma de columnas ya calculado (sum), se devuelve al proceso general.

```
endp;
```

La sentencia *ENDP* marca el final de la definición del procedimiento.

Aquí está nuestro procedimiento completo

```
proc sumd(name);
local fp, sum;
open fp = ^name;
if fp == -1;
errorlog "NO SE PUEDE ABRIR EL FICHERO";
endif;
sum = 0;
do until eof(fp);
sum = sum + sumc(readr(fp, 50));
endo;
fp = close(fp);
retp(sum);
endp;
```

Si su versión de este procedimiento es correcta, presione *Alt-X* y después *W* para salvarlo en el disco. Será salvado en el fichero *SUMD.G* porque así es como lo llamamos en el comando *EDIT.*

Como este fichero tiene una extensión *.G* será compilado automáticamente siempre que *SUMD* se pide en un programa, mientras esté situado en el directorio en vigencia o en uno de los directorios en lista en la variable de entorno GAUSSPATH.

Puede consultarse su contenido usando el comando de ayuda general (HELP) de GAUSS. Para ello, teclee Alt-H, luego H y finalmente SUMD. Recuerde que *cualquier* fichero con extensión .G será accesible de esta forma.

Para emplear el procedimiento una vez construido:

```
sum = sumd("mydata");
print sum;
```


Puesto que esto será un vector columna, si quiere la suma de todo el conjunto de datos, use:

```
sum = sumc(sumd("mydata"));
print sum;
```

Este procedimiento trabaja exactamente igual que cualquier otra función. Nótese la facilidad de encadenar comandos en una misma sentencia.

2.5. Gráficos

En este apartado comentaremos cómo obtener gráficos sencillos usando la facilidad QUICKGRAPHICS de GAUSS.

Los tipos de gráficos que vamos a comentar son diagramas cartesianos, histogramas, diagramas de barras y gráficos para series temporales.

Hay que hacer notar que QUICKGRAPHICS no tiene un comando directo para obtener el gráfico en una impresora. Puede usarse la tecla PRINTSCREEN del teclado, que permite un «vaciado de la pantalla» en la impresora.

Con objeto de obtener una impresión de buena calidad se recomienda utilizar cualquiera de los siguientes procedimientos (monitores EGA-CGA blanco y negro).

a) Editar el fichero QGRAPH.DEC en el subdirectorio SRC y poner el valor de _qdfmode = 6;
b) Antes de efectuar el gráfico teclear SETGMODE (6).

Ambas tienen el mismo efecto, el de situar la pantalla con una resolución 640×200 que es aceptada por la mayoría de las impresoras. Para más detalles, véase el comando SETGMODE en el manual de GAUSS.

Antes de efectuar ningún gráfico hay que cargar la librería de gráficos, tecleando LIBRARY QGRAPH;

2.5.1. Gráficos de series temporales

Aunque no existen explícitamente como tales, es muy sencillo generarlos usando las facilidades de la función XY. Por ejemplo, supongamos que queremos dibujar 100 datos de una variable N(0, 1) como si se hubieran recogido en momentos sucesivos del tiempo.

El procedimiento sería:

```
SERIE = RNDN (100,1);   /* genera los 100 datos N(0,1) */
TIM = SEQA (1,1,100);   /* genera una secuencia 1,100 */
_QLTYPE = 1;            /* Conecta los puntos */
XY (TIM, SERIE);        /* Dibuja el gráfico */
```

y se obtiene un gráfico como el de la figura 1.

Se le pueden añadir títulos en el eje de las X (xlabel("titulo X");) en el eje de las Y (ylabel("titulo Y");), así como un título global (title("titulo");), así como modificar las escalas.

2.5.2. *Diagramas cartesianos*

Se construyen utilizando el mismo comando XY, pero sin conectar los puntos del diagrama de dispersión. En el ejemplo siguiente se crean dos variables, una de ellas función lineal de la otra y se genera su diagrama:

```
X = RNDN(100,1); /* Se generan 100 números N(0,1) */
Z = 3*X + RNDN(100,1); /* Se genera Z */
_QSTYPE = 5; /* Se selecciona el tipo de los puntos
                del gráfico, en este caso, rombos */
XY(Z,X); /* Se genera el gráfico */
```

Con ello se obtiene un gráfico como el de la figura 2.

2.5.3. *Histogramas*

Aunque hay varios comandos gráficos para generar histogramas, aquí sólo vamos a comentar el más sencillo que necesita conocer únicamente el nombre de la variable que contiene los datos y el número de intervalos que se desea construir. El ejemplo crea un conjunto de números aleatorios y a continuación dibuja el histograma con un número de intervalos proporcional al tamaño de la muestra.

```
X = RNDN(100,1);
{c,m,f} = HIST(X,CINT(SQRT(ROWS(X))));
```

Hay varios puntos a destacar: c es un vector de dimensión igual al número de intervalos, que contiene el límite superior de cada intervalo; m contiene la marca de clase y f el número de frecuencias. El

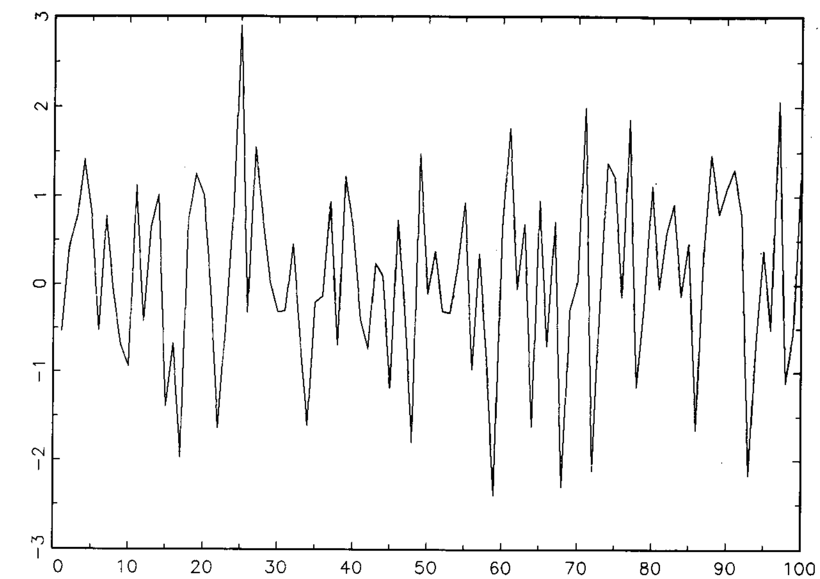

FIGURA 1. *Series temporales.*

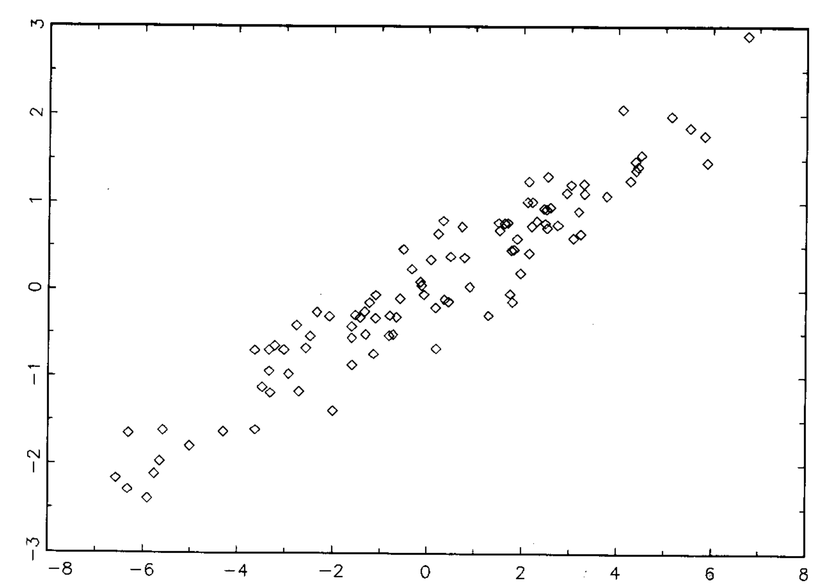

FIGURA 2. *Diagramas cartesianos.*

número de intervalos se escoge como la parte entera de la raíz cuadrada del número total de observaciones.

Con todo ello, se obtiene un gráfico como el de la figura 3.

2.5.4. *Diagramas de barras*

Se generan diagrama con barras cuya altura es el valor que toma la correspondiente observación en el fichero de datos. Como ejemplo, véase el siguiente:

```
X = RNDN(100,1);
BAR(0,X);
```

Se pueden elegir las etiquetas a colocar en las barras; si como es el caso aquí, se pone un cero (0), se crea la secuencia de 1 a ROWS (X) que se emplea como etiqueta. Véase figura 4.

2.5.5. *Gráficos múltiples*

Una interesante posibilidad de QUICKGRAPHICS es que permite subdividir la pantalla en varias partes y mostrar un gráfico diferente en una una de ellas. Usando los cuatro ejemplos anteriores, a continuación se presenta un programa que particiona la pantalla en cuatro elementos y genera los cuatro gráficos de los ejemplos previos.

El programa es:

```
GRAPHSET;          /* Inicia el programa */
BEGGRAPH;
WINDOW (2,2);  /* Particiona la pantalla en cuatro */
X = RNDN(100,1);
Y = SEQA(1,1,100);
Z = 3*X + RNDN(100,1);
_QNUM = 1;
_QAXES = 0;
XY (Y,X);                  /* Primer Gráfico */
{c,m,f} = HIST(X,20);   /* Segundo Gráfico */
_QMINOR = 4;
BAR(0,X);                  /* Tercer Gráfico */
_QLTYPE = 0;
_QSTYPE = 5;
_QMINOR = 0;
```


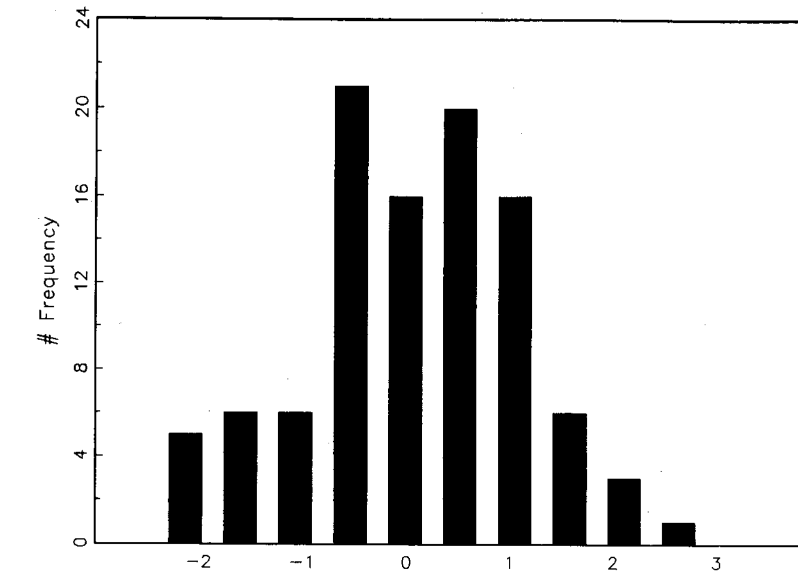

FIGURA 3. *Histograma.*

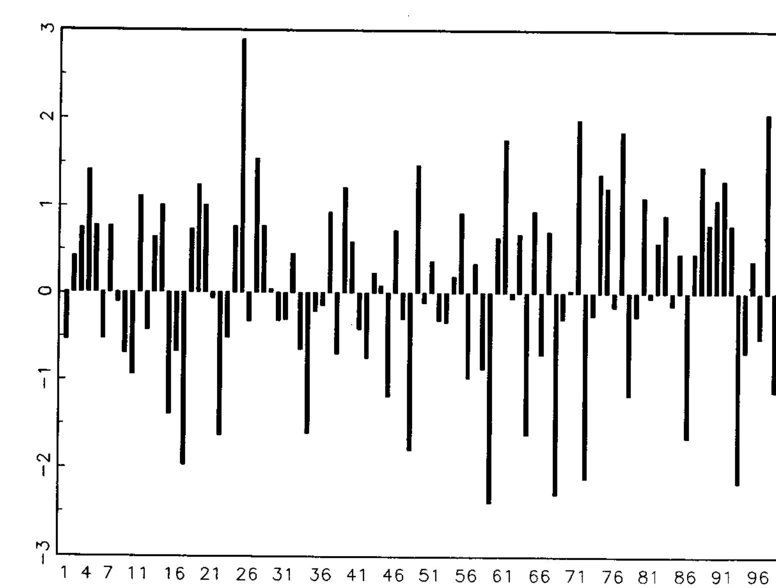

FIGURA 4. *Diagrama de barras.*

```
XY(X,Z);            /* Cuarto Gráfico */
ENDGRAPH;           /* Fin de Programa */
```

El resultado de este programa puede verse en la figura 5.

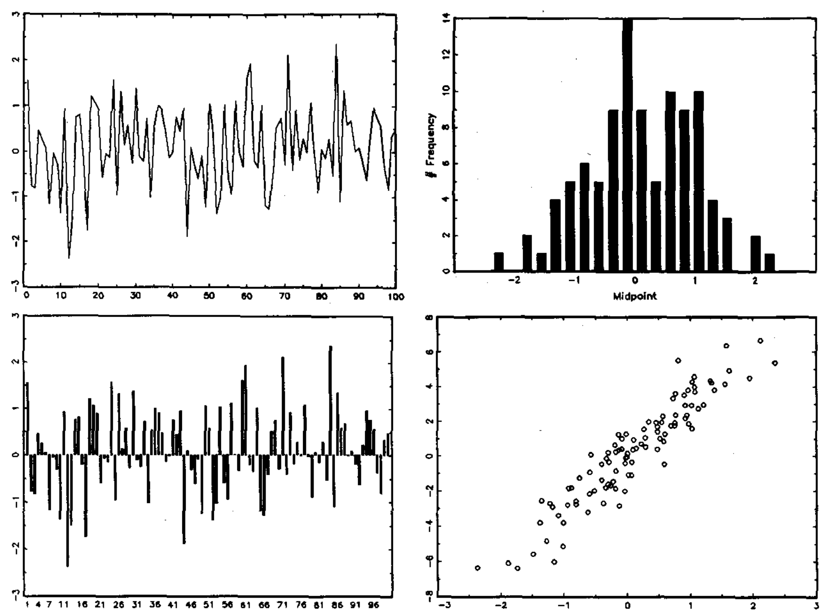

FIGURA 5. *Gráficos múltiples.*

APLICACIONES MATEMATICAS

Capítulo 3
APLICACIONES MATEMATICAS (I)

3.1. Vectores

En el conjunto $\mathbf{R}^2 = R \times R = \{(x, y) \mid x, y \in \mathbf{R}\}$, cada elemento del mismo viene dado por un par ordenado (x, y). Si se representa sobre el plano cartesiano cada par (x, y) corresponderá a un punto de dicho plano y viceversa.

El conjunto $\mathbf{R}^n = R \times R \times \ldots \times R = \{(x_1, \ldots, x_n) \mid x_1, \ldots, x_n \in \mathbf{R}\}$ cada elemento estará representado por un n-upla de números reales ordenados, $x_1, \ldots, x_n$.

Cada elemento del conjunto $\mathbf{R}^n$, representa un vector, cuyas componentes son $x_1, \ldots, x_n$; es decir, $\mathbf{x} = (x_1, \ldots, x_n)$.

Definimos como norma de un vector $x \in \mathbf{R}^n$ al número real dado por:

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

Un vector $\mathbf{x}$ se dice que es unitario cuando $\|\mathbf{x}\| = 1$.

Dados $\mathbf{u}$ y $\mathbf{v}$ dos vectores de $\mathbf{R}^n$

$$\mathbf{u} = (u_1, \ldots, u_n) \qquad \mathbf{v} = (v_1, \ldots, v_n)$$

Se define la *suma* de $\mathbf{u} + \mathbf{v}$, como el vector que se obtiene sumando las componentes correspondientes:

$$\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n)$$

Se define el *producto de un escalar* $K \in \mathbf{R}$ *por un vector* $\mathbf{u} \in \mathbf{R}^n$, como el vector que se obtiene al multiplicar las componentes de $\mathbf{u}$ por K.

$$\mathbf{u}\, K = (Ku_1, Ku_2, \ldots, Ku_n)$$

Se define el *producto escalar* de $\mathbf{u}$ por $\mathbf{v}$, y se representa por $\mathbf{u} \cdot \mathbf{v}$, como el escalar dado por:

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + \ldots + u_nv_n$$

También puede definirse como:

$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \cdot \|\mathbf{v}\| \cos \alpha$ siendo α el ángulo formado por $\mathbf{u}$ y $\mathbf{v}$.

Espacio vectorial

El conjunto $\mathbf{R}^n$, dotado de la suma de vectores, y del producto de un escalar $K \in \mathbf{R}$, por un vector, es un *espacio vettorial* sobre el cuerpo de los números reales.

Se dice que $W \subset \mathbf{R}^n$ es un *subespacio vectorial* de $\mathbf{R}^n$ si para todo $\mathbf{u}, \mathbf{v} \in W$ y $\alpha, \beta \in \mathbf{R}$, el vector $\alpha\mathbf{u} + \beta\mathbf{v} \in W$.

En general, sea E un conjunto y K un cuerpo, se dice que E tiene *estructura de espacio vectorial* sobre el cuerpo K, si están definidas:

A) Una ley de composición interna sobre E, cumpliendo las siguientes condiciones:

i) para todo $x, y, z \in E \quad (x + y) + z = x + (y + z)$
ii) para todo $x, y \in E \quad x + y = y + x$
iii) existe $0 \in E$ tal que $x + 0 = 0 + x = x$
iv) para todo $x \in E$ existe $(-x) \in E$ tal que $x + (-x) = 0$

B) Una ley de composición externa aplicando $K \times E \to E$, que a todo par $(\alpha, x) \in K \times E$ le asigna el elemento $\alpha x \in E$, cumpliendo las siguientes condiciones:

v) para todo $\alpha, \beta \in K$ y para todo $x \in E \quad \alpha(\beta x) = (\alpha\beta)x$
vi) para todo $x \in E \quad 1 \cdot x = x$
vii) para todo $\alpha, \beta \in K$ y para todo $x \in E \ (\alpha + \beta)x = \alpha x + \beta x$
viii) para todo $\alpha \in K$ y para todo $x, y \in E \ \alpha(x + y) = \alpha x + \alpha y$

Dependencia e independencia lineal

Dado el conjunto de vectores $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$ del espacio vectorial $\mathbf{R}^n$, diremos que otro vector $\mathbf{v} \in \mathbf{R}$ es *combinación lineal* de los vectores del sistema $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$ si se puede escribir

$$\mathbf{v} = \alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2 + \ldots + \alpha_n\mathbf{u}_n \qquad \alpha_1\alpha_2, \ldots, \alpha_n \in \mathbf{R}$$

Se llama *expansión lineal* del sistema de vectores $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$ de $\mathbf{R}^n$ y se representa por $L<\mathbf{u}_1, \ldots, \mathbf{u}_n>$, al conjunto formado por todas las combinaciones lineales de los vectores $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$

$$L<\mathbf{u}_1,\ldots,\mathbf{u}_n> = \{(\alpha_1\mathbf{u}_1 + \ldots + \alpha_n\mathbf{u}_n) | \alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbf{R}\}$$

La expansión lineal $L<\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ es un subespacio vectorial de $\mathbf{R}^n$, y se llama *subespacio vectorial engendrado* por $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$.

Cuando un sistema de vectores $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$ genera todo el espacio vectorial $\mathbf{R}^n$, diremos que estos vectores constituyen *un sistema generador del espacio vectorial* $\mathbf{R}^n$.

Un vector $\mathbf{v} \in \mathbf{R}^n$, se dice que es *linealmente dependiente* de los vectores $\mathbf{u}_1, \ldots, \mathbf{u}_n \in \mathbf{R}^n$, cuando se puede escribir como combinación lineal de ellos, es decir: existen $\alpha_1, \ldots, \alpha_n \in R$, tales que

$$\mathbf{v} = \alpha_1\mathbf{u} + \alpha_2\mathbf{u}_2 + \ldots + \alpha_n\mathbf{u}_n \qquad \alpha_1, \ldots, \alpha_n \in \mathbf{R}$$

Dado un sistema de vectores $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$ de $\mathbf{R}^n$, se dice que forman un sistema *linealmente dependiente* (o que los vectores son linealmente dependientes), cuando algún vector $\mathbf{u}_k \in <\mathbf{u}_1, \ldots, \mathbf{u}_n>$ puede expresarse como combinación lineal de los restantes vectores del sistema. Por el contrario, cuando *ningún* vector del sistema puede expresarse como combinación lineal de los restantes, es *linealmente independiente* o los vectores son linealmente independientes.

El vector $0 \in \mathbf{R}^n$ es *linealmente dependiente* de cualquier sistema de vectores.

Un sistema de vectores $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$ es *linealmente dependiente* cuando el vector $O \in \mathbf{R}^n$, puede escribirse como combinación lineal de ellos con algún escalar distinto de cero.

Un sistema de vectores $<\mathbf{u}_1, \ldots, \mathbf{u}_n>$ es *linealmente independiente* cuando el vector $0 \in \mathbf{R}^n$ solamente se puede escribir como combinación lineal de ellos siendo todos los escalares $\alpha_1 = \alpha_2 = \ldots = \alpha_n = 0$.

Base de un espacio vectorial

Se dice que el sistema $B = \{\mathbf{u}_1, ..., \mathbf{u}_n\}$ es una base del espacio vectorial $\mathbf{R}^n$ si:

i) B es un sistema generador de $\mathbf{R}^n$.
ii) B es un conjunto de vectores linealmente independientes.

Si $B = \{\mathbf{u}_1, ..., \mathbf{u}_n\}$ es una base de $\mathbf{R}^n$, todo vector $\mathbf{v} \in \mathbf{R}^n$ se podrá escribir como:

$$\mathbf{v} = \alpha_1\mathbf{u}_1 + ... + \alpha_n\mathbf{u}_n,$$

con $\alpha_1, ..., \alpha_n \in \mathbf{R}$ como escalares únicos. A $\alpha_1, ..., \alpha_n$ se les llama *coordenadas* del vector $\mathbf{v}$ respecto de la base B.

Por convenio, las coordenadas de un vector $\mathbf{v} \in \mathbf{R}^n$ se representan como el vector columna:

$$\mathbf{v} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

Todas las bases de un espacio vectorial tienen el mismo número de vectores. A este número se le llama *dimensión del espacio vectorial.* El máximo número de vectores linealmente independiente que generan un espacio vectorial, nos da la dimensión del mismo.

3.2. Matrices y aplicaciones lineales

Definición

Sean E y F dos espacios vectoriales, se dice que f es una *aplicación lineal* de E en F, y se representa por $f: E \rightarrow F$ si

i) para todo $\mathbf{x}, \mathbf{y} \in E \quad f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$
ii) para todo $\alpha \in \mathbf{R} \quad \mathbf{x} \in E \quad f(\alpha\mathbf{x}) = \alpha f(\mathbf{x})$

Supongamos un vector $\mathbf{a} \in \mathbf{R}^n$ fijo, $\mathbf{a} = (a_1, ..., a_n)$, a cada vector $\mathbf{x} \in \mathbf{R}^n$, el producto escalar $\mathbf{a} \cdot \mathbf{x}$ le hará corresponder un escalar

$$b = \mathbf{a}_1\mathbf{x}_1 + \mathbf{a}_2\mathbf{x}_2 + ... + \mathbf{a}_n\mathbf{x}_n.$$

Como $\mathbf{a}$ es un vector fijo, el valor de b dependerá solamente de las

componentes de $\mathbf{x}$, es decir, para cada $\mathbf{x}$, el escalar b estará definido por:

$$\mathbf{a} \cdot \mathbf{x} = [a_1, ..., a_n] \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b$$

Por tanto, podemos considerar, para una $\mathbf{a}$ fijo la aplicación:
$f_a: \mathbf{R}^n \rightarrow \mathbf{R}$ dado por:

$$f_a: \mathbf{a} \cdot \mathbf{x} = (a_{11}, ..., a_{1n}) \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = b$$

Esta aplicación así definida es una aplicación lineal. f_a nos permite escribir de una manera abreviada la ecuación:

$$a_{11}x_1 + ... + a_{1n}x_n = b \tag{1}$$

Como $f_a(x) = b$, indica que $\mathbf{x}$ es un vector cuya imagen por f es b, es decir que, $\mathbf{x} = (x_1, ..., x_n)$ es solución de la ecuación (1).

Supongamos que tenemos m-vectores $\mathbf{a}_i$ actuando sobre cada uno de los vectores $\mathbf{x} = (x_1, ..., x_n)$, siendo

$$a_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix} \qquad i = 1 ... m$$

$$\begin{array}{ll} \mathbf{a}_{1.} \cdot \mathbf{x} = b_1 & a_{11}x_1 + ... + a_{1n}x_n = b_1 \\ \mathbf{a}_{2.} \cdot \mathbf{x} = b_2 & a_{21}x_2 + ... + a_{2n}x_n = b_2 \\ \cdots\cdots & \cdots\cdots\cdots \\ \mathbf{a}_{m.} \cdot \mathbf{x} = b_m & a_{m1}x_1 + ... + a_{mn}x_n = b_m \end{array} \tag{2}$$

Cada una de las ecuaciones (2) las podemos escribir como:

$$f_{ai}(\mathbf{x}) = b_i = (a_{i1}, ..., a_{in}) \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Análogamente, $f_{ai}(\mathbf{x}) = b_i$, indica que $\mathbf{x}$ es un vector cuya imagen por

f_{ai} es b_i es a decir que $\mathbf{x} = (x_1, ..., x_n)$ es la solución del sistema de ecuaciones (2).

Los escalares $b_1, ..., b_m$ vendrán dados en un cierto orden $[b_1 b_2, ..., b_m]$ y equivaldrá a decir que son las componentes de un vector $\mathbf{b} \in \mathbf{R}^m$.

El sistema (2) se puede escribir de una forma abreviada $A\mathbf{x} = \mathbf{b}$

$$\text{Si } A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} a_{1.} \\ a_{2.} \\ \vdots \\ a_{m.} \end{bmatrix}$$

representa los m vectores $\mathbf{a}_{i.} \in \mathbf{R}^n$, siendo

$$a_{i.} = (a_{i1}, a_{i2}, ..., a_{in}) = (a_{ij}) \qquad j = 1, ..., n$$

en donde A se conoce con el nombre de matriz $(m \times n)$ que recoge ordenadamente *m* vectores de $\mathbf{R}^n$, y cada uno de ellos tiene *n* componentes. En una matriz se distinguen *m* filas (*m* vectores $\in \mathbf{R}^n$) y *n* columnas.

El doble subíndice (ij) determina unívocamente la ubicación de cada elemento de la matriz. En el caso especial en que $m = n$, es decir, n.° filas = n.° de columnas, a la matriz A se le llama matriz cuadrada. En general, si tiene *m* filas y *n* columnas, diremos que A es de dimensión $m \times n$.

Dada una matriz A de orden $m \times n$, se tiene definida de una manera unívoca una aplicación de $\mathbf{R}^n$ en $\mathbf{R}^m$ que hace que al vector $\mathbf{x} \in \mathbf{R}^n$ le asigne el vector $\mathbf{y} = A\mathbf{x} \in \mathbf{R}^m$.

Si se usa la notación habitual:

$$\bar{\mathbf{x}} \xrightarrow{f} f(\bar{\mathbf{x}}) = \mathbf{y} \quad \text{ó} \quad \mathbf{y} = f(\mathbf{x}) = A\mathbf{x}$$

siendo

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

la podemos expresar como:

$$\begin{matrix} y_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ y_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_m = a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \end{matrix} \qquad \text{ó} \qquad \mathbf{y} = A\mathbf{x}$$

siendo $x_1, ..., x_n$ las coordenadas del vector $\mathbf{x}$ respecto de la base B de $\mathbf{R}^n$, $y_1, ..., y_m$ las coordenadas de $\mathbf{y}$, respecto de la base de B' de $\mathbf{R}^m$, y la matriz A, será tal que cada columna representa, los escalares que permiten expresar las imágenes por f del vector e_j $(j = 1, ..., n)$ de la base B, como combinación lineal de los vectores u_i $(i = 1, ..., m)$ de la base B' de $\mathbf{R}^m$.

$$f(e_j) = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$$

3.3. Operaciones con matrices

Sea M(m, n) el conjunto de todas las matrices de orden $m \times n$ y consideremos $A, B \in M(m, n)$ definidas por:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \text{ y } B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

La suma de A + B es otra matriz de orden $m \times n$ dada por

$$A + B = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & \cdots & a_{2n} + b_{2n} \\ \cdots & \cdots & \cdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

Propiedades de la suma de matrices:

i) $A, B \in M(m, n)$ $\qquad A + B = B + A$
ii) $A, B, C \in M(m, n)$ $\qquad A + (B + C) = (A + B) + C$
iii) Si representamos por $0 \in M(m, n)$ la matriz cuyos elementos $a_{ij} = 0$
$i = 1, ..., n \quad j = 1, ..., m$, mediante

$$0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

para $A \in M(m, n)$ $\qquad A + 0 = 0 + A = A$.
iv) A y $B \in M(m, n)$ tales que $a_{ij} = -b_{ij}$
$A + B = 0$

El producto de un escalar $K \in \mathbf{R}$ por una matriz A de orden m×n, es otra matriz de orden m×n dada por:

$$KA = \begin{bmatrix} Ka_{11} & Ka_{12} \dots Ka_{1n} \\ Ka_{21} & Ka_{22} \dots Ka_{2n} \\ \dots\dots & \dots\dots \\ Ka_{m1} & Ka_{m2} \dots Ka_{mn} \end{bmatrix}$$

Se define la matriz $-A$ como: $-A=-1 \cdot A$. Por tanto, $A-B=A+(-B)$.

Propiedades del producto de un escalar por una matriz

i) $1 \cdot A = A$
ii) por todo $\alpha, \beta \in \mathbf{R}$ y $A \in M(m, n)$ $\alpha(\beta A) = (\alpha\beta)A$
iii) por todo $\alpha \in \mathbf{R}$ y $A, B \in M(m, n)$ $\alpha(A+B) = \alpha A+\alpha B$
iv) por todo $\alpha, \beta \in \mathbf{R}$ y $A \in M(m, n)$ $(\alpha+\beta)A = \alpha A+\beta A$

Teniendo en cuenta las propiedades anteriores el conjunto M(m, n) tiene estructura de espacio vectorial sobre **R**.

Producto de matrices

Sean A una matriz de orden m × n y B una matriz de orden n × p

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{i1} & a_{i2} & \dots & a_{in} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{1j} & \dots & b_{1p} \\ b_{21} & b_{2j} & \dots & b_{2p} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{nj} & \dots & b_{np} \end{bmatrix}$$

Se define la matriz D de orden m × p, como $D = A \cdot B$ siendo

$$D = \begin{bmatrix} d_{11} \dots\dots d_{1p} \\ \dots\dots\dots \\ d_{11} \dots d_{ij} \dots d_{ip} \\ \dots\dots\dots \\ d_{m1} \dots\dots d_{mp} \end{bmatrix}$$

donde

$$d_{ij} = \sum_{K=1}^{n} a_{iK}\, b_{Kj}$$

D será una matriz que tiene tantas columnas como la matriz B y tantas filas como la matriz A.

El producto de dos matrices no cumple la propiedad conmutativa, es decir, en general, $A \cdot B \neq B \cdot A$.

Propiedades del producto de matrices

Supongamos definidos todos los productos que aparecen seguidamente, la multiplicación de matrices cumple:

i) $(A \cdot B)C = A(B \cdot C)$
ii) $A(B + C) = AB + AC$
iii) $(B + C)A = B \cdot A + CA$
iv) $K(AB) = (KA)B = A(KB)$

Tipos especiales de matrices

La matriz de orden n × 1 es el *vector columna.*
La matriz de orden 1 × m es el *vector fila.*

Matriz unidad (I_n)

Es una matriz cuadrada de orden n de la forma:

$$I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

donde

$$a_{ij} = 1 \quad \text{si } i = j$$
$$a_{ij} = 0 \quad \text{si } i \neq j$$

Matriz escalar (E)

Es la matriz que resulta al multiplicar I_n por un escalar $\alpha \in \mathbf{R}$

$$E = \alpha I_n = \begin{bmatrix} \alpha & 0 & \dots & 0 \\ 0 & \alpha & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \alpha \end{bmatrix}$$

Matriz diagonal (D)

Es una matriz cuadrada de la forma:

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

donde

$$\begin{array}{ll} a_{ij} = 0 & i \neq j \\ a_{ii} & i = j \end{array}$$

Matriz triangular superior (T)

Es una matriz cuadrada en la que los elementos situados por debajo de la diagonal principal son cero

$$T = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

donde

$$a_{ij} = 0 \qquad i > j$$

Matriz triangular inferior (T)

Es una matriz cuadrada en la que los elementos situados por encima de la diagonal principal son cero

$$T = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

donde

$$a_{ij} = 0 \qquad i < j$$

Matriz simétrica

Es una matriz cuadrada donde $a_{ij} = a_{ji}$ para todo i, j.

Matriz antisimétrica

Es una matriz cuadrada donde $a_{ij} = -a_{ji}$ para todo i, j.
En toda matriz antisimétrica los elementos de la diagonal principal son cero.

Matriz traspuesta (A')

Dada $A \in M(m, n)$ se llama matriz traspuesta, y se representa por A', a la matriz que resulta de cambiar en A filas por columnas.
Propiedades de la matriz traspuesta:

i) $(A')' = A$
ii) $(A + B)' = A' + B'$
iii) $(\alpha A)' = \alpha A'$ para $\alpha \in \mathbf{R}$
iv) $(A \cdot B)' = B' \cdot A'$

La condición necesaria y suficiente para que una matriz sea simétrica es que $A = A'$.

Matriz inversa

Sea A una matriz cuadrada de orden n, se dice que B es la matriz inversa de A, y se representa por $B = A^{-1}$, a una matriz cuadrada de orden n, que verifica que:

$$A \cdot B = B \cdot A = I_n,$$

es decir:

$$A \cdot A^{-1} = A^{-1} \cdot A = I_n$$

Las matrices cuadradas que tienen inversa se llaman *regulares* o *no singulares*. Las demás son matrices *singulares*.

Propiedades de la matriz inversa

i) Si una matriz tiene inversa es única.
ii) $(A^{-1})^{-1} = A$
iii) Si A y B son dos matrices regulares:

$(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$

iv) Si A es regular:

a) $B \cdot A = 0$ implica que $B = 0$
b) $B \cdot A = C \cdot A$ implica que $B = C$

v) $(A^{-1})' = (A')^{-1}$

Matriz idempotente

Si A es una matriz cuadrada de orden n, se dice que es idempotente cuando:

$$A \cdot A = A$$

Matriz ortogonal

Se dice que A es ortogonal, cuando

$$A \cdot A' = I_n$$

Propiedades de la matriz ortogonal:

i) Si A ortogonal $A' = A^{-1}$.
ii) Si A ortogonal A' es también ortogonal.

Matriz nilpotente

Si A matriz cuadrada de orden n, se dice que es nilpotente de índice $p \in \mathbf{N}$ $p > 1$ si $A^{p-1} \neq 0$ y $A^p = 0$.

Propiedades de la matriz nilpotente:

i) Si A es tal que $a_{ij} = 0$ $i > j$, A es nilpotente.
ii) Si A es nilpotente de índice p, son también nilpotentes de índice p todas las matrices semejantes a A.
iii) Si A nilpotente de índice p, entonces $I_{n_1} - A$ es invertible y su inversa vale $1 + A + A^2 + \ldots + A^p$.

Nota: Dos matrices cuadradas de orden *n* A y B, se dice que son semejantes, si existe una matriz cuadrada P de orden *n* tal que:

$$B = P^{-1} \cdot A \cdot P$$

Rango de un sistema de vectores. Rango de una matriz

Se llama rango de un sistema de vectores $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ de un espacio vectorial $\mathbf{V}$ y se representa por rang $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ al mayor número de vectores linealmente independientes del sistema. Es decir, el rango de un sistema de vectores es la dimensión del subespacio vectorial que generan dicho sistema.

Un sistema de *n* vectores de un espacio de dimensión n será una base del mismo si y sólo si el rango es *n*.

Dada una matriz A $m \times n$, podemos considerarla con *n* vectores columnas del $\mathbf{R}^m$ o como m vectores filas de $\mathbf{R}^n$. El rango de los vectores fila y el rango de los vectores columna de A es el mismo.

Definición

Se llama rango de una matriz A de orden $m \times n$ y se representa por rang(A) al rango de los vectores fila o columna de la misma.

Productos de Kronecker de matrices

Definición

Sea A una matriz $m \times n$ y B $p \times q$. El producto de Kronecker de las dos matrices la matriz C

$$A \otimes B = C = \begin{bmatrix} a_{11}B & a_{12}B \ldots a_{1n}B \\ a_{21}B & a_{22}B \ldots a_{2n}B \\ \ldots\ldots & \ldots\ldots\ldots\ldots \\ a_{m1}B & a_{m2}B \ldots a_{mn}B \end{bmatrix}$$

que es una matriz de dimensión $(mp) \times (nq)$.

Cambio de base

Sea $\{e_1, \ldots, e_n\} = B$ una base de V y $(f_1, \ldots, f_n) = B'$ una base de B' de V.

$$f_1 = a_{11}e_1 + a_{12}e_2 + \dots + a_{1n}e_n$$
$$f_2 = a_{21}e_1 + a_{22}e_2 + \dots + a_{2n}e_n$$
$$\dots\dots\dots\dots\dots\dots\dots\dots$$
$$f_n = a_{n1}e_1 + a_{n2}e_2 + \dots + a_{nn}e_n$$

Sea

$$P = \begin{bmatrix} a_{11} & a_{21} \dots a_{n1} \\ a_{12} & a_{22} \dots a_{n1} \\ \dots & \dots \\ a_{1n} & a_{2n} \dots a_{nn} \end{bmatrix}$$

A P se le llama matriz de transición de B a B'. P es invertible por ser $f_1, \dots, f_n$ linealmente independientes. *P^{-1} es la matriz transición de la base* B' a B.

Teorema. Sea P la matriz de transición de la ba B a B' de V. Entonces para todo $\mathbf{v} \in \mathbf{V}$. $P[v]_{B'} = [v]_B$

$$(v)_{B'} = P^{-1}[v]_B$$

Sea f: $\mathbf{R}^n \rightarrow \mathbf{R}^m$ y sea A su matriz asociada, es decir,

$$y = A\mathbf{x}$$

en la que $\mathbf{y} \in \mathbf{R}^m$ viene dado en una base B_1 y $\mathbf{x} \in \mathbf{R}^n$ en una base B. La matriz C asociada a la aplicación f en la que $\mathbf{y}$ viene dado en una base B'_1 de $\mathbf{R}^m$ y $\mathbf{x}$ en una base B' de $\mathbf{R}^n$ será:

$$|Y|_{B'1} = C, \text{ siendo } C = Q^{-1}A.P.$$

donde Q es la matriz de transición de la base B_1 a B'_1 y P la matriz de transición de la base B a B'.

PROBLEMAS PROPUESTOS

1. Hallar el vector $\mathbf{x}$ en las ecuaciones vectoriales:

   a) $3(\mathbf{a}_1 - \mathbf{x}) + 2(\mathbf{a}_2 + \mathbf{x}) = 5(\mathbf{a}_3 + \mathbf{x})$, siendo $\mathbf{a}_1 = (2, 5, 1, 3)$, $\mathbf{a}_2 = (10, 1, 5, 10)$, $\mathbf{a}_3 = (4, 1, -1, 1)$
   b) $\mathbf{b}_1 + 3\,\mathbf{b}_2 + 4\,\mathbf{b}_3 + 2\,\mathbf{x} = 0$, siendo $\mathbf{b}_1 = (5, -8, -1, 2)$, $\mathbf{b}_2 = (2, -1, 4, -3)$, $\mathbf{b}_3 = (-3, 2, -5, 4)$
   c) $4(\mathbf{x} - \mathbf{c}_1) + 3\,\mathbf{c}_4 = 11\,\mathbf{x} - \mathbf{c}_2 + 5(\mathbf{c}_3 - \mathbf{x})$, siendo $\mathbf{c}_1 = (1, 3, 2, -1, 7)$, $\mathbf{c}_2 = (-1, 4, 5, -2, -3)$, $\mathbf{c}_3 = (4, -2, -1, 1, 0)$, $\mathbf{c}_4 = (1, 1, -4, 0, -1)$

2. Calcular:

   a) $\mathbf{v} \cdot \mathbf{w}$, siendo $\mathbf{v} = (2, 3, 1)$ y $\mathbf{w} = (1, -1, -2)$
   b) La combinación lineal $3\,\mathbf{a}_1 + 5\,\mathbf{a}_2 - \mathbf{a}_3$, siendo $\mathbf{a}_1 = 2\,\mathbf{v} + \mathbf{w}$, $\mathbf{a}_2 = \mathbf{v} - \mathbf{w}$, $\mathbf{a}_3 = (\mathbf{v} \cdot \mathbf{w}) \cdot \mathbf{w}$.

3. Dados los vectores: $\mathbf{a}_1 = (3, 6 -2)$, $\mathbf{a}_2 = (6, -4, -3)$, $\mathbf{a}_3 = (3, 2, 1)$, $\mathbf{a}_4 = (1, -1, -1)$:

   a) Indicar cuáles son ortogonales entre sí.
   b) Resolver $\mathbf{a}_1 - 1/2\,\mathbf{a}_2 + \mathbf{a}_3 - 2\,\mathbf{x} = 0$.

4. a) Escribir el vector $v = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$ como combinación lineal de

   $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

b) Escribir el vector $v = \begin{bmatrix} 1 \\ -1 \\ 4 \end{bmatrix}$ como combinación lineal de

$$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

c) Escribir el vector $v = \begin{bmatrix} 14 \\ 9 \\ 5 \\ 11 \end{bmatrix}$ como combinación lineal de

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 4 \\ 0 \\ -1 \\ 3 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 5 \\ 6 \\ 7 \end{bmatrix}$$

5. Comprobar si son o no linealmente independientes los siguientes vectores:

a) $\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ b) $\begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ c) $\begin{bmatrix} 3 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

d) $\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ e) $\begin{bmatrix} 4 \\ -5 \\ 2 \\ 6 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \\ 1 \\ 3 \end{bmatrix} \begin{bmatrix} 6 \\ -3 \\ 3 \\ 9 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ 5 \\ 6 \end{bmatrix}$

f) $\begin{bmatrix} 1 \\ 3 \\ 2 \\ 5 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ -1 \\ 2 \end{bmatrix} \begin{bmatrix} 7 \\ -1 \\ 0 \\ 2 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 3 \\ 0 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix}$

6. Comprobar si los siguientes vectores forman una base de:

a) $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ de $\mathbf{R}^2$

b) $\begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 7 \end{bmatrix}$ de $\mathbf{R}^3$

c) $\begin{bmatrix} -1 \\ 0 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \\ -3 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 0 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \\ -3 \end{bmatrix}$ de $\mathbf{R}^4$

7. Estudiar la dependencia o independencia lineal de las siguientes matrices:

a) $\begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} -2 & 3 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} 5 & 3 \\ 4 & -1 \end{bmatrix}$

b) $\begin{bmatrix} 1 & -2 & 4 \\ 3 & 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & -4 & 8 \\ 6 & 0 & 2 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$

d) $\begin{bmatrix} 1 & 2 & -3 \\ 6 & -5 & 4 \end{bmatrix} \begin{bmatrix} 6 & -3 & 2 \\ 1 & 2 & -3 \end{bmatrix}$

e) $\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 2 & 2 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & -5 \\ -4 & 0 \\ 3 & 2 \end{bmatrix}$

8. Escribir la matriz E, como combinación lineal de:

$A = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}$ $C = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$

a) $E = \begin{bmatrix} 3 & -1 \\ 1 & -2 \end{bmatrix}$ b) $E = \begin{bmatrix} 0 & -4 \\ 5 & -3 \end{bmatrix}$

9. Sea W el subespacio de $\mathbf{R}^4$ generado por los vectores

$$\begin{bmatrix} 1 \\ -2 \\ 5 \\ -3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \\ -4 \end{bmatrix} \begin{bmatrix} 3 \\ 8 \\ -3 \\ -5 \end{bmatrix}$$

a) Hallar una base y la dimensión de W.
b) Extender la base de W a una base total de $\mathbf{R}^4$.

10. Hallar las dimensiones del espacio vectorial generado por:

a) $\begin{bmatrix} 1 \\ -2 \\ 3 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -2 \\ 3 \end{bmatrix}$ b) $\begin{bmatrix} 3 \\ -6 \\ 3 \\ 9 \end{bmatrix} \begin{bmatrix} -2 \\ 4 \\ -2 \\ 6 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$ d) $\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} -3 & -3 \\ 3 & 3 \end{bmatrix}$

11. Hallar las coordenadas del vector V relativas al base

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ de } \mathbf{R}^3$$

a) $V = \begin{bmatrix} 4 \\ -3 \\ 2 \end{bmatrix}$ b) $V = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$

12. Sea el espacio vectorial M(2,2), hallar el vector coordenado de la matriz A relativo a la base:

a) $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} A = \begin{bmatrix} 2 & 3 \\ 4 & -7 \end{bmatrix}$

b) $\begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 4 & -1 \\ -1 & -5 \end{bmatrix} A = \begin{bmatrix} 4 & -11 \\ -11 & -7 \end{bmatrix}$

13. Hallar el rango de:

a) $\begin{bmatrix} 1 & 3 & -2 & 5 & 4 \\ 1 & 4 & 1 & 3 & 5 \\ 1 & 4 & 2 & 4 & 3 \\ 2 & 7 & -3 & 6 & 13 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 2 & -3 & -2 & -3 \\ 1 & 3 & -2 & 0 & -4 \\ 3 & 8 & -7 & -2 & -11 \\ 2 & 1 & -9 & -10 & -3 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 & 2 \\ 4 & 5 & 5 \\ 5 & 8 & 1 \\ -1 & -2 & 2 \end{bmatrix}$ d) $\begin{bmatrix} 2 & 1 \\ 3 & -7 \\ -6 & 1 \\ 5 & -8 \end{bmatrix}$

14. Sean A, B $\in$ M(2, 2), hallar ejemplos tales que:
    a) rang (A + B) < rang (A), rang (B)
    b) rang (A + B) = rang (A) = rang (B)
    c) rang (A + B) > rang (A), rang (B)

15. Una aplicación lineal f: $\mathbf{R}^k \to \mathbf{R}^n$ tiene por matriz asociada:

a) $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ b) $[2 \quad 1]$ c) $\begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 3 & 2 \end{bmatrix}$

determinar las dimensiones k y n de los espacios de partida y llegada respectivamente y la imagen de un vector cualquiera v $\in$ $\mathbf{R}^k$ por f.

16. Sea f: $\mathbf{R}^3 \to \mathbf{R}^2$, $\mathbf{v} \to f(\mathbf{v}) = (2x + y, z)$ para todo $\mathbf{v} \in \mathbf{R}^3$ de componentes (x, y, z). Hallar la matriz asociada a f, si en $\mathbf{R}^3$ se considera la base:

    B = {(1, 0, 1), (1, 1, 0), (0, 1, 1)} y en $\mathbf{R}^2$ se toma B' = {(1, 1), (2, 0)}

17. Sea f: $\mathbf{R}^4 \to \mathbf{R}^3$ definida por:

    $f(x_1, x_2, x_4) = (x_1 - x_2 + x_3 + x_4, x_1 + 2x_3 - x_4, x_1 + x_2 + 3x_3 - 3x_4)$

    a) Hallar por f la imagen de la base canónica de **R4**.
    b) Determinar una base y dimensión del conjunto imagen U por f.

18. Sea f: $\mathbf{R}^3 \to \mathbf{R}^3$ definida por:

    f(x, y, z) = (x + z, 2x + y + z, −x + y − 2z). Hallar:

    a) Una base del conjunto imagen U por f.
    b) Dimensión de U.

19. Igual enunciado problema 18 siendo:

    f: $\mathbf{R}^3 \to \mathbf{R}^2$ definida por f(x, y, z) = (x + y, y + z)

20. Sea f: $\mathbf{R}^3 \to \mathbf{R}^2$ dada por f(x, y, z) = (x + 2y − 4z, x − 5y + 3z)

    Hallar la matriz de la aplicación lineal cuando se considera las bases de $\mathbf{R}^3$ y $\mathbf{R}^2$ siguientes:

    B = {(1, 1, 1), (1, 1, 0), (1, 0, 0)}
    B' = {(1, 3), (2, 5)}

21. Dada la aplicación lineal f: $\mathbf{R}^2 \to \mathbf{R}^2$ definida por f(x, y)=(y, 3x−y) y sean las bases de $\mathbf{R}^2$

    B = {(1, 0), (0, 1)} y B' = {(1, 3), (2, 5)}.

    Hallar:

    a) La matriz de transición P, de la base canónica a la base B'.
    b) La matriz de transición de B' a la base canónica.
    c) Calcular la matriz C asociada a la aplicación lineal cuando B y B' son respectivamente las bases del espacio de partida y llegada.

22. Sea f: $\mathbf{R}^3 \rightarrow \mathbf{R}^2$ definida por:

$f(x, y, z) = (2x - y + z, 3x - y + 2z)$, hallar la matriz asociada a f para las bases de $\mathbf{R}^3$ y $\mathbf{R}^2$, respectivamente dadas por:

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad B' = \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}$$

23. Calcular la matriz C asociada a la aplicación lineal f: $\mathbf{R}^2 \rightarrow \mathbf{R}^3$ dada por $f(x, y) = (x + 2y, 3x - y, x)$ siendo

$$B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B' = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{las bases de } \mathbf{R}^2 \text{ y } \mathbf{R}^3$$

24. Calcular la matriz asociada a cada una de las siguientes aplicaciones lineales

a) $f(x, y) = (3x - y, 2x + 4y, 5x - 6y)$
b) $f(x_1,x_2,x_3,x_4)=(3x_1-4x_2+2x_3-5x_4, 5x_1+7x_2-x_3+2x_4)$
c) $f(x, y, z) = (2x + 3y - 8z, x + y + z, 4x - 5y + 6z)$

25. Calcular la inversa de las siguientes matrices:

$$a) \begin{bmatrix} 1 & 2 & 3 \\ 4 & -2 & 3 \\ 2 & 5 & -1 \end{bmatrix} \quad b) \begin{bmatrix} 2 & 0 & 1 \\ 4 & 2 & -3 \\ 5 & 3 & 1 \end{bmatrix}$$

$$c) \begin{bmatrix} 2 & 5 & -3 & -2 \\ -2 & -3 & 2 & -5 \\ 1 & 3 & -2 & -2 \\ -1 & -6 & 4 & 3 \end{bmatrix} \quad d) \begin{bmatrix} -11 & 2 & 2 \\ -4 & 0 & 1 \\ 6 & -1 & -1 \end{bmatrix} \quad e) \begin{bmatrix} 3 & 2 & -4 \\ 1 & 0 & 2 \\ -2 & 3 & 3 \end{bmatrix}$$

26. Dadas

$$A = \begin{bmatrix} 1 & -4 & 2 \\ -1 & 4 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 \\ -1 & 3 \\ 5 & 2 \end{bmatrix} \quad C = \begin{bmatrix} 2 & 2 \\ 1 & -1 \\ 1 & -3 \end{bmatrix}$$

calcular:

a) $B + C$, b) $A \cdot B$, c) $B \cdot A$, d) $A \cdot C$, e) $B' \cdot C$, f) $A(2B - 3C)$, g) $A + 5C'$, h) producto de Kronecker de A por B, i) producto de Kronecker de Kronecker de C por B.

FORMULACION DE LOS PROBLEMAS EN GAUSS

PROBLEMA 1

```
Apartado 1a */;

    output file = d:tema1.p1 on;
    a1 = {2, 5, 1, 3};
    a2 = {10, 1, 5, 10};
    a3 = { 4, 1, −1, 1 };
    s1a = (3*a1+2*a2−5*a3) / 6;
    print "solución apartado 1a";
    print s1a;

/* Apartado 1b */;

    b1 = {5, −8, −1, 2};
    b2 = {2, −1, 4, −3};
    b3 = {−3, 2, −5, 4};
    s1b = (b1+3*b2+4*b3) / −2;
    print "solución apartado 1b";
    print s1b;

/* Apartado 1c */;

    c1 = {1, 3, 2, −1, 7};
    c2 = {−1, 4, 5, −2, −3};
    c3 = {4, −2, −1, 1, 0};
    c4 = {1, 1, −4, 0, −1};
    s1c = (−4*c1+c2−5*c3+3*c4) / 2;
    print "solución apartado 1c";
    print s1c;
    output file = d:tema1.p1 off
```

```
/* PROBLEMA 2

    Apartado 2a */;

        output file = d:tema1.p2 on;

        v = {2, 3, 1};
        w = {1, −1, −2};
        s2a = v'*w;
        print "solución apartado 2a";
        print s2a;

    /* Apartado 2b */;

        a1 = 2*v+w;
        a2 = v−w;
        a3 = (v'*w)*w;
        s2b = 3*a1+5*a2−a3;
        print "solución apartado 2b";
        print s2b;
        output file = d:tema1.p2 off;


/* PROBLEMA 3

    Apartado 3a */;

        output file = d:tema1.p3 on;

        a1 = {3, 6, −2};
        a2 = {6, −4, −3};
        a3 = {3, 2, 1};
        a4 = {1, −1, −1};
        s3a1 = a1'*a2;
        s3a2 = a1'*a3;
        s3a3 = a1'*a4;
        s3a4 = a2'*a3;
        s3a5 = a2'*a4;
        s3a6 = a3'*a4;
        print "solución apartado 3a";
        print s3a1;
        print s3a2;
        print s3a3;
        print s3a4;
        print s3a5;
        print s3a6;
```



```
    /* Apartado 3b */;

        s3b = (2*a1−a2+2*a3) / 4;
        print "solución apartado 3b";
        print s3b;
        output file = tema1.p3 off;


/* PROBLEMA 4

    Apartado 4a */;

        output file = d:tema1.p4 on;

        v = {1, 3};
        u1 = {1, 1};
        u2 = {1, 0};
        A = {1 1, 1 0};
        /* v = A * X, X = inv(A) * v */
        s4a = inv(A)*v;
        print "solución apartado 4a";
        print s4a;

    /* Apartado 4b */;

        v = {1, −1, 4};
        u1 = {1, 0, 1};
        u2 = {0, 1, 1};
        u3 = {1, 1, 1};
        A = {1 0 1, 0 1 1, 1 1 1};
        s4b = inv(A)*v;
        print "solución apartado 4b";
        print s4b;

    /* Apartado 4c */;

        v = {14, 9, 5, 11};
        A = {1 3 4 3, 0 2 0 5, 0 0 −1 6, 1 0 3 7};
        s4c = inv(A)*v;
        print "solución apartado 4c";
        print s4c;
        output file = d:tema1.p4 off;
```

```
/* PROBLEMA 5

    Apartado 5a */;

        output file = d:tema1.p5 on;
        u1 = {0, 1};
        u2 = {-1, 1};
        u3 = {1, 2};
        A = {0 -1 1, 1 2 1};
        s5a = rank(A);
        print "solución apartado 4a";
        print s5a;

    /* Apartado 5b */

        A = {0 3, 2 1};
s5b = rank(A)
print "solución apartado 5b";
print s5b;

    /* Apartado 5c */;

        A = {3 1 0, -1 1 0};
        s5c = rank(A);
        print "solución apartado 5c1";
        print s5c;

    /* Apartado 5d */;

        A = {1 3 1, 2 1 0, 1 1 -1};
        s5d = rank(A);
        print "solución apartado 5d";
        print s5d;

    /* Apartado 5e */;

        A = {4 2 6 4, -5 -2 -3 -1, 2 1 3 5, 6 3 9 6};
        s5e = rank(A);
        print "solución apartado 5e";
        print s5e;

    /* Apartado 5f */;

        A = {1 4 7 4 5, 3 3 -1 2 2, 2 -1 0 3 -1, 5 2 2 0 3};
        s5f = rank(A);
        print "solución apartado 5f";
        print s5f;
        output file ; d:tema1.p5 off;
```



```
/* PROBLEMA 6

    Apartado 6a */;

        output file = d:tema1.p6 on;

        /* Como la dimensión de R2 es 2, si los vectores son
        linealmente independientes, entonces constituyen una
        base */;
        A = {1 3, 2 1};
        s6a = rank(A);
        print "solución apartado 6a";
        print s6a;

    /* Apartado 6b */;

        /* dim(R3) = 3 */;
        a = {2 3 4, 1 2 3, 1 1 7};
        s6b = rank(A);
        print "solución apartado 6b";
        print s6b;

    /* Apartado 6c */;

        /* dim(R4) = 4 */;
        A = {-1 3 4 0, 0 2 1 -1, 2 1 0 0, 3 -3 2 -3};
        s6c = rank(A);
        print "solución apartado 6c";
        print s6c;
        output file = d:tema1.p6 off;


/* PROBLEMA 7

    Apartado 7a */

        output file = d:tema1.p7 on;
        /* A = {3 -1, 2 1}, B = {-2 3, -1 -2},
        C = {5 3, 4 -1},
        A, B, C, pertenecen al espacio vectorial M(2, 2).
        Sea u1 = {1 0, 0 0}, u2 = {0 1, 0 0}, u3 = {0 0, 1 0},
        u4 = {0 0, 0 1}, la base canónica U del espacio
        vectorial M(2, 2).

        Las coordenadas de A, B, C, respecto de U, serán
        respectivamente:
```

```
     (3 −1 2 1), (−2 3 −1 −2), (5 3 4 −1). */;
     X = {3 −1 2 1, −2 3 −1 −2, 5 3 4 −1};
     s7a = rank(X);
     print "solución apartado 7a";
     print s7a;

  /* Apartado 7b */;

     /* Razonando análogamente al apartado 7a */;
     X = {1 −2 4 3 0 −1, 2 −4 8 6 0 2};
     s7b = rank(X);
     print "solución apartado 7b";
     print s7b;

  /* Apartado 7c */;

     X = {1 1 1 1, 1 0 0 1, 1 1 0 0};
     s7c = rank(X);
     print "solución apartado 7c";
     print s7c;

  /*Apartado 7d */;

     X = {1 2 −3 6 −5 4, 6 −3 2 1 2 −3};
     s7d = rank(X);
     print "solución apartado 7d";
     print s7d;

  /* Apartado 7e */;

     X = {1 2 3 1 2 0, 3 −1 2 2 4 1, 1 −5 −4 0 3 2};
     s7e = rank(X);
     print "solución apartado 7e";
     print s7e;
     output file = D:tema1.p7 off;


/* PROBLEMA 8

  Apartado 8a */;

     output file = d:tema1.p8 on;
     /* Teniendo en cuenta el razonamiento expuesto
     en 7a y siendo:
     A = {1 1, 0 −1}; B = {1 1, −1 0 }; C = {1 −1, 0 0};
```



```
     E = {3 −1, 1 −2}*/;
     X = {1 1 1, 1 1 −1, 0 −1 0, −1 0 0};
     Y = {3, −1, 1, −2};
     s8a = Y / X;
     print "solución apartado 8a";
     print s8a;

  /* Apartado 8 b*/;

     X = {1 1 1, 1 1 −1, 0 −1 0, −1 0 0};
     Y = {0, −4, 5, −3};
     s8b = Y / X;
     print "solución apartado 8b";
     print s8b;
     output file = d:tema1.p8 off;


/* PROBLEMA 9

  Apartado 9a */;

     output file = d:tema1.p9 on;
     A = {1 2 3, −2 3 8, 5 1 −3, −3 −4 −5};
     s9a1 = rank(A);
     B = {1 −2 5 −3, 2 3 1 −4};
     s9a2 = rank(B);
     print "solución apartado 9a";
     print s9a1;
     print s9a2;

  /* Apartado 9b */;

     A = {1 −2 5 −3, 2 3 1 −4, 0 0 1 0, 0 0 0 1};
     s9b = rank(A);
     print "solución apartado 9b";
     print s9b;
     output file = d:tema1.p9 off;


/* PROBLEMA 10

  Apartado 10a */;

     output file = d:tema1. p10 on;
```

```
A = {1 1, −2 1, 3 −2, −1 3};
s10a = rank(A);
print "solución apartado 10a";
print s10a;

/* Apartado 10b */;

A = {3 −2, −6 4, 3 −2, 9 6};
s10b = rank(A);
print "solución apartado 10b";
print s10b;

/* Apartado 10c */;

A = {1 2 1 2, 1 1 2 2};
s10c = rank(A);
print "solución apartado 10c";
print s10c;

/* Apartado 10d */;

A = {1 1 −1, −1, −3 −3 3 3};
s10d = rank(A);
print "solución 10d";
print s10d;
output file = d:tema1.p10 off;


/* PROBLEMA 11

Apartado 11a */;

output file = d:tema1.p11 on;
A = {1 1 1, 1 1 0, 1 0 0};
v = {4, −3, 2};
s11a = v/A;
print "solución apartado 11a";
print s11a;

/* Apartado 11b */;

A = {1 1 1, 1 1 0, 1 0 0};
v = {3, 2, 1};
s11b = v/A;
print "solución apartado 11b";
print s11a;
output file = d:tema1.p11 off;
```



```
/* PROBLEMA 12

Apartado 12a */

output file = d:tema1.p12 on;
X = {1 0 1 1, 1 −1 −1 0, 1 1 0 0, 1 0 0 0};
A = {2, 3, 4, −7};
s12a = A/X;
print "solución apartado 12a";
print s12a;

/* Apartado 12b */;

X = {1 2 4, −2 1 −1, −2 1 −1, 1 3 −5};
A = {4, −11, −11, −7};
s12b = A/X;
print "solución apartado 12b";
print s12b;
output file = d:tema1.p12 off;


/* PROBLEMA 13

Apartado 13a */;

output file = tema1.p13 on;
A = {1 3 −2 5 4, 1 4 1 3 5, 1 4 2 4 3, 2 7 −3 6 13};
s13a = rank(A);
print "solución apartado 13a";
print s13a;

/* Apartado 13b */;

A = {1 2 −3 −2 −3, 1 3 −2 0 −4, 3 8 −7 −2 −11,
2 1 −9 −10 −3};
s13b = rank(A);
print "solución apartado 13b"; print s13b;

/* Apartado 13c */;

A = {1 1 2, 4 5 5, 5 8 1, −1 −2 2};
s13c = rank(A);
print "solución apartado 13c";
print s13c;
```

```
/* Apartado 13d */;

    A = {2 1, 3 −7, −6 1, 5 −8};
    s13d = rank(A);
    print "solución apartado 13d";
    print s13d;
    output file = d:tema1. p13 off;
```

```
/* PROBLEMA 14

  Apartado 14a */;

    output file = d:tema1.p14 on;
    A = {1 0, 1 0};
    B = {−1 0, −1 0};
    /* rang(A) = 1 rang(B) = 1 */;
    C = A+B;
    s14a = rank(C);
    print "solución apartado 14a";
    print s14a;

  /* Apartado 14b */;

    A = {1 0, 0 0};
    B = {0 0, 2 0};
    /* rang(A) = 1 rang (B) = 1 */;
    C = A+B;
    s14b = rank(C);
    print "solución apartado 14b";
    print s14b;

  /* Apartado 14c */;

    A = {1 0, 0 0};
    B = {0 0, 0 1};
    /* rang(A) = 1 rang(B) = 1 */;
    C = A+B;
    s14c = rank(C);
    print "solución apartado 14c";
    print s14c;
    output file = d:tema1. p14 off;
```



```
/* PROBLEMA 15

  Apartado 15a */;

    output file = d:tema1.p15 on;
    /* dimensión espacio partida 3, dimensión espacio
    llegada 2 */;
    A = {1 0 1, 1 1 1};
    v = {2, 4, 5};
    s15a = A*v;
    print "solución apartado 15a";
    print s15a;

  /* Apartado 15b */;

    /* dimensión espacio partida 2, dimensión espacio
    llegada 1 */;
    b = {2 1};
    v = {3, 6};
    s15b = b*v;
    print "solución apartado 15b";
    print s15b;

  /* Apartado 15c */;

    /* dimensión espacio partida 2, dimensión espacio
    llegada 3 */;
    c = {1 1, −1 0, 3 2};
    v = {2, 5};
    s15 c = c*v;
    print "solución apartado 15c";
    print s15c;
    output file = d:tema1.p15 off;
```

```
/* PROBLEMA 16 */;

    output file = d:tema1.p16 on;
    / A matriz de la aplicación
    Q matriz de transición de la base canónica de
    R3 a la base B, P matriz de transición de la base
    canónica de R2 a la base B' */;
    A = {2 1 0, 0 0 1};
    Q = {1 1 0, 0 1 1, 1 0 1};
    P = {1 2, 1 0};
```

```
s16 = inv(P)*A*Q;
print "solución 16";
print s16;
output file = d:tema1.p16 off;


/* PROBLEMA 17

   Apartado 17a */;

      output file = d:tema1.p17 on;
      /* A matriz aplicación lineal
      a1 imagen de e1 = {1, 0, 0, 0},
      a2 imagen de e2 = {0, 1, 0, 0},
      a3 imagen de e3 = {0, 0, 1, 0},
      a4 imagen de e4 = {0, 0, 0, 1} */;
      A = {1 -1 1 1, 1 0 2 -1, 1 1 3 -3};
      e1 = {1, 0, 0, 0};
      e2 = {0, 1, 0, 0};
      e3 = {0, 0, 1, 0};
      e4 = {0, 0, 0, 1};
      s17a1 = A*e1;
      s17a2 = A*e2;
      s17a3 = A*e3;
      s17a4 = A*e4;
      print "solución apartado 17a";
      print s17a1;
      print s17a2;
      print s17a3;
      print s17a4;

   /* Apartado 17b */;

      B = {1 -1 1 1, 1 0 2 -1, 1 1 3 -3};
      s17b = rank(B);
      print "solución apartado 17b";
      print s17b;
      output file = d:tema1.p17 off;


/* PROBLEMA 18

   Apartado 18a */;

      output file = d:tema1.p18 on;
```



```
      /* las imágenes de los generaciones de R3 generan
      la imagen de U, A matriz de la aplicación lineal */;
      e1 = {1, 0, 0};
      e2 = {0, 1, 0};
      e3 = {0, 0, 1};
      A = {1 0 1, 2 1 1, -1 1 -2};
      s18a1 = A*e1;
      s18a2 = A*e2;
      s18a3 = A*e3;
      print "solución apartado 18a";
      print s18a1;
      print s18a2;
      print s18a3;

   /* Apartado 18b */;

      B = {1 0 1, 2 1 1, -1 1 -2};
      s18b = rank(B);
      print "solución apartado 18b";
      print s18b;
      output file = d:tema1.p18 off;


/* PROBLEMA 19

   Apartado 19a */;

      output file = d:tema1.p19 on;
      e1 = {1, 0, 1};
      e2 = {0, 1, 0};
      e3 = {0, 0, 1};
      A = {1 1 0, 0 1 1};
      s19a1 = A*e1;
      s19a2 = A*e2;
      s19a3 = A*e3;
      print "solución apartado 19a";
      print s19a1;
      print s19a3;

   /* Apartado 19b */;

      B = {1 1 0, 0 1 1};
      s19b = rank(B);
      print "solución apartado 19b";
      print s19b;
      output file = d:tema1.p19;
```

```
/* PROBLEMA 20 */;

        output file = d:tema1.p20 on;
        /* A matriz aplicación
        Q matriz transición de la base canónica de
        R3 a la base B
        P matriz de transición de la base canónica de
        R2 a la base B' */;
        A = {1 2 -4, 1 -5 3};
        Q = {1 1 1, 1 1 0, 1 0 0};
        P = {1 2, 3 5};
        s20 = inv(P)*A*Q;
        print "solución p20";
        print s20;
        output file = d:tema1.p20 off;

/* PROBLEMA 21

    Apartado 21a */;

        output file = d:tema1.p21a on;
        s21a = {1 2, 3 5};
        print "solución apartado 21a";
        print s21a;

    /* Apartado 21b */;

        s21b = inv(s21a);
        print "solución apartado 21b;
        print s21b;

    /* Apartado 21c */;

        A = {0 1, 3 -1};
        Q = {1 0, 0 1};
        P = {1 2, 3 5};
        s21c = inv(P)*A*Q;
        print "solución apartado 21c";
        print s21c;
        output file = d:tema1.p21 off;
```



```
/* PROBLEMA 22 */;

        output file = d:tema1.p22 on;
        A = {2 -1 1, 3 -1 2};
        Q = {1 2 3, 0 1 0, 1 0 0};
        P = {2 4, 3 1};
        s22 = inv(P)*A*Q;
        print "solución p22";
        print s22;
        output file = d:tema1.p22 off;

/* PROBLEMA 23 */;

        output file = d:tema1.p23 on;
        A = {1 2, 3 -1, 1 0};
        Q = {1 2, 3 4};
        P = {1 1 1, 1 1 0, 1 0 0};
        s23 = inv(P)*A*Q;
        print "solución p23";
        print s23;
        output file = d:tema1.p23 off;

/* PROBLEMA 24

    Apartado 24a */;

        output file = d:tema1.p24 on;
        /* dimensión espacio partida 2, dimensión espacio
        llegada 3 */;
        s24a = {3 -1, 2 4, 5 -6};
        print "solución apartado 24a";
        print s24a;

    /* Apartado 24b */;

        /* dimensión espacio partida 4, dimensión espacio
        llegada 2 */;
        s24b = {3 -4 2 -5, 5 7 -1 2};
        print "solución apartado 24b";
        print 24b;
```

```
/* Apartado 24c */;

    /* dimensión espacio partida 3, dimensión espacio
    llegada 3 */;
    s24c = {2 3 −8, 1 1 1, 4 −5 6};
    print "solución apartado 24c";
    print s24c;
    output file = d:tema1.p24 off; */;

/* PROBLEMA 25

  Apartado 25a */;

    output file = d:tema1.p25 on;
    X = {1 2 3, 4 −2 3, 2 5 −1};
    s25a = inv(X);
    print "solución apartado 25a";
    print s25a;

  /* Apartado 25b */;

    X = {2 0 1, 4 2 −3, 5 3 1};
    s25b = inv(X);
    print "solución apartado 25b";
    print s25b;

  /* Apartado 25c */;

    X = {2 5 −3 −2, −2 −3 2 −5, 1 3 −2 −2, −1 −6 4 3};
    s25c = inv(X);
    print "solución apartado 25c";
    print s25c;

  /* Apartado 25d */;

    X = {−11 2 2, −4 0 1, 6 −1 −1};
    s25d = inv(X);
    print "solución apartado 25d";
    print s25d;

  /* Apartado 25e */;

    X = {3 2 −4, 1 0 2, −2 3 3};
    s25e = inv(X);
    print "solución apartado 25e";
    print s25e;
    output file = d:tema1.p25 off;
```



```
/* PROBLEMA 26

  Apartado 26a */;

    output file = d:tema1.p26 on;
    B = {1 2, −1 3, 5 2};
    C = {2 2, 1 −1, 1 −3};
    s26a = B+C;
    print "solución apartado 26a";
    print s26a;

  /* Apartado 26b */;

    A = {1 −4 2, −1 4 −2};
    B = {1 2, −1 3, 5 2};
    s26b = A*B;
    print "solución apartado 26b";
    print s26b;

  /* Apartado 26c */;

    A = {1 −4 2, −1 4 −2};
    B = {1 2, −1 3, 5 2};
    s26c = B*A;
    print "solución apartado 26c";
    print s26c;

  /* Apartado 26d */;

    A = {1 −4 2, −1 4 −2};
    C = {2 2, 1 −1, 1 −3};
    s26d = A*C;
    print "solución apartado 26d";
    print s26d;

  /* Apartado 26e */;

    B = {1 2, −1 3, 5 2};
    C = {2 2, 1 −1, 1 −3};
    s26e = B'*C;
    print "solución apartado 26e";
    print s26e;

  /* Apartado 26f */;

    A = {1 −4 2, −1 4 −2};
    B = {1 2, −1 3, 5 2};
    C = {2 2, 1 −1, 1 −3};
```

```
    s26f = A*(2*b-3*C);
    print "solución apartado 26f";
    print s26f;

/* Apartado 26g */;

    A = {1 -4 2, -1 4 -2};
    C = {2 2, 1 -1, 1 -3};
    s26g = A+5*c';
    print "solución apartado 26g";
    print s26g;

/* Apartado 26h */;

    A = {1 -4 2, -1 4 -2};
    B = {1 2, -1 3, 5 -2};
    s26h = B .*. A;
    print "solución apartado 26h";
    print s26h;

/* Apartado 26i */;

    B = {1 2, -1 3, 5 -2};
    C = {2 2, 1 -1, 1 -3};
    s26i = C .*. B';
    print "solución apartado 26i";
    print s26i;
    output file = d:tema1.p26 off;
```


SOLUCIONES A LOS PROBLEMAS

PROBLEMA 1

```
solución apartado 1a

    1.0000000
    2.0000000
    3.0000000
    4.0000000


solución apartado 1b

    0.50000000
    1.5000000
    4.5000000
   -4.5000000


solución apartado 1c

   -11.000000
    2.5000000
   -5.0000000
   -1.5000000
   -17.000000
```

PROBLEMA 2

solución apartado 2a

−3.0000000

solución apartado 2b

23.000000
32.000000
9.0000000

PROBLEMA 3

solución apartado 3a

0.00000000
19.000000
−1.0000000
7.0000000
13.000000
0.000000

Son ortogonales a1, a2 y a3, a4.

solución apartado 3b

1.5000000
5.0000000
0.25000000

PROBLEMA 4

solución apartado 4a

3.000000
−2.0000000

v = 3u1 − 2u2

solución apartado 4b

5.0000000
3.0000000
−4.0000000

v = 5u1 + 3u2 − 4u3

solución apartado 4c

1.0000000
2.0000000
1.0000000
1.0000000

v = u1 + 2u2 + u3 + u4

PROBLEMA 5

solución apartado 5a

2.0000000

linealmente dependientes

solución apartado 5b

2.0000000

linealmente independientes

solución apartado 5c

2.0000000

linealmente dependientes

solución apartado 5d

3.0000000

linealmente independientes

solución apartado 5e

3.0000000

linealmente dependientes

solución apartado 5f

4.0000000

linealmente dependientes

PROBLEMA 6

solución apartado 6a

2.0000000

Como dim $\mathbf{R}^2$ en dos y son linealmente independientes, son una base

solución apartado 6b

3.0000000

son base

solución apartado 6c

4.0000000

son base

PROBLEMA 7

solución apartado 7a

2.0000000

linealmente dependientes

solución apartado 7b

2.0000000

linealmente independientes

solución apartado 7c

3.0000000
linealmente independientes

solución apartado 7d

2.0000000
linealmente independientes

solución apartado 7e

3.0000000
linealmente independientes

PROBLEMA 8

solución apartado 8a

2.0000000
−1.0000000
2.0000000

E = 2A − B + 2C

solución apartado 8b

3.0000000
−5.0000000
2.0000000

E = 3A − 5B + 2C

PROBLEMA 9

solución apartado 9a

2.0000000 Dim(W) = 2
2.0000000

Como $\begin{bmatrix} 1 \\ -2 \\ 5 \\ -3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \\ -4 \end{bmatrix}$ linealmente independientes son base W

solución apartado 9b

4.0000000

$\begin{bmatrix} 1 \\ -2 \\ 5 \\ -3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \\ -4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ como dim $(\mathbf{R}^4) = 4$ y son linealmente independientes, si forman base

PROBLEMA 10

solución apartado 10a

2.0000000

dimensión 2

solución apartado 10b

2.0000000

dimensión 2

solución apartado 10c

2.0000000

dimensión 2

solución 10d

1.0000000

dimensión 1

PROBLEMA 11

solución apartado 11a

2.0000000
−5.0000000
7.0000000

solución apartado 11b

1.0000000
1.0000000
1.0000000

PROBLEMA 12

solución apartado 12a

−7.0000000
11.000000
−21.000000
30.000000

solución apartado 12b

4.0000000
−2.0000000
1.0000000

PROBLEMA 13

solución apartado 13a

3

solución apartado 13b

2

solución apartado 13c

3

solución apartado 13d

2

PROBLEMA 14

solución apartado 14a

0.00000000

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -1 & 0 \\ -1 & 0 \end{bmatrix}$$

solución apartado 14b

1.0000000

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}$$

solución apartado 14c

2.0000000

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

PROBLEMA 15

solución apartado 15a

7.0000000
11.0000000

$k = 3 \qquad n = 2$

solución apartado 15b

12.000000

k = 2   n = 1

solución apartado 15c

7.0000000
−2.0000000
16.0000000

k = 2   n = 3

PROBLEMA 16

solución 16

1.0000000	0.00000000	1.0000000
0.50000000	1.5000000	0.00000000

PROBLEMA 17

solución apartado 17a

1.0000000
1.0000000
1.0000000

−1.0000000
0.00000000
1.0000000

1.0000000
2.0000000
3.0000000

1.0000000
−1.0000000
−3.0000000

Se podría determinar directamente, ya que las columnas de A son las imágenes por f de los vectores de la base de R4.

solución apartado 17b

2.0000000

$$\text{Dim(U)} = 2 \text{ y } \begin{bmatrix}1\\1\\1\end{bmatrix} \begin{bmatrix}1\\2\\3\end{bmatrix} \text{ forman una base}$$

PROBLEMA 18

solución apartado 18a y 18b

1.0000000
2.0000000
−1.0000000

0.00000000
1.0000000
1.0000000

1.0000000
1.0000000
−2.0000000

Dim(U) = 2

$$\text{Los vectores } \begin{bmatrix}1\\2\\-1\end{bmatrix} \text{ y } \begin{bmatrix}0\\1\\1\end{bmatrix} \text{ son una base de U}$$

PROBLEMA 19

solución apartado 19a y 19b

1.0000000
0.00000000

1.0000000
1.0000000

0.00000000
1.0000000

2.0000000

Dim(U) = 2

$$\text{Los vectores } \begin{bmatrix}1\\0\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} \text{ son una base de U}$$

PROBLEMA 20

solución p20

3.0000000	−23.000000	−3.0000000
−2.0000000	13.000000	2.0000000

PROBLEMA 21

solución apartado 21a

1.0000000	2.0000000
3.0000000	5.0000000

solución apartado 21b

−5.0000000	2.0000000
3.0000000	−1.0000000

solución apartado 21c

6.0000000	−7.0000000
−3.0000000	4.0000000

PROBLEMA 22

solución p22

1.7000000	1.7000000	3.0000000
−0.10000000	−0.10000000	8.3266727E-017

PROBLEMA 23

solución p23

1.0000000	2.0000000
−1.0000000	0.00000000
7.0000000	8.0000000

PROBLEMA 24

solución apartado 24a

$$\begin{bmatrix} 3 & -1 \\ 2 & 4 \\ 5 & -6 \end{bmatrix}$$

solución apartado 24b

$$\begin{bmatrix} 3 & -4 & 2 & -5 \\ 5 & 7 & -1 & 2 \end{bmatrix}$$

solución apartado 24c

$$\begin{bmatrix} 2 & 3 & -8 \\ 1 & 1 & 1 \\ 4 & -5 & 6 \end{bmatrix}$$

PROBLEMA 25

solución apartado 25a

−0.16455696	0.21518987	0.15189873
0.12658228	−0.088607595	0.11392405
0.30379747	−0.012658228	−0.12658228

solución apartado 25b

0.45833333	0.12500000	−0.083333333
−0.79166667	−0.12500000	0.41666667
0.083333333	−0.25000000	0.16666667

solución apartado 25c

3.4206579E-017	−0.12500000	1.6250000
0.87500000		
2.0000000	0.37500000	−3.8750000
−0.62500000		
3.0000000	0.62500000	−5.1250000
−0.37500000		
3.4203190E-017	−0.12500000	−0.37500000
−0.12500000		

solución apartado 25d

1.0000000	0.00000000	2.0000000
2.0000000	−1.0000000	3.0000000
4.0000000	1.0000000	8.0000000

solución apartado 25e

0.13636364	0.40909091	−0.090909091
0.15909091	−0.022727273	0.22727273
−0.068181818	0.29545455	0.045454545

PROBLEMA 26

solución apartado 26a

3.0000000	4.0000000
0.00000000	2.0000000
6.0000000	−1.0000000

solución apartado 26b

15.000000	−6.0000000
−15.000000	6.0000000

solución apartado 26c

−1.0000000	4.0000000	−2.0000000
−4.0000000	16.000000	−8.0000000
3.0000000	−12.000000	6.0000000

solución apartado 26d

0.00000000	0.00000000
0.00000000	0.00000000

solución apartado 26e

6.0000000	−12.000000
9.0000000	−5.0000000

solución apartado 26f

30.000000	−12.000000
−30.000000	12.000000

solución apartado 26g

11.000000	1.0000000	7.0000000
9.0000000	−1.0000000	−17.000000

solución apartado 26h

1.0000000	−4.0000000	2.0000000	2.0000000
−8.0000000	4.0000000		
−1.0000000	4.0000000	−2.0000000	−2.0000000
8.0000000	−4.0000000		
−1.0000000	4.0000000	−2.0000000	3.0000000
−12.000000	6.0000000		
1.0000000	−4.0000000	2.0000000	−3.0000000
12.000000	−6.0000000		
5.0000000	−20.000000	10.000000	−2.0000000
8.0000000	−4.0000000		
−5.0000000	20.000000	−10.000000	2.0000000
−8.0000000	4.0000000		

solución apartado 26i

2.0000000	−2.0000000	10.000000	2.0000000
−2.0000000	10.00000		
4.0000000	6.0000000	−4.0000000	4.0000000
6.0000000	−4.0000000		
1.0000000	−1.0000000	5.0000000	−1.0000000
1.0000000	−5.0000000		
2.0000000	3.0000000	−2.0000000	−2.0000000
−3.0000000	2.0000000		
1.0000000	−1.0000000	5.0000000	−3.0000000
3.0000000	−15.000000		
2.0000000	3.0000000	−2.0000000	−6.0000000
−9.0000000	6.0000000		

Capítulo 4
APLICACIONES MATEMATICAS (II)

4.1. Determinantes

Sea A una matriz cuadrada de orden n

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Definición

Se llama determinante de la matriz cuadrada A, y se designa por $\det(A) = |A|$, al número definido por:

$$|A| = \Sigma\ (\text{sig}\ \sigma)\ a_{ij1}\ a_{2j2} \dots a_{njn}$$

donde:

— $a_{iji} \dots a_{njn}$ es un producto de n factores tal que uno y sólo un elemento pertenece a cada fila y uno y sólo un elemento pertenece a cada columna de A,

— $(\text{sig}\ \sigma) = \begin{matrix} 1 & \text{si}\ \sigma\ \text{par} \\ -1 & \text{si}\ \sigma\ \text{impar} \end{matrix}$

siendo $\sigma = j_1, \dots, j_n$ par o impar si hay un número par o impar de parejas (i, k) *tales que* $i > k$ *pero* i *precede a* k *en* σ.

Con frecuencia se suele representar como

$$|A| = \mathrm{Det}(A) = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

Definición

Sea A una matriz cuadrada de orden n, si $|A| \neq 0$, se dice que A es una matriz *no singular o regular.*

Definición

Sea A una matriz cuadrada de orden *n*, denotamos por M_{ij} la submatriz cuadrada de orden $(n - 1)$ que se obtiene de suprimir la fila i-ésima y la columna j-ésima. El determinante de $|M_{ij}|$ se llama *menor* del elemento a_{ij} de la matriz A y definimos el *cofactor* de a_{ij}, denotado por A_{ij}, a:

$$A_{ij} = (-1)^{i+j}\, |M_{ij}|$$

Teorema

Si A es una matriz cuadrada de orden n no singular, entonces $rg(A) = n$.

Teorema

Una matriz A cuadrada de orden n, es invertible, es decir, existe A^{-1}, si A es de rango máximo, o lo que es lo mismo, $rg(A) = n$.

Teorema

Para toda matriz A cuadrada de orden n de rango máximo, $|A| \neq 0$

$$A^{-1} = \frac{1}{|A|}(\mathrm{adj}\ A')$$

siendo (adj A') la matriz traspuesta de los cofactores de los elementos a_{ij} de A

$$(\mathrm{adj}\ A') = \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{bmatrix}$$

Propiedades

Sea A una matriz cuadrada de orden **n**

1) $|A| = |A'|$

2) (i) Si A tiene una fila (columna) de ceros $|A| = 0$.
   (ii) Si A tiene dos filas (columnas) idénticas $|A| = 0$.
   (iii) Si A es triangular, $|A|$ = producto de los elementos de la diagonal. En particular $|I_n| = 1$.

3) Sea B la matriz que se obtiene de una matriz A:

   (i) Multiplicando una fila (columna) por un escalar K, entonces:

$$|B| = K|A|$$

   (ii) Intercambiando dos filas (columnas) de A, entonces

$$|B| = -|A|$$

   (iii) Sumando un múltiplo de una fila (columna) de A a otra, entonces:

$$|B| = |A|$$

4) Sea B una matriz cuadrada de orden n, entonces

$$|AB| = |A| \cdot |B|$$

4.2. Autovalores y autovectores de una matriz

Definición

Sea A una matriz cuadrada de orden *n* dada por:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

diremos que un escalar $\alpha \in \mathbf{R}$ es un valor propio o *autovalor* de A si existe un vector $\mathbf{x} \neq 0$ tal que:

$$A\mathbf{x} = \alpha\mathbf{x}$$

Al vector **x** correspondiente al autovalor α se llama *vector propio* o *autovector* de A.

α será un autovector de A si es tal que el sistema homogéneo

$$(A - \alpha I_n)\mathbf{x} = 0$$

tiene solución distinta de la trivial $\mathbf{x} = 0$, es decir, si α es tal que

$$|A - \alpha I_n| = 0$$

siendo

$$I_n = \begin{bmatrix} 1 & 0 \dots 0 \\ 0 & 1 \dots 0 \\ \dots & \dots \\ 0 & 0 \dots 1 \end{bmatrix}$$

$$|A - \alpha I_n| = \begin{vmatrix} a_{11}-\alpha & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22}-\alpha & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn}-\alpha \end{vmatrix} = P_n(\alpha) = 0$$

El polinomio $P_n(\alpha)$ de grado n es el *polinomio característico* de la matriz A y la ecuación $P_n(\alpha) = 0$ es la *ecuación característica* de A.

Propiedades

(i) Los autovectores son siempre distintos de cero.
(ii) Si **x** es autovector de A correspondiente al autovalor α y $a \in \mathbf{R}$ entonces $a \cdot \mathbf{x}$ es también autovector de A.
(iii) Sean $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n$ autovectores de A, correspondientes al mismo autovalor α, entonces el vector

$$\mathbf{v} = a_1\mathbf{x}^1 + a_2\mathbf{x}^2 + \dots + a_n\mathbf{x}^n;\ a_1, a_2, \dots, a_n \in \mathbf{R},$$

es también autovector de A correspondiente a α.
(iv) Los autovectores asociados a un mismo autovalor α forman un subespacio vectorial de E que llamaremos $\mathbf{V}_\alpha$.
(v) Si $\alpha_1 \neq \alpha_2$ son autovalores de A, $\mathbf{V}_{\alpha 1}$ y $\mathbf{V}_{\alpha 2}$ son subespacios vectoriales disjuntos. Es decir, ningún autovector puede serlo de dos autovalores distintos a la vez.
(vi) Los vectores $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ correspondientes respectivamente a los valores propios $\alpha_2, \dots, \alpha_n$ diferentes entre sí, son *linealmente independientes*.
(vii) Todas las matrices semejantes tienen los mismos autovalores y autovectores.

Matrices diagonales

Definición

Una matriz A cuadrada de orden n se dice que es diagonalizable, cuando existe una matriz P cuadrada de orden n, y *no singular,* tal que $P^{-1} \cdot A \cdot P = D$, donde D es una matriz diagonal.

Propiedades

(i) En una matriz diagonal D, los autovalores coinciden con los elementos de la diagonal principal.
(ii) Los autovalores de A son los mismos que los de la matriz

$$D = P^{-1} \cdot A \cdot P$$

(iii) Si **u** es un autovector de la matriz $D = P^{-1} \cdot A \cdot P$ correspondiente al autovalor α, el vector $\mathbf{x} = P \cdot \mathbf{u}$ es un autovector de A correspondiente al mismo autovalor α, es decir:

$$\begin{bmatrix} D \cdot \mathbf{u} = \alpha \cdot \mathbf{u} \\ D \quad = P^{-1}A \cdot P \end{bmatrix} \text{ implica que } [A\mathbf{x} = \alpha\mathbf{x}] \text{ con } \mathbf{x} = P \cdot \mathbf{u}$$

(iv) *Teorema fundamental de la diagonalización*

Sean $\alpha_1, \dots, \alpha_n$ los autovalores de A

a) A es diagonizable sí y sólo sí existe en E una base $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ de autovalores de A, siendo E el conjunto de los autovectores de A relativos a todos y cada uno de los autovalores.
b) Si A es diagonizable la matriz P viene dada por $P = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, es decir, las columnas de P son los vectores $\mathbf{x}_1, \dots, \mathbf{x}_n$ que constituyen una base.

Como consecuencia de esta propiedad, una condición *suficiente* para que una matriz A sea diagonizable es que sus autovalores sean todos distintos entre sí.

(v) a) Si A es simétrica es diagonizable.
b) Si A es una simétrica de orden n, es decir, $A = A'$ entonces existe una base ortonormal de autovectores $B = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ correspondientes a los autovalores $\alpha_1, \dots, \alpha_n$.
c) Si P es la matriz que tiene por columnas los autovectores de B, es decir, $P = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, se verifica que:

$$P^{-1} \cdot A \cdot P = D$$

siendo D la matriz diagonal

$$D = \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \alpha_\alpha \end{bmatrix}$$

4.3. Formas cuadráticas

Definición

Se llama *forma cuadrática* de n variables a toda expresión de la forma:

$$q(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}x_ix_j \text{ con } a_{iji}, x_i \in R \text{ y } \mathbf{x}=(x_i) \quad i = 1, \dots, n$$

Como a cada vector $\mathbf{x} \in \mathbf{R}^n$ $q(\mathbf{x})$ toma un valor real, será pues una aplicación de $\mathbf{R}^n$ en $\mathbf{R}$.

Se puede expresar también como:

$$q(\mathbf{x}) = [x_1, \dots, x_n] \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

o si

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

$$q(\mathbf{x}) = \mathbf{x}'A\mathbf{x}$$

expresión matricial de la forma cuadrática $q(\mathbf{x})$.

Toda forma cuadrática $q(\mathbf{x})$ tiene asociada a ella diferentes matrices, y en particular, siempre existirá entre ellas una que sea simétrica.

Definición

Se llama *matriz asociada* a la forma cuadrática $q(\mathbf{x})$, a una matriz A simétrica, tal que

$$q(\mathbf{x}) = \mathbf{x}'A\mathbf{x}$$

Se dice que una forma $q(\mathbf{x})$ viene dada en forma canónica cuando:

$$\begin{aligned} q(\mathbf{x}) = {} & \alpha_1 (C_{11}x_1 + \dots + C_{1n}x_n)^2 + \\ & + \alpha_2 (C_{21}x_1 + \dots + C_{2n}x_n)^2 + \\ & \dots\dots\dots\dots\dots\dots \\ & + \alpha_n (C_{n1}x_1 + \dots + C_{1n}x_n)^2 \end{aligned}$$

Con $C_{ij} \in \mathbf{R}$ para todo i, j.

Si llamamos

$$\begin{aligned} y_1 &= C_{11}x_1 + \dots + C_{1n}x_n \\ y_2 &= C_{12}x_1 + \dots + C_{2n}x_n \\ & \dots\dots\dots\dots \\ y_n &= C_{n1}x_1 + \dots + C_{nn}x_n \end{aligned}$$

con $|C| \neq 0$ para

$$C = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \dots & \dots & \dots & \dots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$

$$q(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i^2 = [y_1, \dots, y_n] \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \alpha_n \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \mathbf{y}'D\mathbf{y}$$

siendo

$$D = \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \alpha_n \end{bmatrix} \quad \text{y} \quad D = C^{-1}AC$$

Como A es simétrica siempre existirá D.

Clasificación de las formas cuadráticas

Dada $q(\mathbf{x}) = \mathbf{x}'A\mathbf{x}$ se dice que es:

(i) Definida positiva si: $q(\mathbf{x}) > 0$ para todo $\mathbf{x} \neq 0$
(ii) Definida negativa si: $q(\mathbf{x}) < 0$ para todo $\mathbf{x} \neq 0$
(iii) Semidefinida positiva si:
  a) $q(\mathbf{x}) \geq 0$ para todo $\mathbf{x} \in \mathbf{R}^n$;
  b) existe algún $\mathbf{x} \neq 0$ tal que $q(\mathbf{x}) = 0$
(iv) Semidefinida negativa si:
  a) $q(\mathbf{x}) \leq 0$ para todo $\mathbf{x} \in \mathbf{R}^n$
  b) existe algún $\mathbf{x} \neq 0$ tal que $q(\mathbf{x}) = 0$
(v) Indefinida cuando $q(\mathbf{x})$ puede tomar valores positivos y negativos.

Definición

Sea la matriz A dada por

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

definimos como menores principales de la matriz A, a los elementos $D_1, \ldots, D_n$, dados por

$$D_1 = |a_{11}| \quad D_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \quad D_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{23} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Criterios de los menores principales para la definición del signo de una forma cuadrática

Dada $q(\mathbf{x}) = \mathbf{x}'\, A\, \mathbf{x}$ con A simétrica

(i) Es definida positiva si: $D_1 > 0, D_2 > 0, \ldots, D_n > 0$.
(ii) Es definida negativa si;

$$D_1 < 0,\ D_2 > 0,\ \ldots,\ (-1)^n D_n > 0$$

Criterios de los autovalores de la matriz A para la definición del signo de una forma cuadrática

Teorema

Sea $q(\mathbf{x}) = \mathbf{x}'\, A\, \mathbf{x}$ con A matriz cuadrada de orden n simétrica, entonces si α_i son los autovalores de A una condición necesaria y suficiente para que:

(i) $q(\mathbf{x})$ sea definida positiva es que todos los autovalores de A sean positivos.
(ii) $q(\mathbf{x})$ sea definida negativa es que todos los autovalores de A sean negativos.
(iii) $q(\mathbf{x})$ sea semidefinida positiva es que todos los autovalores de A sean no negativos y al menos uno de ellos nulo.
(iv) $q(\mathbf{x})$ sea semidefinida negativa, es que todos los autovalores de A sean no positivos y al menos uno de ellos nulo.
(v) $q(\mathbf{x})$ sea indefinida cuando exista algún autovalor de A positivo y algún autovalor negativo.

4.4. Sistema de ecuaciones lineales

Un sistema de ecuaciones lineales viene dado por:

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_m \end{array} \qquad (1)$$

Cuando $b_1 = b_2 = \ldots = b_m = 0$, el sistema se llama homogéneo.

Se dice que dos sistemas son equivalentes si tienen la misma solución.

Sea

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{21} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{22} & \cdots & a_{mn} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ \ldots \\ x_n \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ \ldots \\ b_m \end{bmatrix}$$

y

$$A^* = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix}$$

El sistema (1) puede escribirse como $A\mathbf{x} = \mathbf{b}$, donde $\mathbf{x} \in \mathbf{R}^n$ $\mathbf{b} \in \mathbf{R}^m$, luego (1) es una aplicación lineal $f: \mathbf{R}^n \to \mathbf{R}^m$.

$(x_1, ..., x_n)$ coordenadas del vector $\mathbf{x}$ respecto de la base B de $\mathbf{R}^n$.

$(b_1, ..., b_m)$ coordenadas del vector $\mathbf{b}$ respecto de la base B' de $\mathbf{R}^m$.

A es la matriz asociada a f respecto de B y B', y tiene tantas filas como la dimensión del espacio final y tantas columnas como la dimensión del espacio inicial.

Definición

Sea el sistema de ecuaciones lineales (1), se define como *solución* del mismo, a todo vector $\mathbf{c}$ perteneciente a $\mathbf{R}^n$ tal que

$$A\mathbf{c} = \mathbf{b}$$

Dos sistemas de ecuaciones lineales se dice que son equivalentes cuando tienen la misma solución.

Teorema (Rouché-Frobenius)

1) Para sistemas no homogéneos:

— Si $\text{rang}(A) = \text{rg}(A^*)$ el sistema tiene solución (sistema compatible):

i) $\text{rang}(A) = \text{rang}(A^*) =$ dimensión espacio inicial (número de incógnitas) la solución será única (sistema compatible y determinado).

ii) $\text{rang}(A) = \text{rag}(A^*) <$ número de incógnitas, habrá infinitas soluciones. Sistema compatible e indeterminado.

— Si $\text{rang}(A) \neq \text{rang}(A^*)$ el sistema no tiene solución (sistema no compatible).

2) Para sistemas homogéneos:

Como $\text{rang}(A) = \text{rang}(A^*)$ tiene siempre la solución trivial $\mathbf{x} = 0$:

i) Si $\text{rang}(A) =$ n.º incógnitas tiene solución trivial como única (sistema compatible y determinado).

ii) Si $\text{rang}(A) <$ n.º de incógnitas tiene infinitas soluciones (sistema compatible e indeterminado).

Método de Cramer para la solución de un sistema no homogéneo compatible y determinado (Número de ecuaciones igual al de incógnitas)

Las condiciones que debe cumplir un sistema de ecuaciones lineales para resolverlo por el método de Cramer son:

i) A es una matriz cuadrada.

ii) $|A| \neq 0$.

Es decir:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_1 \quad + a_{2n}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

$$\Leftrightarrow A\mathbf{x} = \mathbf{b} \Leftrightarrow A^{-1}A\mathbf{x} = A^{-1}\mathbf{b} \Leftrightarrow A^{-1}\mathbf{b}$$

ya que $\text{rang}(A) = \text{rang}(A^*) = n$.

La solución del sistema se puede expresar también como

$$x_i = \frac{D(x_1, x_2, ..., x_{i-1}, b_1, x_{i+1}, ..., x_n)}{|A|}$$

siendo

$$D(x_1, ..., x_{i-1}, b_1, x_{i+1}, ..., x_n) =$$

$$= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1i-1} & b_1 & a_{1i+1} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2i-1} & b_2 & a_{2i+1} & \dots & a_{2n} \\ \dots & & & & & & & \dots \\ a_{n1} & a_{n2} & \dots & a_{ni-1} & b_n & a_{ni+1} & \dots & a_{nn} \end{bmatrix}$$

Método de eliminación de Gauss

Si el sistema a resolver es muy grande, el método de Cramer resulta muy laborioso y en este caso es conveniente utilizar el método de Gauss.

Definición

Llamaremos *operaciones básicas* a:

i) Intercambio de dos ecuaciones.

ii) Producto de una ecuación por un escalar no nulo.
iii) Sumar a una ecuación otra multiplicada por un escalar no nulo.

Estas operaciones pueden hacerse sobre las ecuaciones del sistema o bien sobre las filas de la matriz A^*.

El método de eliminación de Gauss, consiste en utilizar las operaciones básicas, para transformar el sistema

$$A\mathbf{x} = \mathbf{b}$$

en un sistema $U\mathbf{x} = \mathbf{c}$, llamado sistema reducido, con las siguientes características:

— Todos los elementos por debajo de a_{ii} ($i = 1 \dots n$) son nulos.
— El elemento no nulo de cada fila, está a la derecha del primer elemento diferente de cero de la fila anterior.
— Cualquier fila formada únicamente por ceros está bajo todas las filas con elementos diferentes de cero.

PROBLEMAS PROPUESTOS

TEMA 4.1

1. Calcular el determinante de las siguientes matrices:

a) $\begin{bmatrix} 1 & 2 & 3 \\ 4 & -2 & 3 \\ 2 & 5 & -1 \end{bmatrix}$ b) $\begin{bmatrix} 2 & 5 & -3 & -2 \\ -2 & -3 & 2 & -5 \\ 1 & 3 & -2 & 2 \\ -1 & 6 & 4 & 3 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 2 & 4 & 6 & 7 \\ 2 & 3 & -1 & -2 & 0 \\ -1 & -2 & 4 & 1 & -1 \\ 3 & 2 & 1 & -1 & -2 \\ 2 & 5 & -3 & 4 & -1 \end{bmatrix}$

2. Calcular el cofactor a) de 7 y b) de 6 en la matriz:

$$\begin{bmatrix} 2 & 1 & -3 & 4 \\ 5 & -4 & 7 & -2 \\ 4 & 0 & 6 & -3 \\ 3 & -2 & 5 & 2 \end{bmatrix}$$

3. Calcular:

a) det(A)
b) Adj(A)

c) verificar que $A \times \mathrm{Adj}(A') = \det(A) \times I$
d) inversa de A

siendo A la matriz:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 1 & 5 & 7 \end{bmatrix}$$

4. Comprobar que $\det(A) = \det(A')$, siendo A:

a) $\begin{bmatrix} 1 & 3 & 4 & 5 & 6 \\ 2 & 0 & 0 & 3 & 5 \\ 0 & 2 & 1 & 4 & -1 \\ 4 & 4 & 3 & 6 & 2 \\ 0 & 0 & -1 & 3 & 0 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 3 & 0 & 2 & 4 \\ 3 & 0 & 6 & 0 & 3 \\ 4 & 1 & 7 & 1 & 2 \\ 2 & 2 & 8 & 3 & 0 \\ 3 & 5 & 0 & 4 & 4 \end{bmatrix}$

c) $\begin{bmatrix} 3 & 1 & 2 & 0 \\ 0 & 1 & 4 & 2 \\ 4 & 5 & -1 & 0 \\ 0 & 2 & 3 & 1 \end{bmatrix}$

5. Calcular la matriz adjunta de:

a) $\begin{bmatrix} 2 & 3 & -1 \\ -1 & 1 & 2 \\ 1 & -4 & 1 \end{bmatrix}$ b) $\begin{bmatrix} 2 & -1 & 4 \\ 1 & 0 & -3 \\ 3 & 2 & -5 \end{bmatrix}$

6. Comprobar que:

a) $\det(A \cdot B) = \det(A) \cdot \det(B)$
b) $\det(A + B) \neq \det(A) + \det(B)$, siendo:

$$A = \begin{bmatrix} 1 & 3 & 4 & 2 \\ 5 & -1 & 0 & -4 \\ -3 & 2 & 7 & 0 \\ 4 & -1 & 3 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 2 & -1 & 5 \\ 4 & 3 & 2 & 7 \\ -2 & -4 & 1 & -1 \\ -1 & 2 & 5 & 6 \end{bmatrix}$$

7. Dados los siguientes vectores indicar si forman una base:

a) $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 4 \\ 7 \end{bmatrix}$ de $\mathbf{R}^2$

b) $\begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 2 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ -2 \end{bmatrix}$ de $\mathbf{R}^3$

c) $\begin{bmatrix} 1 \\ 2 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} 3 \\ -2 \\ 0 \\ 2 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ -1 \\ 5 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \\ 3 \\ 4 \end{bmatrix}$ de $\mathbf{R}^4$

1. Calcular los autovalores y autovectores de:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 6 & -2 \\ 2 & 2 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 5 \end{bmatrix}$$

2. Diagonalizar cuando sea posible las matrices del problema 1.

3. Dada

$$A = \begin{bmatrix} 3 & 4 & -6 \\ 4 & 3 & -6 \\ -6 & -6 & -14 \end{bmatrix}$$

calcular $\Sigma\alpha_i$, siendo α_i los autovalores de A.

4. Encontrar la matriz P tal que P^{-1} A P sea diagonal, siendo

$$A = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 5 & -1 \\ 1 & -1 & 3 \end{bmatrix}$$

5. Demostrar que la matriz A no es diagonizable

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

6. Comprobar que las matrices A y B no son semejantes, aunque poseen el mismo polinomio característico.

$$A = \begin{bmatrix} 7 & -2 & 1 \\ -2 & 10 & -2 \\ 1 & -2 & 7 \end{bmatrix} \quad B = \begin{bmatrix} 6 & 1 & 1 \\ 0 & 6 & 1 \\ 0 & 0 & 12 \end{bmatrix}$$

7. Diagonalizar:

$$A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 0 \end{bmatrix}$$

8. Estudiar la diagonalización de:

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

9. Dada

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$$

se pide: a) autovalores, b) hallar una matriz inversible P tal que P^{-1} A P sea diagonal.

10. Hallar los autovalores y dar una base para los subespacios vectoriales correspondientes a ellos

$$A = \begin{bmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 2 & -1 \\ -1 & 1 & 4 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

11. Hallar los autovalores y autovectores de la matriz asociada a la aplicación lineal:

$$f(x, y, z) = (2x + y, y - z, 2y + 4z)$$

12. Clasificar las siguientes formas cuadráticas:

$$Q(x) = x_1^2 + 2x_1x_2 + 4x_2x_3 + 6x_2^2 + x_3^2$$
$$Q(x) = -x_1^2 - x_2^2 - x_3^2 - x_4^2 + 6x_2x_4$$
$$Q(x) = x_1^2 + x_2^2 + 3x_3^2 - 4x_1x_2$$
$$Q(x) = -x_1^2 - 6x_2^2 - 2x_3^2 + 4x_1x_2$$

Estudiar la compatibilidad de los siguientes sistemas de ecuaciones lineales, dando la solución de aquellos que sean compatibles y determinados:

1. $$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 6 \\ 2x_1 - x_2 - x_3 &= 0 \\ 2x_1 - 3x_2 + 2x_3 &= 1 \\ 5x_1 - 2x_2 + 4x_3 &= 7 \end{aligned}$$

2. $$\begin{aligned} 2x_1 + 2x_2 + 3x_3 + x_4 &= 1 \\ 3x_1 - x_2 + x_3 + 3x_4 &= 2 \\ -2x_1 + 3x_2 + x_3 - 2x_4 &= 0 \\ x_1 + 5x_2 + 3x_3 - 3x_4 &= 2 \\ 2x_1 + 7x_2 + 3x_3 - 2x_4 &= 8 \end{aligned}$$

3. $$\begin{aligned} 3x + 2y &= 3 \\ y - z &= 1 \\ x + z &= 0 \end{aligned}$$

4. $$\begin{aligned} 2x + y + 3z &= 1 \\ 4x + 2y + 2z &= 2 \end{aligned}$$

5. $$\begin{aligned} 2x + y + z &= 6 \\ 3x - 2y + 3z &= 7 \\ x - 3y + 2z &= 0 \end{aligned}$$

6. $$\begin{aligned} x + 2y &= 5 \\ x - 3y &= -5 \\ 3x + y &= 5 \end{aligned}$$

7. $$\begin{aligned} x - y + z - u + v &= -1 \\ -x + y + u - 2v &= 2 \\ x - y + 2z - u &= 0 \\ 2x - y + 2z - u &= 1 \\ 3x - 2y + 5z - 2u + v &= 0 \end{aligned}$$

8. $$\begin{aligned} 2x + y - 2z &= 10 \\ 3x + 2y + 2z &= 1 \\ 5x + 4y + 3z &= 4 \end{aligned}$$

9. $$\begin{aligned} x - 2y + 3z - 2w &= 0 \\ 3x - 7y - 2z + 4w &= 0 \\ 4x + 3y + 5z + 2w &= 0 \end{aligned}$$

10. $$\begin{aligned} x + 2y - 3z &= 0 \\ 2x + 5y + 2z &= 0 \\ 3x - y - 4z &= 0 \end{aligned}$$

11. $$\begin{aligned} x + 2y - z &= 0 \\ 2x + 5y + 2z &= 0 \\ x + 4y + 7z &= 0 \\ x + 3y + 3z &= 0 \end{aligned}$$

12. $$\begin{aligned} x + 2y + 2z &= 2 \\ 3x - 2y - z &= 5 \\ 2x - 5y + 3z &= -4 \\ x + 4y + 6z &= 0 \end{aligned}$$

13. $$\begin{aligned} 2x + y - 3z &= 5 \\ 3x - 2y + 2z &= 5 \\ 5x - 3y - z &= 16 \end{aligned}$$

14. $$\begin{aligned} 2x + 3y &= 4 \\ x - 2y &= 1 \\ 3x + 5y &= 2 \end{aligned}$$

15. $$\begin{aligned} x + y &= 0 \\ y + z &= 0 \\ x + t &= 0 \\ y + t &= 0 \end{aligned}$$

16. $$\begin{aligned} 3x - 3y - 2z - t &= 0 \\ 3y - 2z + t &= 0 \\ x + y + 3z + 5t &= 0 \\ x + y + z + 3t &= 0 \end{aligned}$$

17. $$\begin{aligned} x - 2y + 2z &= 0 \\ 2x + y - 2z &= 0 \\ 3x + 4y - 6z &= 0 \\ 3x - 3y + 12z &= 0 \end{aligned}$$

18. $$\begin{aligned} 2x + y &= 0 \\ x + y &= 1 \\ x - y &= -1 \\ x + y &= 0 \end{aligned}$$

FORMULACION DE LOS PROBLEMAS EN GAUSS

TEMA 4.1

```
/* PROBLEMA 1

     Apartado 1a */;

          output file = d:tema2.p1 on;
          A = {1 2 3, 4 −2 3, 2 5 −1};
          s1a = det(A);
          print "solución apartado 1a";
          print s1a;

     /* Apartado 1b */;

          A = {2 5 −3 −2, −2 −3 2 −5, 1 3 −2 2, −1 6 4 3};
          s1b = det(A);
          print "solución apartado 1b";
          print s1b;

     /* Apartado 1c */;

          A = {1 2 4 6 7, 2 3 −1 −2 0, −1 −2 4 1 −1, 3 2 1 −1 −2,
          2 5 −3 4 −1};
          s1c = det(A);
          print "solución apartado 1c";
          print s1c;
          output file = d:tema2.p1 off;
```

```
/* PROBLEMA 2

     Apartado 2a */;

          output file = d:tema2.p2 on;

          c7 = {2 1 4, 4 0 −3, 3 −2 2};
          s2a = det(c7);
          print "solución apartado 2a";
          print s2a;

     /* Apartado 2b */;

          c6 = {2 1 4, 5 −4 −2, 3 −2 2};
          s2b = det(c6);
          print "solución apartado 2b";
          print s2b;
          output file = d:tema2.p2 off;


/* PROBLEMA 3

     Apartado 3a */;

          output file = d:tema2.p3 on;

          A = {1 2 3, 2 3 4, 1 5 7};
          s3a = det(A);
          print "solución apartado 3a";
          print s3a;

     /* Apartado 3b */;

          A11 = {3 4, 5 7};
          A12 = {2 4, 1 7};
          A13 = {2 3, 1 5};
          A21 = {2 3, 5 7};
          A22 = {1 3, 1 7};
          A23 = {1 2, 1 5};
          A31 = {2 3, 3 4};
          A32 = {1 3, 2 4};
          A33 = {1 2, 2 3};
          s3b1 = det(A11);
          s3b2 = det(A12);
          s3b3 = det(A13);
          s3b4 = det(A21);
          s3b5 = det(A22);
          s3b6 = det(A23);
```



```
          s3b7 = det(A31);
          s3b8 = det(A32);
          s3b9 = det(A33);
          print "solución apartado 3b";
          print s3b1;
          print s3b2;
          print s3b3;
          print s3b4;
          print s3b5;
          print s3b6;
          print s3b7;
          print s3b8;
          print s3b9;

     /* Apartado 3c */;

          /* B = adj(A) */;
          A = {1 2 3, 2 3 4, 1 5 7};
          B = {1 1 −1, −10 4 2, 7 −3 −1};
          s3c = A*B;
          print "solución apartado 3c";
          print s3c;

     /* Apartado 3d */;

          A = {1 2 3, 2 3 4, 1 5 7};
          s3d = inv(A);
          print "solución apartado 3d";
          print s3d;
          output file = d:tema2.p3 off;


/* PROBLEMA 4

     Apartado 4a */;

          output file = d:tema2.p4 on;

          A = {1 3 4 5 6, 2 0 0 3 5, 0 2 1 4 −1, 4 4 3 6 2, 0 0 −1 3 0};
          s4a1 = det(A);
          s4a2 = det(A');
          print "solución apartado 4a";
          print s4a1;
          print s4a2;

     /* Apartado 4b */;

          A = {1 3 0 2 4, 3 0 6 0 3, 4 1 7 1 2, 2 2 8 3 0, 3 5 0 4 4};
```

```
        s4b1 = det(A);
        s4b2 = det(A');
        print "solución apartado 4b";
        print s4b1;
        print s4b2;

    /* Apartado 4c */;

        A = {3 1 2 0, 0 1 4 2, 4 5 −1 0, 0 2 3 1};
        s4c1 = det(A);
        s4c2 = det(A');
        print "solución apartado 4c";
        print s4c1;
        print s4c2;
        output file = d:tema2.p4 off;


/* PROBLEMA 5

    Apartado 5a */;

        output file = d:tema2.p5 on;
        A11 = {1 2, −4 1};
        A12 = {−1 2, 1 1};
        A13 = {−1 1, 1 −4};
        A21 = {3 −1, −4 1};
        A22 = {2 −1, 1 1};
        A23 = {2 3, 1 −4};
        A31 = {3 −1, 1 2};
        A32 = {2 −1, −1 2};
        A33 = {2 3, −1 1};
        s5a1 = det(A11);
        s5a2 = det(A12);
        s5a3 = det(A13);
        s5a4 = det(A21);
        s5a5 = det(A22);
        s5a6 = det(A23);
        s5a7 = det(A31);
        s5a8 = det(A32);
        s5a9 = det(A33);
        print "solución apartado 5a";
        print s5a1;
        print s5a2;
        print s5a3;
        print s5a4;
        print s5a5;
        print s5a6;
        print s5a7;
        print s5a8;
        print s5a9;

    /* Apartado 5b */;

        B11 = {0 −3, 2 −5};
        B12 = {1 −3, 3 −5};
        B13 = {1 0, 3 2};
        B21 = {−1 4, 2 −5};
        B22 = {2 4, 3 −5};
        B23 = {2 −1, 3 2};
        B31 = {−1 4, 0 −3};
        B32 = {2 4, 1 −3};
        B33 = {2 −1, 1 0};
        s5b1 = det(B11);
        s5b2 = det(B12);
        s5b3 = det(B13);
        s5b4 = det(B21);
        s5b5 = det(B22);
        s5b6 = det(B23);
        s5b7 = det(B31);
        s5b8 = det(b32);
        s5b9 = det(B33);
        print "solución apartado 5b";
        print s5b1;
        print s5b2;
        print s5b3;
        print s5b4;
        print s5b5;
        print s5b6;
        print s5b7;
        print s5b8;
        print s5b9;


/* PROBLEMA 6

    Apartado 6a */;

        output file = d:tema2.p6 on;

        A = {1 3 4 −2, 5 −1 0 −4, −3 2 7 0, 4 −1 3 −1};
        B = {3 2 −1 5, 4 3 2 7, −2 −4 1 −1, −1 2 5 6};
        s6a1 = det(A*B);
```

```
        s6a2 = det(A)*det(B);
        print "solución apartado 6a";
        print s6a1;
        print s6a2;

    /* Apartado 6b */;

        s6b1 = det(A+B);
        s6b2 = det(A) + det(B);
        print "solución apartado 6b";
        print s6b1;
        print s6b2;
        output file = d:tema2.p6 off;

/* PROBLEMA 7

    Apartado 7a */;

        output file = d:tema2.p7 on;
        u1 = {1, 2};
        u2 = {4, 7};
        x= {1 4, 2 7};
        s7a = det(x);
        print "solución apartado 7a";
        print s7a;

    /* Apartado 7b */;

        u1 = {0, 1, 3};
        u2 = {4, 2, 1};
        u3 = {4, 1, −2};
        x = {0 4 4, 1 2 1, 3 1 −2};
        s7b = det(x);
        print "solución apartado 7b";
        print s7b;

    /* Apartado 7c */;

        u1 = {1, 2, −1, 0};
        u2 = {3, −2, 0, 2};
        u3 = {4, 2, −1 5};
        u4 = {−1, −2, 3, 4};
        x = {1 3 4 −1, 2 −2 2 −2, −1 0 −1 3, 0 2 5 4};
        s7c = det(x);
        print "solución apartado 7c";
        print s7c;
        output file = d:tema2.p7 off;
```


TEMAS 4.2 y 4.3

```
/* PROBLEMA 1

    Apartado 1a */;

        output file = d:tema3.p1 on;
        a = {0 0 0 1, 0 0 1 0, 0 1 0 0, 1 0 0 0};
        {var, vai, ver, vei} = eigrg2(a);
        print "solución apartado 1a";
        print var;
        print ver;

    /* Apartado 1b */;

        b = {6 −2, 2 2};
        {var, vai, ver, vei} = eigrg2(b);
        print "solución apartado 1b";
        print var;
        print ver;

    /* Apartado 1c */;

        c = {1 0 0, 1 2 0, 1 1 5};
        {var, vai, ver, vei} = eigrg2(c);
        print "solución apartado 1c";
        print var;
        print ver;
        output file = d:tema3.p1 off;


/* PROBLEMA 2

    Apartado 2a */;

        output file = d:tema3.p2 on;

        P = {−1 0 1 0, 0 1 0 1, 0 −1 0 1, 1 0 1 0};
        a = {0 0 0 1, 0 0 1 0, 0 1 0 0, 1 0 0 0};
        s2a = inv(P)*a*P;
        print "solución apartado 2a";
        print s2a;

    /* Apartado 2c */;

        P = {1 0 0, −1 3 0, 0 −1 1};
        c = {1 0 0, 1 2 0, 1 1 5};
        s2c = inv(P)*c*P;
```

```
print "solución apartado 2c";
print s2c;
output file = d:tema3.p2 off;

/* PROBLEMA 3 */;

output file = d:tema3.p3 on;

A = {3 4 −6, 4 3 −6, −6 −6 −14};
{var, vai, ver, vei} = eigrg2(A);
print "solución problema 3";
print var;
output file = d:tema3.p3 off;

/* PROBLEMA 4 */;

output file = d:tema3.p4 on;
A = {3 −1 1, −1 5 −1, 1 −1 3};
{var, vai, ver, vei} = eigrg2(A);
print "solución problema 4";
print var;
print ver;
output file = d:tema3.p4 off;

/* PROBLEMA 5 */;

output file = d:tema3.p5 on;
A = {1 1, 0 1};
{var, vai, ver vei} = eigrg2(A);
print "solución problema 5";
print var;
print ver;
output file = d:tema3.p5 off;

/* PROBLEMA 6

Apartado 6a */;

output file = d:tema3.p6 on;
A = {7 −2 1, −2 10 −2, 1 −2 7};
{var, vai, ver, vei} = eigrg2(A);
print "solución apartado 6a";
print var;
print ver;
```



```
/* Apartado 6b */;

B = {6 1 1, 0 6 1, 0 0 12};
{var, vai, ver, vei} = eigrg2(B);
print "solución apartado 6b";
print var;
print ver;
output file = d:tema3.p6 off;

/* PROBLEMA 7

Apartado 7a */;

output file = d:tema3.p7 on;
A = {1 −1 0, −1 2 −1, 0 −1 1};
{var, vai, ver, vei} = eigrg2(A);
P = {1 −1 1, 1 0 −2, 1 1 1};
s7a = inv(P)*A*P;
print "solución apartado 7a";
print var;
print ver;
print s7a;

/* Apartado 7b */;

B = {0 0 2, 0 2 0, 2 0 0};
{var, vai, ver, vei} = eigrg2(B);
P = {−1 1 0, 0 0 1, 1 1 0};
s7b = inv(P)*B*P;
print "solución apartado 7b";
print var;
print ver;
print s7b;
output file = d:tema3.p7 off;

/* PROBLEMA 8 */;

output file = d:tema3. p8 on;
A = {2 1 0, 0 2 1, 0 0 3};
{var, vai, ver, vei} = eigrg2(A);
print "solución problema 8";
print var;
print ver;
output file = d:tema3.p8 off;
```

```
/* PROBLEMA 9 */;

    output file = d:tema3.p9 on;
    A = {1 4, 2 3};
    {var, vai, ver, vei} = eigrg2(A);
    print "solución problema 9";
    print var;
    print ver;
    output file = d:tema3.p9 off;

/* PROBLEMA 10

  Apartado 10a */;

    output file = d:tema3.p10 on;
    A = {1 −3 3, 3 −5 3, 6 −6 4};
    {var, vai, ver, vei} = eigrg2(A);
    print "solución apartado 10a";
    print var;
    print ver;

  /* Apartado 10b */;

    B = {1 2 2, 1 2 −1, −1 1 4};
    {var, vai, ver, vei} = eigrg2(B);
    print "solución apartado 10b";
    print var;
    print ver;

  /* Apartado 10c */;

    C = {1 1 0, 0 1 0, 0 0 1};
    {var, vai, ver, vei} = eigrg2(C);
    print "solución apartado 10c";
    print var;
    print ver;
    output file = d:tema3.p10 off;

/* PROBLEMA 11 */;

    output file = d:tema3.p11 on;
    A = {2 1 0, 0 1 −1, 0 2 4};
    {var, vai, ver, vei} = eigrg2(A);
    print "solución problema 11";
    print var;
    print ver;
    output file = d:tema3.p11 off;
```



```
/* PROBLEMA 12

  Apartado 12a */;

    output file = d:tema3.p12 on;
    A = {1 1 0, 1 6 2, 0 2 1};
    {var, vai, ver, vei} = eigrg2(A);
    print "solución apartado 12a";
    print var;

  /* Apartado 12b */;

    B = {−1 0 0 0, 0 −1 0 3, 0 0 −1 0, 0 3 0 −1};
    {var, vai, ver, vei} = eigrg2(B);
    print "solución apartado 12b";
    print var;

  /* Apartado 12c */;

    C = {1 −2 0, −2 1 0, 0 0 3};
    {var, vai, ver, vei} = eigrg2(C);
    print "solución apartado 12c";
    print var;

  /* Apartado 12d */;

    D = {−1 2 0, 2 −6 0, 0 0 −2};
    {var, vai, ver, vei} = eigrg2(D);
    print "solución apartado 12d";
    print var;
    output file = d:tema3.p12 off;
```

```
/* PROBLEMA 1 */;

    output file = d:tema4.p on;
    A = {1 2 3, 2 −1 −1, 2 −3 2, 5 −2 4};
    B = {1 2 3 6, 2 −1 −1 0, 2 −3 2 1, 5 −2 4 7};
    x = rank(A);
    y = rank(B);
    C = {6, 0, 1};
    D = {1 2 3, 2 −1 −1, 2 −3 2};
    z = rank(D);
    s1 = C/D;
    print "solución problema 1";
    print x;
    print y;
    print z;
    print s1;

/* PROBLEMA 2 */;

    A = {2 2 3 1, 3 −1 1 3, −2 3 1 −2, 1 5 3 −3, 2 7 3 −2};
    B = {2 2 3 1 1, 3 −1 1 3 2, −2 3 1 −2 0, 1 5 3 −3 2, 2 7 3
    −2 8};
    x = rank(A);
    y = rank(B);
    C = {2 2 3 1, 3 −1 1 3, −2 3 1 −2, 1 5 3 −3};
    D = {1, 2, 0, 2};
    z = rank(C);
    s2 = D/C;
    print "solución problema 2";
    print x;
    print y;
    print z;
    print s2;

/* PROBLEMA 3 */;

    A = {3 2 0, 0 1 −1, 1 0 1};
    B = {3 2 0 3, 0 1 −1 1, 1 0 1 0};
    x = rank(A);
    y = rank(B);
    D = {3, 1, 0};
    s3 = D/A;
```



```
    print "solución problema 3";
    print x;
    print y;
    print s3;

/* PROBLEMA 4 */;

    A = {2 1 3, 4 2 2};
    B = {2 1 3 1, 4 2 2 2};
    x = rank(A);
    y = rank(B);
    print "solución problema 4";
    print x;
    print y;

/* PROBLEMA 5 */;

    A = {2 1 1, 3 −2 3, 1 −3 2};
    B = {2 1 1 6, 3 −2 3 7, 1 −3 2 0};
    x = rank(A);
    y = rank(B);
    print "solución problema 5";
    print x;
    print y;

/* PROBLEMA 6 */;

    A = {1 2, 1 −3, 3 1};
    B = {1 2 5, 1 −3 −5, 3 1 5};
    x = rank(A);
    y = rank(B);
    C = {5, −5};
    D = {1 2, 1 −3};
    z = rank(D);
    s6 = C/D;
    print "solución problema 6";
    print x;
    print y;
    print z;
    print s6;

/* PROBLEMA 7 */;

    A = {1 −1 1 −1 1, −1 1 0 1 −2, 1 −1 2 −1 0, 2 −1 2 −1
    0, 3 −2 5 −2 1};
    B = {1 −1 1 −1 1 −1, −1 1 0 1 −2 2, 1 −1 2 −1 0 0, 2 −1
    2 −1 0 1, 3 −2 5 −2 1 0};
```

```
x = rank(A);
y = rank(B);
print "solución problema 7";
print x;
print y;

/* PROBLEMA 8 */;

A = {2 1 −2, 3 2 2, 5 4 3};
B = {2 1 −2 10, 3 2 2 1, 5 4 3 4};
x = rank(A);
y = rank(B);
C = {10, 1, 4};
s8 = C/A;
print "solución problema 8";
print x;
print y;
print s8;

/* PROBLEMA 9 */;

A = {1 −2 3 −2, 3 −7 −2 4, 4 3 5 2};
x = rank(A);
print "solución problema 9";
print x;

/* PROBLEMA 10 */;

A = {1 2 −3, 2 5 2, 3 −1 −4};
x = det(A);
print "solución problema 10";
print x;

/* PROBLEMA 11 */;

A = {1 2 −1, 2 5 2, 1 4 7, 1 3 3};
x = rank(A);
print "solución problema 11";
print x;

/* PROBLEMA 12 */;

A = {1 2 2, 3 −2 −1, 2 −5 3, 1 4 6};
B = {1 2 2 2, 3 −2 −1 5, 2 −5 3 −4, 1 4 6 0};
x = rank(A);
y = rank(B);
C = {2, 5, −4};
```



```
D = {1 2 2, 3 −2 −1, 2 −5 3};
z = rank(D);
s12 = C/D;
print "solución problema 12";
print x;
print y;
print z;
print s12;

/* PROBLEMA 13 */;

A = {2 1 −3, 3 −2 2, 5 −3 −1};
B = {2 1 −3 5, 3 −2 2 5, 5 −3 −1 16};
C = {5, 5, 16};
x = rank(A);
y = rank(B);
s13 = C/A;
print "solución problema 13";
print x;
print y;
print s13;

/* PROBLEMA 14 */;

A = {2 3, 1 −2, 3 5};
B = {2 3 4, 1 −2 1, 3 5 2};
x = rank(A);
y = rank(B);
print "solución problema 14";
print x;
print y;

/* PROBLEMA 15 */;

A = {1 1 0 0, 0 1 1 0, 1 0 0 1, 0 1 0 1};
x = rank(A);
print "solución problema 15";
print x;

/* PROBLEMA 16 */;

A = {3 −3 −2 −1, 0 3 −2 1, 1 1 3 5, 1 1 1 3};
x = rank(A);
print "solución problema 16";
print x;
```

```
/* PROBLEMA 17 */;
    A = {1 −2 2, 2 1 −2, 3 4 −6, 3 −3 12};
    x = rank(A);
    print "solución problema 17";
    print x;

/* PROBLEMA 18 */;
    A = {2 1, 1 1, 1 −1, 1 1};
    B = {2 1 0, 1 1 1, 1 −1 −1, 1 1 0};
    x = rank(A);
    y = rank(B);
    print "solución problema 18";
    print x;
    print y;
    output file = d:tema4.p off;
```


SOLUCIONES A LOS PROBLEMAS

TEMA 4.1

PROBLEMA 1

solución apartado 1a

79.000000

solución apartado 1b

−160.00000

solución apartdo 1c

1181.0000

PROBLEMA 2

solución apartado 2a

−61.000000

Cofactor de 7 = 61

solución apartado 2b

−32.000000

Cofactor de 6 = −32

PROBLEMA 3

solución apartado 3a

2.0000000

solución apartado 3b

1.0000000
10.000000
7.0000000
−1.0000000
4.0000000
3.0000000
−1.0000000
−2.0000000
−1.0000000

$$\mathrm{Adj}\,(A') = \begin{bmatrix} 1 & 1 & -1 \\ -10 & 4 & 2 \\ 7 & -3 & -1 \end{bmatrix}$$

solución apartado 3c

2.0000000	0.00000000	0.00000000
0.00000000	2.0000000	0.00000000
0.00000000	0.00000000	2.0000000

$$A.\mathrm{Adj}\,(A') = 2\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

solución apartado 3d

0.50000000	0.50000000	−0.50000000
−5.0000000	2.0000000	1.0000000
3.5000000	−1.5000000	−0.50000000

PROBLEMA 4

solución apartado 4a

20.000000
20.000000

solución apartado 4b

108.00000
108.00000

solución apartado 4c

55.000000
55.000000

PROBLEMA 5

solución apartado 5a

9.0000000
−3.0000000
3.0000000
−1.0000000
3.0000000
−11.000000
7.0000000
3.0000000
5.0000000

$$\mathrm{Adj}\,(A) = \begin{bmatrix} 9 & 3 & 3 \\ 1 & 3 & 11 \\ 7 & -3 & 5 \end{bmatrix}$$

solución apartado 5b

6.0000000
4.0000000
2.0000000
−3.0000000
−22.000000
7.0000000
3.0000000
−10.000000
1.0000000

$$\mathrm{Adj}\,(A) = \begin{bmatrix} 6 & -4 & 2 \\ 3 & -22 & 7 \\ 3 & 10 & 1 \end{bmatrix}$$

PROBLEMA 6

solución apartado 6a

−67620.000
−67620.000

$\det(A \cdot B) = \det(A) \cdot \det(B)$

solución apartado 6b

−1096.0000
−31.000000

$\det(A + B) \neq \det(A) + \det(B)$

PROBLEMA 7

solución apartado 7a

−1.0000000

Como son linealmente independientes forman una base $\mathbf{R}^2$

solución apartado 7b

0.00000000

Como son linealmente dependientes no forman base de $\mathbf{R}^3$

solución apartado 7c

32.000000

Como son linealmente independientes forman una base de $\mathbf{R}^4$

PROBLEMA 1

solución apartado 1a

1.0000000
−1.0000000
1.0000000
−1.0000000

Autovalores −1 y 1

0.70710678	−0.70710678	0.00000000	0.00000000
0.00000000	0.00000000	0.70710678	0.70710678
0.00000000	0.00000000	0.70710678	−0.70710678
0.70710678	0.70710678	0.00000000	0.00000000

Autovectores:

— Autovalor −1 $\begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ y $\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$

— Autovalor 1 $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

solución apartado 1b

4.0000000
4.0000000

Autovalores: 4

0.70710678	0.70710678
0.70710678	0.70710678

Autovectores: $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

solución apartado 1c

5.0000000
2.0000000
1.0000000

Autovalores: 1, 2 y 5

0.00000000	0.00000000	1.0000000
0.00000000	1.0000000	−1.0000000
1.0000000	−0.33333333	0.00000000

Autovectores:

— Autovalor 1 $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$

— Autovalor 2 $\begin{bmatrix} 0 \\ 3 \\ -1 \end{bmatrix}$

— Autovalor 5 $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

PROBLEMA 2

solución apartado 2a

−1.0000000	0.00000000	0.00000000	0.00000000
0.00000000	−1.0000000	0.00000000	0.00000000
0.00000000	0.00000000	1.0000000	0.00000000
0.00000000	0.00000000	0.00000000	1.0000000

$$P = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

solución apartado 2b

1.0000000	0.00000000	0.00000000
0.00000000	2.0000000	0.00000000
0.00000000	−4.4408921E-016	5.0000000

$$P = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 3 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

PROBLEMA 3

solución problema 3

10.000000
−1.0000000
−17.000000

La suma de los autovalores es −8

PROBLEMA 4

solución problema 4

6.0000000
2.0000000
3.0000000

Autovalores 2, 3 y 6

−0.40824829	−0.86602540	0.471450452
0.81649658	−4.4408921E-016	0.47140452
−0.40824829	0.86602540	0.47140452

$$P = \begin{bmatrix} -1 & 1 & -1 \\ 0 & 1 & 2 \\ 1 & 1 & -1 \end{bmatrix} \qquad D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

PROBLEMA 5

solución problema 5

1.0000000
1.0000000

1.0000000	−1.0000000
0.00000000	3.0000000E-020

No es diagonizable porque el orden de multiplicidad de autovector 1, es dos, y $\dim(V_1) = 1$

PROBLEMA 6

solución apartado 6a

6.0000000
12.000000
6.0000000

Autovalores de A: 6 doble y 12

−0.92847669	0.44876373	−81.948741
−0.37139068	−0.89752747	−32.379496
0.18569534	0.44876373	17.189748

$\dim(V_6) = 2$ es diagonalizable

solución apartado 6b

6.0000000
6.0000000
12.000000

Autovalores de B: 6 doble y 12

1.0000000	−1.0000000	0.19444444
0.00000000	2.7000000E-019	0.16666667
0.00000000	0.00000000	1.0000000

$\dim(V_6) = 1$ no es diagonalizable
Por tanto, A y B no son semejantes

PROBLEMA 7

solución apartado 7a

0.00000000
1.0000000
3.0000000

0.57735027	−0.70710678	0.40824829
0.57735027	−1.1300905E-018	−0.81649658
0.57735027	0.70710678	0.40824829

0.00000000	0.00000000	3.3306691E-016
0.00000000	1.0000000	0.00000000
0.00000000	0.00000000	3.0000000

$$P^{-1}\,A\,P = D \text{ siendo } P = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & -2 \\ 1 & 1 & 1 \end{bmatrix} \text{ y } D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

solución apartado 7b

2.0000000
−2.0000000
2.0000000

0.70710678	−0.70710678	0.00000000
0.00000000	0.00000000	1.0000000
0.70710678	0.70710678	0.00000000

−2.0000000	0.00000000	0.00000000
0.00000000	2.0000000	0.00000000
0.00000000	0.00000000	2.0000000

$$P^{-1}\,B\,P = D \text{ siendo } P = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} p' \text{ y } D = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

PROBLEMA 8

solución problema 8

2.0000000
2.0000000
3.0000000

1.0000000	−1.0000000	1.0000000
0.00000000	9.0000000E-020	1.0000000
0.00000000	0.00000000	1.0000000

$\dim(V_2) = 1$ no es diagonalizable

PROBLEMA 9

solución problema 9

−1.0000000
5.0000000

−0.89442719	−0.74535599
0.44721360	−0.74535599

Autovalores: −1 y 5

$$P = \begin{bmatrix} -2 & -1 \\ 1 & -1 \end{bmatrix}$$

PROBLEMA 10

solución apartado 10a

4.0000000
−2.0000000
−2.0000000

0.44721360	−55901699	0.50000000
0.44721360	0.55901699	1.5000000
0.89442719	1.1180340	1.0000000

Autovalores: −2 y 4

Base para V_{-2} $\begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$

Base para V_4 $\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$

solución apartado 10b

1.0000000
3.0000000
3.0000000

−0.89442719	0.00000000	1.0000000
0.44721360	−1.1180340	−0.50000000
−0.44721360	1.1180340	1.5000000

Autovalores: 1 y 3

Base para V $\begin{bmatrix} -2 \\ 1 \\ -1 \end{bmatrix}$

Base para V_3 $\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$

solución apartado 10c

1.0000000
1.0000000
1.0000000

1.0000000	−1.0000000	0.00000000
0.00000000	4.0000000E-020	0.00000000
0.00000000	0.00000000	1.0000000

Autovalores: 1

Base para V_1 $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

PROBLEMA 11

solución problema 11

2.0000000
2.0000000
3.0000000

1.0000000	1.0000000	1.4142136
0.00000000	−1.1000000E-019	1.4142136
0.00000000	1.1000000-019	−2.8284271

Autovalores: 2 y 3

Autovectores:

— Autovalor 2 $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

— Autovalor 3 $\begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}$

PROBLEMA 12

solución apartado 12a

1.0000000
6.8541020
0.14589803

Definida positiva

solución apartado 12b

2.0000000
−4.0000000
−1.0000000
−1.0000000

Indefinida

solución apartado 12c

3.0000000
−1.0000000
3.0000000

Indefinida

solución apartado 12d

−0.29843788
−6.7015621
−2.0000000

Definida negativa

TEMA 4.4

solución problema 1

3.0000000
3.0000000
3.0000000

sistema compatible con solución única

1.0000000
1.0000000
1.0000000

$x_1 = 1$; $x_2 = 1$; $x_3 = 1$

solución problema 2

4.0000000
4.0000000
4.0000000

sistema compatible con solución única

1.0000000
2.0000000
−2.0000000
1.0000000

$x_1 = 1$; $x_2 = 2$; $x_3 = -2$; $x_4 = 1$

solución problema 3

3.0000000
3.0000000

sistema compatible con solución única

1.0000000
0.00000000
−1.0000000

$x = 1$; $y = 0$; $z = -1$

solución problema 4

2.0000000
2.0000000
2.0000000

sistema compatible indeterminado

solución problema 5

2.0000000
3.0000000

sistema incompatible

solución problema 6

2.0000000
2.0000000
2.0000000

sistema compatible con solución única

1.0000000
2.0000000

$x = 1; \quad y = 2$

solución problema 7

4.0000000
4.0000000

sistema compatible indeterminado

solución problema 8

3.0000000
3.0000000

sistema compatible con solución única

1.0000000
2.0000000
−3.0000000

$x = 1; \quad y = 2; \quad z = -3$

solución problema 9

3.0000000

sistema homogéneo compatible indeterminadó

solución problema 10

61.000000

$x = 0, \quad y = 0, \quad z = 0$

sistema homogéneo compatible con la solución trivial única.

solución problema 11

2.0000000

sistema homogéneo compatible indeterminado

solución problema 12

3.0000000
3.0000000
3.0000000

sistema compatible con solución única

1.0000000
−3.0000000
−2.0000000

$x = 1; \quad y = -3; \quad z = -2$

solución problema 14

2.0000000
3.0000000

sistema incompatible

solución problema 15

4.0000000

sistema homogéneo compatible con la solución trivial única

solución problema 16

4.0000000

sistema homogéneo compatible con la solución trivial única

solución problema 17

3.0000000

sistema homogéneo compatible con la solución trivial única

solución problema 18

2.0000000
3.0000000

sistema incompatible

APLICACIONES ESTADISTICAS

Capítulo 5
ESTADISTICA (I). MEDIDAS DESCRIPTIVAS

5.1. Introducción

En este capítulo se definen, en primer lugar, los elementos básicos de GAUSS para realizar el análisis estadístico descriptivo de una variable, y, a continuación, se analiza la descripción conjunta de varias variables.

La representación gráfica de este tipo de variables se puede obtener mediante diagramas de barras e histogramas, que se construyen fácilmente usando los comandos gráficos BAR e HIST (véase ejemplos en capítulo 2).

5.2. Medidas de Centralización

— *Media Aritmética:* Dado un conjunto de datos numéricos X_1, ..., X_n se define la media aritmética por:

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i = \frac{1}{N}(X_1 + X_2 + ... + X_N)$$

La sentencia GAUSS adecuada es (calcula las medias de las *columnas*)

XBAR = MEANC (X);

donde X puede ser una matriz n×k y en ese caso XBAR será el k×1 vector de medias. En el caso de una sola variable, X será un vector y XBAR un escalar.

Una forma alternativa de computar la media de una matriz de datos es usar el comando SUMC que suma las columnas de una matriz, del modo siguiente:

```
XBAR = SUMC(X) / ROWS (X);
```

donde se divide la suma de cada columna por el número de filas (observaciones) de la matriz.

— *Mediana*

La mediana es aquel valor de la distribución tal que, ordenados por orden de magnitud los datos, el 50% de ellos es menor que ella y el 50% mayor.

En realidad, la mediana es un caso particular del concepto de cuantil de una distribución.

Los *cuantiles* son aquellos valores que dividen la distribución en partes iguales. Así, tenemos los *cuartiles*, que son aquellos tres valores que dividen la distribución en 4 partes iguales. El primero de ellos tiene el 25% de los valores por debajo y el 75% por arriba, el segundo de ellos (Mediana) 50% abajo, 50% arriba, y el tercero, 75% abajo y 25% arriba. De igual forma podrían definirse los *deciles* (10 partes), los *percentiles* (100 partes), etc.

A continuación se presenta un procedimiento (que puede almacenarse con extensión .G), que calcula los cuantiles de un vector de datos.

Procedimiento «cuantil» para el cálculo de cuantiles de una distribución.

```
/* Cálculo de cuantiles de una distribución
** data = vector de datos (t*1)
** theta = vector de cuantiles solicitado (k*1)
**
**
*/
proc cuantil (data, theta);
local y, t, n, x, est;
x = data;
y = sortc(x, 1);
t = rows(y);
n = trunc((theta ./ 100) .* t) + 1;
est = (y[n, 1] + y[n - 1, 1]) ./ 2;
retp(est);
endp;
```


Veamos un ejemplo en el cual se generan 100 números N(0, 1) y se calculan los tres cuartiles de esa distribución (recuerden que el segundo cuartil es la Mediana).

```
RNDSEED 39875; /* Fija una semilla para la generación de números
aleatorios */
DATOS = RNDN(100, 1); /* Se generan los 100 datos */
VAL = {25 50 75}; /* Vector de cuantiles */
CUART = CUANTIL (DATOS, VAL);
CUART';
-0.8458   0.0704   0.6410
```

Con éste procedimiento, puede calcularse cualquier cuantil de la distribución de interés.

Una medida de centralización adicional de interés es la *Moda* o valor más frecuente de la distribución. Su cálculo con GAUSS no es inmediato, aunque existe el procedimiento general FREQSTAT que computa esta cantidad (es un módulo de las aplicaciones generales de GAUSS y habitualmente no se incluye en la SV).

— *Media Geométrica*

Se define como la raíz N-ésima del producto de los N valores de una distribución. Es decir

$$G = \sqrt[N]{X_1 \cdot X_2 \dots X_N}$$

Para su cálculo en GAUSS, usaremos la propiedad de que el logaritmo de la media geométrica es igual a la media aritmética de los logaritmos de la variable. No existe en GAUSS una sentencia directa que compute esta cantidad, así que presentaremos una función que la compute.

```
fn MEDG (X) = EXP(MEANC (LN(X)));
```

Nótese que calculamos el antilogaritmo natural (función EXP) de la media aritmética de los logaritmos (naturales) de la variable (X es un vector columna).

Para utilizar esta función basta con seguir los principios generales ya expuestos. En este caso, véase el ejemplo:

```
DATOS = {5, 8, 11, 24, 6};
MG = MEDG(DATOS);
MG;
9.1277
```

— *Medias Recortadas*

En ocasiones, los datos pueden presentar gran cantidad de valores atípicos (lejos de la media) lo cual ocasiona que las medidas habituales de centralización estén distorsionadas por esos datos. Un enfoque que pretende minimizar la influencia de esos datos extremos a la hora de analizar las características de centralización de una variable aconseja el uso de Medias Recortadas («Trimmed Means») que consiste en calcular la media de subconjuntos centrales de datos. Concretamente, definimos la media recortada a nivel α, como la media aritmética de los datos que restan después de eliminar de la muestra inicial el α % de las observaciones de cada extremo. Por tanto, se eliminan el 2α % en total. Así, una media recortada de nivel 10 es la media aritmética de los valores resultantes al eliminar el 10% superior y el 10% inferior de los valores totales (que se suponen ordenados de mayor a menor).

Específicamente, si los datos ordenados los escribimos $X_{(i)}$, la media recortada a nivel α, $T(\alpha)$, se calcula por:

$$T(\alpha) = \frac{1}{n(1-2\alpha)} \sum_{i=n\alpha+1}^{n-n\alpha} X_{(i)}$$

lo cual es equivalente simplemente a tomar la media de los datos «centrales» de la distribución, con la muestra recortada correspondiente.

A continuación se muestra un procedimiento que calcula medias recortadas

```
/*
* PROCEDIMIENTO RMEDIA
* CALCULA MEDIAS RECORTADAS (TRIMMED) DE UN
  VECTOR DE DATOS
*
* FORMATO RMEDIA (DATA, ALFA)
*
* DATA ES EL VECTOR DE DATOS
* ALFA ES EL NIVEL DE RECORTE DESEADO EN CADA
  COLA DE LA DISTR.
*
* EJEMPLO RES=RMEDIA (DATOS, 10);
* SE RECORTA EL 10% EXTREMO EN CADA LADO (TOTAL
  20%)
*/

proc rmedia (data, alfa);
```



```
local x, beta, cuart, meant, i;
beta = {0, 0};
x = data;
beta [1, 1] = alfa;
beta [2, 1] = 100-alfa;
cuart = cuantil (x, beta); /* SE USA EL PROC CUANTIL */
i = 1;
do while i <= rows(x);
if x[i, 1] <= cuart[1, 1];
x[i, 1] = miss(x[i, 1], x[i, 1]);
elseif x[i, 1] >= cuart[2, 1];
x[i, 1] = miss(x[i, 1], x[i, 1]);
endif;
i = i + 1;
endo;
meant = meanc (packr(x));
retp (meant);
endp;
```

La lógica del procedimiento RMEDIA es la siguiente:

- Se cargan como datos el vector de observaciones (DATA) y el nivel de corte ALFA (α) para cada extremo.
- A continuación se definen las variables locales que se van a emplear en el procedimiento. Se crean BETA y X como variables auxiliares. Beta contiene los cuantiles superior e inferior (10%, 90%) cuyos valores numéricos concretos se obtienen llamando al procedimiento CUANTIL. Los resultados (un vector 2×1) se almacenan en CUART.
- Se efectúa una comparación usando el bucle «do while» en cada elemento de X para ver si es de un valor inferior (igual) o superior (igual) a los percentiles correspondientes. Aquellos valores que cumplen esta condición se sustituyen por el símbolo de MISSING VALUE.
- Finalmente, se calcula la media de los valores restantes, que se obtiene eliminando de X los MISSING VALUES (comprimiendo la matriz con el operador PACKR) y calculando la media de los restantes.
- El valor así obtenido, MEANT, es la media recortada de orden α, que se obtiene como resultado final.

5.3. Medidas de dispersión

— *Varianza:* Es la medida de dispersión asociada a la media aritmética. Su raíz cuadrada, la *desviación típica* suele utilizarse con mayor frecuencia al ser más fácil de interpretar (tiene las mismas unidades que la variable original).

Se define como

$$S^2 = (N-1)^{-1} \Sigma (x_i - xmed)^2 = (N-1)^{-1} [(x_1 - xmed)^2 + (x_2 - xmed)^2 + \ldots + (x_N - xmed)^2]$$

La desviación típica es

$$S = (S^2)^{1/2}$$

y puede interpretarse como el promedio de las desviaciones de las observaciones a su media.

La sentencia GAUSS adecuada para el cálculo de la desviación típica es (de nuevo X puede ser una matriz)

```
S = STDC(X);
```

Una forma alternativa de computar la varianza de un vector de datos es

```
S2 = SUMC((X - MEANC(X))^ 2) ./ (ROWS(X) -1);
```

y la desviación típica será:

```
S = SQRT(S2);
```

— *Coeficiente de variación:* Se denomina coeficiente de variación al cociente

$$CV = S / \bar{X}$$

que tiene la ventaja de ser adimensional. Es una medida relativa de variabilidad que se suele emplear para comparar dispersiones de variables con unidades diferentes.

El cálculo con GAUSS es sencillo:

```
CV = STDC(X)/MEANC(X);
```

si X es un vector y

```
CV = STDC(X) ./ MEANC(X);
```

si X es una matriz.

— *Mediana de desviaciones absolutas*

Es la medida de dispersión que se utiliza si tenemos como medida de centralización de referencia a la Mediana.

Se define como la Mediana de las desviaciones en valor absoluto de todos los valores de la variable, con respecto a la Mediana global; es decir,

$$MEDIA = \underset{i}{\text{mediana}} |X_i - Med|$$

No existe en GAUSS un comando que calcule directamente esta cantidad, pero usando el procedimiento CUANTIL ya expuesto, es sencillo obtener la MEDA.

Una forma simple sería usar dos sentencias, por ejemplo: (X contiene el vector columna de datos)

```
ME1 = ABS(X - CUANTIL (X, 50));
MEDA = CUANTIL(ME1, 50); /* Se calcula la Mediana
                             del vector anterior */
```

Una forma más compacta y eficiente de realizar la anterior tarea es construir una función que realice simultáneamente ambas operaciones.

```
fn MEDA(X) = CUANTIL((ABS(X-CUANTIL(X, 50))), 50);
```

Se permite usar este tipo de llamadas recurrentes a procedimientos, que es una de las grandes ventajas de GAUSS.

El uso de la función anterior es sencillo, como muestra el siguiente ejemplo

```
DATOS = {5, 8, 11, 24 ,6};
ME = MEDA(DATOS);
ME;
1.5000;
```

Recuerden que si conocemos la Mediana y la MEDA de los datos, sabemos que el 50% están en el intervalo (Med ± MEDA) y el 50% fuera de él.

5.4. Medidas de forma

— *Medidas de asimetría:* El objetivo de las medidas de asimetría es elaborar un indicador que permita establecer el grado de simetría de los datos con respecto a una medida de centralización.

Si la medida elegida es la *Media,* entonces el coeficiente más utilizado es el β_1 definido como:

$$\beta_1 = \frac{\Sigma(X_i - \bar{X})^3}{nS^3}$$

para una distribución simétrica, éste coeficiente vale cero. Para distribuciones asimétricas a la derecha es positivo y para distribuciones asimétricas a la izquierda es negativo.

Su cálculo, mediante una función de GAUSS, ya se explicó anteriormente (ver 1.3.1) y es

```
fn asim(X) =
((X-SUMC((X-MEANC))^3)/(ROWS(X)*(STDC(X)^3));
```

Nótese que el coeficiente así obtenido es adimensional.

Si la medida de centralización es la *Mediana,* entonces el coeficiente adecuado es llamado coeficiente de asimetría de Bowley, que usa los cuartiles de la distribución. Su expresión es:

$$A_B = \frac{C_3 + C_1 - 2Me}{C_3 - C_1}$$

donde C_3 y C_1 son el tercer y primer cuartil respectivamente.

Usando el procedimiento CUANTIL, su cálculo será:

```
AB = (CUANTIL(X, 75) + CUANTIL(X, 25)
        -2*CUANTIL(X, 50))/
     (CUANTIL(X, 75) - CUANTIL(X, 25));
```

Si la distribución es simétrica, el valor de AB será cero, y la interpretación de su signo es la misma que en el coeficiente anterior.

— *Medidas de Apuntamiento:* Las medidas de apuntamiento se centran en estudiar la distribución de frecuencias de la distribución en su centro y en las colas, y en compararla con otra distribución de referencia, que suele ser la Gaussiana o Normal. Para una distribución de ese tipo, el coeficiente β_2 definido como:

$$\beta_2 = \frac{\Sigma(X_i - \bar{X})^4}{nS^4}$$

toma el valor 3.

Si el valor obtenido es menor que 3, a la distribución se le denomina *platicúrtica* y si es superior a 3 *leptocúrtica.* Su cálculo es sencillo.


```
func CUR(X) = SUMC ((X - MEANC (X)^4)) /
                (ROWS(X)*(STDC(X)^4));
```

Pueden combinarse estas medidas de forma en un procedimiento general, como ya se mostró en apartados anteriores.

CASO PRACTICO 1

(Este problema está adaptado del libro de D. Peña, *Estadística 1, Modelos y Métodos.* "Alianza Universidad (1989), con el permiso del autor.)

En 1879 Michelson obtuvo los siguientes valores para la velocidad de la luz en el aire (damos los resultados restando 299.000 de los datos originales, en km/seg para facilitar su manejo): 850, 740, 900, 1070, 930, 850, 950, 980, 980, 880, 1000, 980, 930, 650, 760.

En 1882 Newcomb utilizando otro procedimiento obtuvo: 883, 816, 778, 796, 682, 711, 611, 599, 1051, 781, 578, 796, 774, 820, 772.

Se pide:

a) Calcular las siguientes medidas de centralización: Media, Media Geométrica, Mediana y Media Recortada de orden 10.
b) Calcular las medidas de dispersión: Desviación típica, Coeficiente de variación, MEDA, así como medidas de asimetría y apuntamiento.
c) Construir histogramas y gráficos para cada variable.
d) Con los resultados anteriores, ¿qué concluciones pueden extraerse?

```
/* PROBLEMA 1 MICHELSON-NEWCOMB
* SE INTRODUCEN LOS DATOS. LOS DE MICHELSON EN MD Y LOS DE
* NEWCOMB EN ND */;

md = {850, 740, 900, 1070, 930, 850, 950, 980, 980, 880, 1000, 980,
930, 650, 760};

nd = {883, 816, 778, 796, 682, 711, 611, 599, 1051, 781, 578, 796, 774,
820, 772};

ti = seqa (1, 1, rows (md));

/*
* SECCION DE FUNCIONES
*/;

fn as(x)=sumc((x-meanc(x))^3)/(rows(x)*(stdc(x)^3));
```

```
fn cu(x)=sumc((x-meanc(x))^4)/(rows(x)*(stdc(x)^4));
fn medg(x)=exp(meanc(ln(x)));
fn meda(x)=cuantil((abs(x-cuantil(x, 50))), 50);

/*
*
* SE CALCULAN LOS ESTADISTICOS SOLICITADOS
*
*/;


men=meanc(nd);mem=meanc(md);                    /* medias */
sdn=stdc(nd);sdm=stdc(md);                 /* desv. típicas */
medn=cuantil(nd, 50);medm=cuantil(md,50);        /* medianas */
medgn=medg(nd);medgm=medg(md);      /*medias geométricas*/
medrn10=rmedia(nd,10);medrm10=rmedia(md,10);
    /* medias recortadas orden 10*/
cvn=sdn/men;cvm=sdm/mem;          /*coeficientes de variación*/
medan=meda(nd);medam=meda(md);                     /* medas */
asin=as(nd);asim=as(md);                     /* coef. asimetría */
curn=cu(nd);curm=cu(md);             /* coef. apuntamiento */


/*
* GRAFICOS
*/


library qgraph;
graphset;
beggraph;
window(2, 2);
title("MICHELSON");xy(ti,md);
title("NEWCOMB");xy(ti,nd);
_qmajor=3; qnum=0;
et1= " media = " $+ ftocv(mem, 3,2) ;xlabel(et1);
title("MICHELSON");{c,m,f}=hist(md,cint(sqrt(rows(md))));
et2= " media = " $+ ftocv(men,3,2);xlabel(et2);
title("NEWCOMB");{c,m,f}=hist(nd,cint(sqrt(rows(nd))));
endgraph;
output file = out.pr1 on;
/*
* PRESENTACION DE RESULTADOS
*/
```



```
print;
"                         DATOS DE MICHELSON      DATOS DE NEWCOMB ";
"                         __________________      ________________ ";

format 6,3;
print;
"............... MEDIA         "mem"             "men;
"............... DESVTI        "sdm"             "sdn;
".......... MEDIANA            "medm"            "medn;
"........... MEDGEO            "medgm"           "medgn;
"........ COEF VAR             "cvm"             "cvn;
"........... MRECO10           "medrm10"         "mdrn10;
"................ MEDA         "medam"           "medan;
"....... ASIMETRIA             "asim"            "asin;
"......... CURTOSIS            "curm"            "curn;
"...... V. MAXIMO              "maxc(md)"        "maxc(nd);
"....... V. MINIMO             "minc(md)"        "minc(nd);
/*
* fin
*/
output off;
```

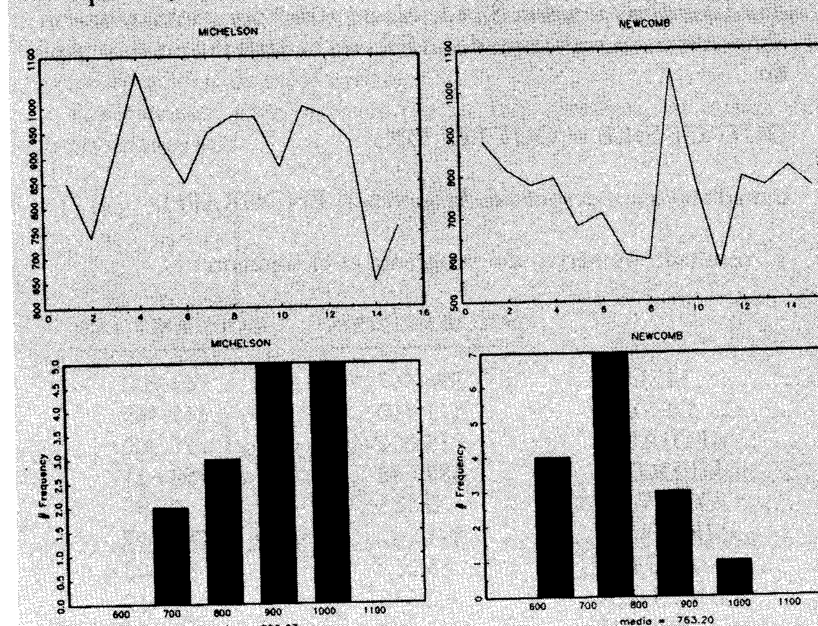

FIGURA 6. *Caso práctico 1.*

Solución

Con el programa que se ha mostrado, se responde al problema. El programa tiene cinco secciones.

a) *Introducción de datos:* se crean dos vectores 15×1 que contienen los datos. También se crea la secuencia aritmética T1 de 1 a 15, que se utilizará posteriormente.
b) Funciones: se introducen las funciones necesarias para el cálculo de los coeficientes de asimetría (as), curtosis (cu), media geométrica (medg) y MEDA (meda).
c) *Cálculo de Estadísticos:* usando tanto las funciones propias como los comandos de GAUSS, se calculan los estadísticos solicitados.
d) *Gráficos:* se generan cuatro gráficos; dos histogramas y dos diagramas de líneas para cada vector de observaciones. Se emplea la funciónb FTOCV que transforma un valor numérico en su equivalente alfabético (es decir, el número 103.5 se transforma en cinco caracteres 1, 0, 3, ..., 5 que se consideran como alfabéticos) para poderlos incluir dentro del título del eje horizontal del histograma, que sólo admite caracteres alfabéticos.
e) *Resultados:* De forma simple, se presentan por pantalla los resultados del análisis anterior. Si, además de verlos por pantalla, desean almacenar estos resultados de un fichero, bastaría incluir el comando

OUTPUT FILE = OUT.PR1 ON;

inmediatamente después de la sentencia ENDGRAPH.

El resultado numérico del programa es el siguiente:

	DATOS DE MICHELSON	DATOS DE NEWCOMB
.... MEDIA	896.667	763.200
.... DESVTI	111.910	119.469
.... MEDIANA	915.000	776.000
.... MEDGEO	889.648	754.621
.... COEF VAR	0.125	0.157
.... MRECO10	894.167	744.667
.... MEDA	65.000	42.000
.... ASIMETRIA	−0.635	0.430
.... CURTOSIS	2.480	3.134
.... V. MAXIMO	1070.000	1051.000
.... V. MINIMO	650.000	578.000

Comentario: Las medias son bastante diferentes, siendo la de Michelson claramente mayor y la dispersión es más marcada en Newcomb. Los datos de Newcomb (N) presentan un valor muy alto, mientras que los de Michelson (M) parecen más homogéneos. En ambos casos, hay asimetría, en el caso N a la derecha y M a la izquierda.

Las medidas más resistentes a datos extremos (Mediana y Media Recortada) dan el mismo mensaje: en general, las mediciones de M dan valores más altos que las de N.

5.5. Descripción conjunta de múltiples variables

Cuando tratamos con un conjunto de variables, nos interesará conocer sus características conjuntas, además de sus características individuales.

Para el análisis por separado de cada variable, pueden emplearse los métodos expuestos en apartados anteriores. En este apartado nos centraremos en las características de la *distribución conjunta* de las variables.

— *Medidas de Centralización:* Si tenemos un conjunto k-dimensional de variables, una medida simple de centralización es el *vector de medias* de las k variables.

Por ejemplo, para un conjunto de tres variables, su matriz de observaciones será:

$$X = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \vdots & \vdots & \vdots \\ a_n & b_n & c_n \end{bmatrix}$$

y su vector de medias

$$\bar{X} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \Sigma a_i \\ \Sigma b_i \\ \Sigma c_i \end{bmatrix}$$

El comando GAUSS adecuado es el ya visto MEANC, donde si escribimos

y = MEANC(X);

si X es una matriz N×k, y será un vector k×1 que contendrá la media de cada *columna* de X.

— *Medidas de Dispersión:* Una forma útil de observar las interrelaciones de las variables, así como su propia dispersión es la *matriz de varianzas-covarianzas,* que es una matriz cuadrada (k×k) simétrica que tiene en la diagonal principal las varianzas de las observaciones y fuera de ellas las covarianzas entre las variables. En el ejemplo tridimensional anterior sería:

$$M = \begin{bmatrix} S_a^2 & S_{ab} & S_{ac} \\ S_{ba} & S_b^2 & S_{bc} \\ S_{ca} & S_{cb} & S_c^2 \end{bmatrix}$$

donde S_a^2 es varianza de a y S_{ab} covarianza de a y b definida como

$$S_{ab} = \frac{1}{N} \Sigma\, (a_i - \text{amed})(b_i - \text{bmed})$$

Usando la notación matricial anterior es fácil ver que

$$M = \frac{1}{N} (X - \bar{X})'(X - \bar{X})$$

GAUSS dispone de varias sentencias para obtener esta matriz o bien otras muy relacionadas con ésta.

Así, para obtener de varianzas-covarianzas M, el comando es

```
M = VCX(X);
```

donde X es una matriz de datos N×k, y M es la matriz k×k de varianzas-covarianzas.

En ocasiones puede ser interesante calcular la *matriz de momentos* de la distribución, definida como

$$MN = X'X$$

que se obtiene con el comando

```
MN = MOMENTD (DATOS, VARIABLES);
```

Donde DATOS es el nombre del conjunto de datos y VARIABLES es el nombre de las variables de ese conjunto de datos cuya matriz MN se desea computar.

Adicionalmente, suele ser de interés computar la *matriz de correlaciones* que nos da la misma información que la de varianzas-covarianzas, pero de forma adimensional. Tendremos, por tanto, una

matriz cuadrada simétrica cuya diagonal principal estará compuesta de unos y fuera de ella, los coeficientes de correlación a pares entre las variables.

En el ejemplo tridimensional tendremos

$$R = \begin{bmatrix} 1 & r_{ab} & r_{ac} \\ r_{ba} & 1 & r_{bc} \\ r_{ca} & r_{cb} & 1 \end{bmatrix}.$$

donde r_{ab} es el coeficiente de correlación entre a y b definido como

$$r_{ab} = \frac{S_{ab}}{S_a S_b}$$

y mide el grado de asociación lineal entre a y b (nótese que por construcción $|r| \leq 1$).

Para obtener esta matriz, usamos la sentencia

```
R = CORRX(X);
```

donde X es N×k y R es la matriz de correlaciones.

— *Varianza generalizada*

Además de las matrices anteriores, resulta conveniente tener una medida global de la variabilidad conjunta de las k variables de interés. Esta medida es la *varianza generalizada* y es el determinante de la matriz M de varianzas y covarianzas. Su cálculo es sencillo

```
VG = DET(VCX(X));
```

usando el comando DET que computa el determinante.

La interpretación de esta medida es que será mayor, en general, cuando menos relacionadas estén las variables (y la «dispersión global» sea mayor) y más pequeña cuando el grado de asociación lineal sea alto.

CASO PRACTICO 2

Se obtienen los datos del cuadro 1 sobre tasas de variación del PIB (en volumen), tasa de paro y tasa de variación de la inflación para USA, CEE y Japón durante el período 1983-1988. (Fuente: OCDE).

Se pide:

a) Calcular las medias de cada variable en cada uno de los casos.
b) Calcular coeficientes de variación y varianzas generalizadas.
c) Calcular la matriz de correlación de cada indicador económico dentro de cada país.

CUADRO 1 de datos

USA			CEE			JAPON		
PIB	PARO	IPC	PIB	PARO	IPC	PIB	PARO	IPC
3.6	9.5	3.2	1.7	9.6	8.0	3.3	2.6	1.8
6.8	7.4	4.3	2.4	9.5	6.7	5.1	2.7	2.3
3.4	7.1	3.5	2.5	9.5	6.1	4.9	2.6	2.0
2.7	6.9	1.9	2.6	9.5	3.5	2.5	2.8	0.6
3.7	6.1	3.7	2.7	9.3	3.2	4.6	2.9	0.1
4.4	5.4	4.1	3.7	8.7	3.6	5.8	2.5	0.7
3.0	5.2	4.8	3.2	7.9	5.2	4.8	2.2	2.3

(1983-1989)

```
/*
* PROBLEMA 2   DATOS MACROECONOMICOS
*
* LA PRIMERA COLUMNA ES EL PIB
* SEGUNDA PARO Y TERCERA IPC
* (TASAS DE VARIACION EN TODOS LOS CASOS)
* PERIODO 1983-1989
*
*/

USA={3.6 9.5 3.2, 6.8 7.4 4.3, 3.4 7.1 3.5, 2.7 6.9 1.9, 3.7
6.1 3.7, 4.4 5.4 4.1, 3.0 5.2 4.8};

CEE={1.7 9.6 8.0, 2.4 9.5 6.7, 2.5 9.5 6.1, 2.6 9.5 3.5, 2.7
9.3 3.2, 3.7 8.7 3.6, 3.2 7.9 5.2};

JAP={3.3 2.6 1.8, 5.1 2.7 2.3, 4.9 2.6 2.0, 2.5 2.8 0.6, 4.6
2.9 0.1, 5.8 2.5 0.7, 4.8 2.2 2.3};
```



```
USAMED=MEANC(USA);
CEEMED-MEAND(CEE);
JAPMED=MEANC(JAP);
MUSA=VCX(USA);
MCEE=VCX(CEE);
MJAP=VCX(JAP);
RUSA=CORRX(USA);
RCEE=CORRX(CEE);
RJAP=CORRX(JAP);
VGUSA=DET(MUSA);
VGCEE=DET(MCEE);
VGJAP=DDET(MJAP);
CVUSA=SQRT(DIAG(MUSA)) ./ USAMED;
CVCEE=SQRT(DIAG(MCEE)) ./ CEEMED;
CVJAP=SQRT(DIAG(MJAP))  ./ JAPMED;
"MEDIAS  PIB  PARO  IPC";
"    ";
"USA"    USAMED[1, 1]   USAMED[2, 1]   USAMED[3, 1]
"        " ;
"CEE"    CEEMED[1, 1]   CEEMED[2, 1]   CEEMED[3, 1];
"        " ;
"JAPON"  JAPMED[1, 1]   JAPMED[2, 1]   JAPMED[3, 1];
"------------------------------------------------------------ ";
" COEF VAR     PIB   PARO   IPC";
"  ";
"USA"      CVUSA[1, 1] CVUSA]2, 1] CVUSA[3, 1];
"  ";
"CEE";     CVCEE[1, 1] CVCEE]2, 1] CVCEE[3, 1];
"  ";
"JAPON"    CVJAP[1, 1] CVJAP[2, 1] CVJAP[3, 1];
"WAIT;
"VARIANZA GENERALIZADA";
"  ";
"USA";        VGUSA;
"  ";
"CEE"         VGCEE;
"  ";
"JAPON"       VGJAP;
"------------------------------------------------------------ ";
WAIT;
CLS;
"MATRIZ DE CORRELACIONES USA";
```

```
RUSA;
" ------------------------------------------------------------------------";
"MATRIZ DE CORRELACIONES CEE";
RCEE;
" ----------------------------------------------------------------------- ";
"MATRIZ DE CORRELACIONES JAPON";
RJAP;
```

Para obtener los gráficos correspondientes, habrá que añadir el bloque de gráficos, como sigue:

```
/*
* SECCION DE GRAFICOS
*/;
TI=SEQA(83, 1, 7);
LIBRARY QGRAPH;
GRAPHSET;
BEGGRAPH;
WINDOW(3, 3);
_QMAJOR=1;
XLABEL ("PIB USA");
XY (TI, USA[., 1]);
XLABEL ("PARO USA");
XY (TI, USA[., 2]);
XLABEL ("IPC USA");
XY (TI, USA[., 3]);
XLABEL ("PIB CEE");
XY (TI, CEE[., 1]);
XLABEL ("PARO CEE");
XY (TI, CEE[., 2]);
XLABEL ("IPC CEE");
XY (TI, CEE[., 3]);
XLABEL ("PIB JAP");
XY (TI, JAP[., 1]);
XLABEL ("PARO JAP");
XY (TI, JAP[., 2]);
XLABEL ("IPC JAP");
XY (TI, JAP[., 3]);
ENDGRAPH;
```


Los resultados del programa son:

MEDIAS	PIB	PARO	IPC
USA	3.9429	6.8000	3.6429
CEE	2.6857	9.1429	5.1857
JAPON	4.4286	2.6143	1.4000

COEF VAR	PIB	PARO	IPC
USA	0.3477	0.2145	0.2565
CEE	0.2349	0.0686	0.3551
JAPON	0.2560	0.0867	0.6494

VARIANZA GENERALIZADA	
USA	1.9288
CEE	0.0981
JAPON	0.0301

MATRIZ DE CORRELACIONES USA

1.0000	0.1284	0.4394
0.1284	1.0000	−0.4343
0.4394	−0.4343	1.0000

MATRIZ DE CORRELACIONES CEE

1.0000	−0.7315	−0.6973
−0.7315	1.0000	0.2778
−0.6973	0.2778	1.0000

MATRIZ DE CORRELACIONES JAPON

1.0000	−0.3583	0.1827
−0.3583	1.0000	−0.6062
0.1827	−0.6062	1.0000

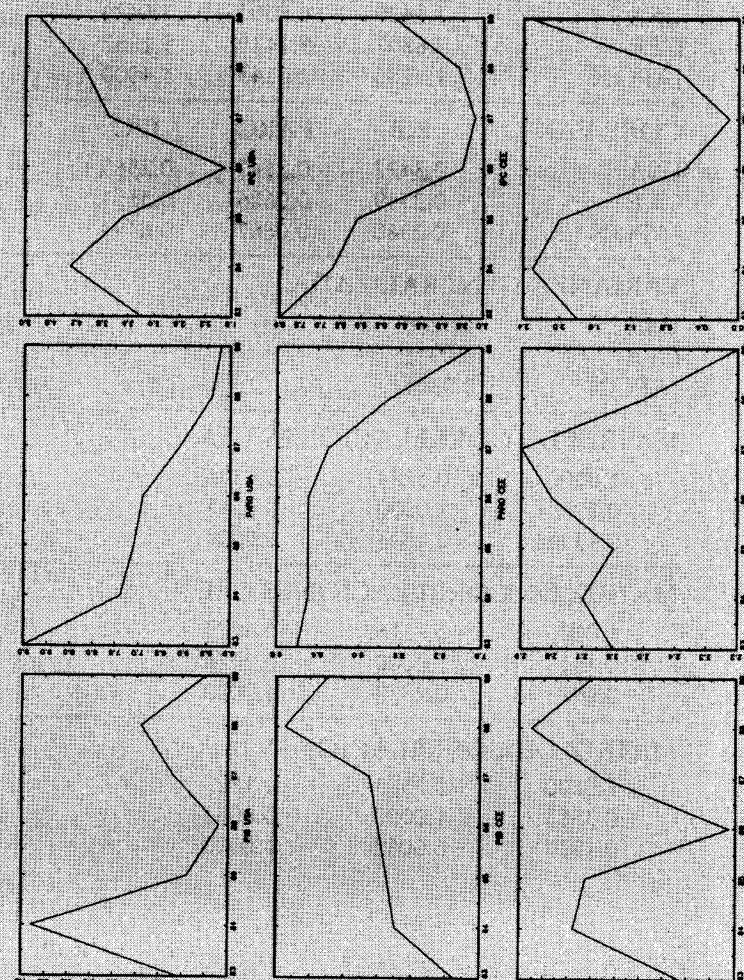

FIGURA 7. *Caso práctico 2.*

Comentario

Se observa como, en el período analizado, fue el Japón el que presenta una mejor evolución, con mayor crecimiento medio del PIB, menor tasa de paro y menor IPC. La peor posición en esos tres apartados corresponde a la CEE, estando USA en una posición intermedia.

En lo referente a dispersión relativa, es USA la que presenta mayor grado de ella en PIB y PARO y Japón en IPC, siendo la CEE la más estable en PIB y PARO y USA la más estable en IPC.

La varianza generalizada indica que de nuevo es Japón la economía en la que estas tres variables están más estrechamente relacionadas, mientras que en USA es donde se da el mayor grado de dispersión, quedando la CEE en posición intermedia.

En cuanto a las correlaciones encontradas hay que destacar los siguientes puntos:

a) La correlación entre PIB y PARO es fuerte y negativa (−0.732) en la CEE y algo menos en Japón (−0.358), mientras que es pequeña y positiva en USA (0.128).
b) La correlación entre PIB e IPC es alta y negativa en la CEE (−0.697) y positiva en USA (.439) y pequeña y positiva en Japón (0.183).
c) La correlación entre PARO e IPC es negativa y fuerte en Japón (−0.606) así como algo más fuerte en USA (−0.434), mientras que aparece como no grande (.278) y positiva en la CEE.

Capítulo 6

ESTADISTICA (II). REGRESION LINEAL

6.1. Introducción

Este capítulo requiere para su uso un mayor nivel de conocimientos que los anteriores. Para poder aprovecharlo plenamente, el lector debería estar familiarizado con los fundamentos de Probabilidad, Distribuciones y Contraste de Hipótesis a un nivel introductorio como el de Rohatgi (1976), Lindgren (1976) o Peña (1986). Asimismo sería conveniente que el lector consultara manuales clásicos de Econometría como Maddala (1977), Johnston (1984) o Judge et al. (1988).

6.2. Formulación del modelo

Un objetivo habitual de investigación es la especificación de una relación funcional entre una variable de interés y otras que la influyen de alguna manera, más un término de error que refleje errores de medida, variables omitidas y otros elementos no explícitamente considerados. Es decir, una relación de la forma expresable como $\mathbf{y} = f(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k) + \mathbf{e}$. Llamaremos a la variable $\mathbf{y}$ el «regresando» de la ecuación y a las k variables $\mathbf{x}$ los «regresores» de la ecuación. Supondremos en lo que sigue que la relación es lineal (o linealizable) entre las variables y que se dispone de una muestra de tamaño $\mathbf{n}$. Es decir, el modelo que emplearemos será:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_k x_{ki} + e_i \quad i = 1, 2, ..., n$$

Este modelo puede escribirse de forma compacta como

$$y = X\beta + e$$

donde y es un vector $n \times 1$, X es una matriz $n \times (k+1)$, e es un vector $n \times 1$, y β es un vector de parámetros a estimar de dimensión $(k+1) \times 1$.

Formularemos las siguientes hipótesis sobre el modelo:

a) Los residuos se distribuyen idéntica e independientemente con media cero y varianza constante σ^2, es decir

$$E(e) = 0, \quad V(e) = \sigma^2 I$$

b) Las variables **x** son fijas en muestras repetidas e independientes de los errores **e**

$$E(X'e) = 0$$

c) Las **x** son linealmente independientes, y por tanto el rango de **X** es k + 1 y $(X'X)^{-1}$ existe.

Bajo estas condiciones, el estimador lineal insesgado de mínima varianza es el minimocuadrático, que viene dado por

$$\beta^* = (X'X)^{-1}X'y$$

su matriz de varianzas covarianzas es

$$V(\beta^*) = \sigma^2(X'X)^{-1}$$

y la matriz de correlaciones se calcula como

$$R(\beta^*) = SS^{-1}V(\beta^*)SS^{-1}$$

donde la matriz diagonal **SS** contiene la raíz cuadrada de la diagonal principal de la matriz $V(\beta^*)$. Un estimador centrado de la varianza residual se computa como

$$s^2 = (n-k-1)^{-1}(y - x\beta^*)'(y - x\beta^*)$$

nótese que los residuos de la regresión se calculan

$$e^* = y - X\beta^*$$

Una medida simple del grado de ajuste del modelo a los datos es el coeficiente de determinación que se calcula como

$$R^2 = 1 - \frac{(e^{*\prime}e^*)}{(y-ymed)'(y-ymed)}$$

donde **ymed** es la media muestral de la **y**.

Los comandos de GAUSS para computar las expresiones anteriores son sencillos. Calcularemos todos los estadísticos mencionados hasta ahora, mediante el siguiente programa. Supondremos que los datos del regresando ya estan almacenados en el vector Y ($n \times 1$) y los de los regresores (incluyendo un vector de unos para la constante) en la matriz X ($n \times (k+1)$).

Las instrucciones serán

```
N      = ROWS(X);                 /* tamaño muestra */
K1     = COLS(X);                 /* número de regresores
XXI    = INV(X'X);                   (k+1) */
BETA   = XXI*(X'Y);               /* estimador mco */
RESI   = Y - X*BETA;              /* residuos mco */
SUMAC  = RESI'RESI;               /* suma cuadrática de residuos */
VARI   = SUMAC/ (N-K1);           /* varianza residual */
COVV   = VARI*XXI;                /* matriz de covarianzas de
STDE   = SQRT(DIAG(COVV));        beta */
SS     = DIAGRV (EYE(K1)), STDE);
RR     = INV(SS)*COVV*INV(SS);
RDOS   = 1 - (SUMAC/ (Y'Y - N*MEANC(Y)^2));
         /* R cuadrado */
```

La inversa de la matriz X'X se almacena en XXI. A continuación se computa el vector de parámetros β^*, que se almacena en BETA. Se computan los residuos, su suma de cuadrados SUMAC y la matriz de covarianzas. En el cálculo de la matriz **SS** se emplea la sentencia DIAGRV que permite incluir el vector SQRT(DIAG(COVV))) en la diagonal principal de la matriz identidad creada por EYE(K1), con las dimensiones adecuadas. Por fin, se calcula R^2.

6.3. Inferencia

Si a las hipótesis a), b) y c) del apartado anterior le añadimos la siguiente:

d) Los residuos se distribuyen normalmente, es decir

$$e \sim N(0, \sigma^2 I)$$

Puede efectuarse inferencia sobre los coeficientes del modelo, ya que los estimadores minimocuadráticos se distribuyen

$$\beta^* \sim N(\beta, \sigma^2(X'X)^{-1})$$

Sin embargo, al ser σ^2 desconocida, en general se sustituye por su estimador insesgado s^2. En esas condiciones, puede demostrarse que el cociente

$$\frac{(\beta^* - \beta)'X'X(\beta^* - \beta)/(k)}{(y - X\beta^*)'(y - X\beta^*)/(n-k-1)}$$

sigue una distribución $F_{k,n-k-1}$. Este resultado puede emplearse para construir intervalos de confianza y contrastes generales para cualquier subconjunto de parámetros. En particular, si se desea contrastar la hipótesis de que todos los coeficientes de regresión son nulos (i.e. ninguna de los regresores tiene relación lineal con el regresando, excepto el término constante), el estadístico anterior se reduce a

$$F = \frac{R^2}{1 - R^2}\,\frac{n-k-1}{k}$$

que bajo la hipótesis nula sigue una distribución F con k, $n-k-1$ grados de libertad.

Asimismo, si se desea contrastar un subconjunto cualquiera de coeficientes **r**, el contraste adecuado es

$$F = \frac{(\beta^*_r - \beta_r)'[(X'X)^{-1}_{(r)}]^{-1}\,(\beta^*_r - \beta_r)}{rs^2}$$

que seguirá, bajo la hipótesis nula una F con r y $n-k-1$ grados de libertad. La matriz $(X'X)^{-1}_{(r)}$ es la parte de la matriz $(X'X)^{-1}$ asociada al subconjunto de coeficientes.

Adicionalmente puede tener interés contrastar restricciones lineales generales sobre los coeficientes. Supondremos que existen $h<k$ restricciones linealmente independientes que formularemos como $R\beta=r$ donde **R** es una matriz $h\times k$ de rango h y **r** es un vector $h\times 1$. Bajo la hipótesis nula de que las restricciones son ciertas, el estadístico

$$F = \frac{(R\beta^* - r)'[R(X'X)^{-1}R']^{-1}\,(R\beta^* - r)/h}{(y - X\beta^*)'(y - X\beta^*)/(n-k-1)}$$

sigue una distribución F con h y $n-k-1$ grados de libertad.

Si se emplea el estimador de la varianza de los errores s^2 definido

anteriormente, para el contraste individual de coeficientes puede emplearse el cociente

$$t_i = \frac{(\beta^*_i - \beta_i)}{l_{ii}\,s}$$

que bajo la hipótesis nula, sigue una distribución Student-t con $n-k-1$ grados de libertad, donde l_{ii} es la raíz cuadrada del elemento diagonal correspondiente en la matriz de varianzas covarianzas de los parámetros, $V(\beta^*)$.

Veamos ahora los comandos de GAUSS necesarios para efectuar los contrastes. Calcularemos los test individuales para cada coeficiente y el contraste global de regresión.

```
TSTU  = BETA ./ STDE;
PVAL  = CDFTC(ABS(TSTU), N−K1);
FSNE  = (RDOS*(N−K1)) / ((1−RDOS)*(K1+1));
FVAL  = CDFFC(FSNE,K1+1, N−K1);
```

En primer lugar se calculan los t-ratios para todos los coeficientes, y a continuación la función CDFTC nos da los niveles críticos de los contrastes (recuérdese que el nivel crítico p es la probabilidad de obtener una discrepancia mayor o igual que la observada, si la hipótesis nula $\beta_i=0$ es cierta). De igual modo se calcula el contraste general de regresión usando la forma simplificada y se calcula su nivel crítico.

Supongamos que se desea contrastar que un subconjunto de coeficientes cumplen determinadas restricciones. Como ejemplo veamos el caso en el cual se desea contrastar si

$$\beta_2 = 0 \qquad \beta_5 = -1$$

Las instrucciones serían

```
R      = 2;
BETAR  = BETA[2, 1] | BETA[5, 1];
REST   = {0, −1};
XXIR   = XXI[2, 2]~XXI[2, 5] | XXI[5, 2]~XXI[5, 5];
FR     = ((BETAR-REST)'*INV(XXIR)*(BETAR-REST))
         /(R*VARI);
FRVAL  = CDFFC(FR,R,N-K1);
```

En primer lugar se fija el número de restricciones; luego se crea el vector que contiene los parámetros de interés BETAR. Se crea el vector REST que incluye las restricciones. Se seleccionan los términos de la matriz $(X'X)^{-1}$ relacionados con los parámetros y se almacenan en la matriz 2×2 XXIR. Finalmente se calcula el contraste de acuerdo con la fórmula mencionada y se computa el nivel crítico del mismo.

Ahora veamos el caso de restricciones lineales generales. Como ejemplo, supongamos que tenemos 6 regresores y se desean contrastar las hipótesis

$$\beta_3 + \beta_5 = 8 \qquad \beta_6 = 6$$

Las instrucciones serían

```
H       = 2;
r       = {8, 6};
R       = {0 0 1 0 1 0, 0 0 0 0 0 1};
FSR     = ((R*BETA-r)'*INV(R*INV(X'X)*R')*(R*BETA-r))
          /(H*VARI);
FSRVAL  = CDFFC(FSR,H,N-K1);
```

En primer lugar se construye la matriz **R** y el vector **r** dados el número y la forma de las restricciones. Luego se calcula el contraste y se computa su nivel crítico.

Finalmente veamos un caso de interés particular, el de cambio en el término independiente de la regresión de un período a otro, mediante variables ficticias. El ejemplo será un modelo del tipo

$$Y_i = \beta_0 + \beta_1 X_i + e_i \qquad i = 1, ..., t-1$$
$$Y_i = \beta_0' + \beta_1 X_i + e_i \qquad i = t, ..., n$$

donde β_0 es distinto de β_0'. Para formular el contraste, usaremos una variable ficticia D_i que toma valores cero hasta $i=t-1$ inclusive y valores 1 desde $i=t$ en adelante. Estimaremos dos modelos, uno con la variable ficticia y otro sin ella. Mediante un contraste tipo F seleccionaremos el modelo que mejor se adapte a los datos. Nótese que, en este caso, el contraste podría hacerse estimando una única ecuación para toda la muestra con la variable ficticia y observar, mediante un contraste T Student si el coeficiente estimado de la variable ficticia es diferente de cero.

Estimaremos dos modelos

$$Y_i = \beta_0 + \beta_1 X_i + e_i \qquad i = 1, ..., n$$
$$Y_i = \alpha_0 + \alpha_1 X_i + \alpha_2 D_i + e_i \qquad i = 1, ..., n$$

Si llamamos S_1 a la suma de cuadrados de residuos del primer modelo y S_2 a la suma de cuadrados de residuos del segundo, el cociente

$$F = \frac{(S_1 - S_2)/1}{S_2/(n-3)}$$

se distribuirá F con 1 y $n-3$ grados de libertad, si la hipótesis nula $\alpha_2 = 0$ es cierta.

Las sentencias de GAUSS serán

```
LET L1[1,T-1]=0;    /* primera parte de la var ficticia */
LET L2[T,N]=1;      /* segunda parte de la var ficticia */
DUMMY=L1|L2;        /* variable ficticia completa       */

/* primera regresión */

BETA1 = INV(X'X)*X'Y;
S1    = (Y - X*BETA1)'(Y - X*BETA1);

/* segunda regresión */

XX    = X~DUMMY;
BETA2 = INV(XX'XX)*XX'Y;

/* contraste F */

S2    = (Y - XX*BETA2)'(Y - XX*BETA2);
FSNE  = ((S1 - S2)*(N - 3))/S2;
FRVAL = CDFFC(FSNE,1,3);
```

La generalización al caso de más variables así como cambios en la pendiente de la ecuación, es inmediata.

6.4. Análisis de los residuos

Una vez efectuada la regresión, resulta necesario comprobar si los residuos estimados e^* satisfacen las hipótesis de partida, en particular las de varianza constante (homocedasticidad) y ausencia de autocorrelación.

Hay una gran cantidad de contrastes en la literatura, como puede comprobarse consultando cualquier texto de Econometría. Aquí sólo vamos a mencionar el contraste de White (1980) para varianza no constante y los de Durbin-Watson y Breusch (1978) y Godfrey (1978) para autocorrelación serial.

El *contraste de White* usa los residuos minimocuadráticos ordinarios al cuadrado y efectúa una regresión auxiliar de estos sobre una constante, los regresores del modelo de base, sus cuadrados y todos los productos cruzados. Puede demostrarse que bajo la hipótesis nula de varianza constante, el producto $\mathbf{nR^2}$, donde $\mathbf{n}$ es el tamaño de la muestra y $\mathbf{R^2}$ es el coeficiente de determinación de la regresión auxiliar, sigue asintóticamente una distribución chi-cuadrado con grados de libertad igual al número de regresores de la regresión auxiliar.

Es decir, en primer lugar se calcula la regresión ordinaria

$$\mathbf{y = X\beta + e}$$

y con los residuos $\mathbf{e2=e^* \times e^*}$ se efectúa la regresión auxiliar

$$\mathbf{e2 = X\alpha + Z\delta + u}$$

donde $\mathbf{Z}$ contiene todos los cuadrados y los productos cruzados de los regresores originales.

Supongamos que la regresión tenía tres regresores, x1 x2 x3, y término constante que ocupaba la *última* columna (cuarta) de la matriz de datos X y que se dispone de los residuos almacenados en la variable RESI, el cálculo del test será:

```
E2    = RESI^2;
X2    = X[.,1:3]^2;
X21   = X[.,1] .* X[.,2];
X31   = X[.,1] .* X[.,3];
X23   = X[.,2] .* X[.,3];
XX    = X~X2~X21~X31~X23;
COEF  = INV(XX'XX)*XX'E2;
SUMR  = (E2 - XX*COEF)'(E2 - XX*COEF);
WHIT  = N*(1 - (SUMR/(E2'E2 - N*MEANC(E2)^2)));
PVAL  = CDFCHIC(WHIT,COLS(XX));
```

Nótese que se elevan al cuadrado (matriz X2) sólo las tres primeras columnas de X, ya que al crear XX se añade la matriz completa de regresores (con término constante).

El *contraste de Durbin-Watson* para detectar autocorrelación de primer orden en los residuos se define como:

$$\mathbf{DW} = \frac{\Sigma(\mathbf{e^*}_t - \mathbf{e^*}_{t-1})^2}{\Sigma \mathbf{e^{*2}}_t}$$

donde $\mathbf{e^*}_t$ es el residuo en el período t. Es fácil demostrar, que en muestras de tamaño grande, tenemos que

$$\mathbf{DW \approx 2(1 - \alpha)}$$

donde $\boldsymbol{\alpha}$ es el coeficiente de correlación entre e^*_t y e^*_{t-1}. Por tanto, si DW está cerca de 0 o de 4, es una señal de alta correlación de primer orden en los residuos. Existen tablas para el estadístico DW, que permiten contrastar si $\alpha=0$ o no.

El cálculo de éste estadístico puede hacerse con una sola sentencia en GAUSS, suponiendo que tenemos los residuos almacenados en la variable RESI.

```
DWSTAT = SUMC((RESI[2:N,1]-RESI[1:N-1,1])^2)/SUMAC;
```

otra forma de calcular este estadístico es

```
DWSTAT = SUMC[(TRIMR(RESI, 1,0)-TRIMR(RESI, 0,1)]^2/
SUMAC;
```

(la sentencia TRIM(x, t, b) quita filas arriba (TOP) o abajo (BOTTOM) de una matriz).

Recuérdese que SUMAC era la suma de los cuadrados de los residuos de la regresión original. El nivel crítico puede calcularse a partir de las tablas disponibles en cualquier libro de Econometría.

El *contraste de Breusch-Godfrey* es un método que permite considerar autocorrelación de cualquier orden entre los residuos del modelo. De nuevo se parte de los residuos minimocuadráticos y se utiliza una regresión auxiliar de éstos sobre los regresores originales del modelo y además $\mathbf{p}$ retardos de $\mathbf{e^*}_t$. Se compara el estadístico $\mathbf{nR^2}$ de la regresión auxiliar con una chi-cuadrado con $\mathbf{p}$ grados de libertad, que es su distribución asintótica bajo la nula de ausencia de autocorrelación.

Supongamos un caso sencillo de $\mathbf{p}$=3. De nuevo los residuos iniciales están en RESI.

```
RESIX = RESI[4:N,1];
P     = 3;
XX    = X[4:N,.]~RESI[3:N-1,1]~RESI[2:N-2,1]~
        RES[1:N-3,1];
COEF  = INV(XX'XX)*XX'RESIX;
SUMR  = (RESIX - XX*COEF)'(RESIX - XX*COEF);
BRGO  = N*(1 - (SUMR/ (RESIX'RESIX -
        N*MEANC(RESIX)^2)));
PVAL  = CHFCHIC(BRGO,P);
```

Nótese que en la regresión se pierden las **p** primeras observaciones y por eso se seleccionan en RESIX y XX las adecuadas para tener en cuenta los tres retardos incluidos.

6.5. Generación de distribuciones derivadas de la normal

Aunque el programa GAUSS dispone de la posibilidad de generar muestras de variables N(0, 1), o U(0, 1), mediante las sentencias RNDN y RNDU, no hay una forma directa de generar variables Student T o Chi-cuadrado. En este apartado se muestra de forma sencilla cómo generar variables que sean muestras de las tres distribuciones más empleadas en Econometría: Chi-cuadrado, Student T y F Snedecor.

— *Chi-cuadrado con* **k** *grados de libertad*

Se genera 100 datos de una chi-cuadrado con 3 grados de libertad

```
T   = 100;
K   = 3;
ZZ  = RNDN(T, K);
CHI = SUMC((ZZ^2)');
```

— *Student T con* **k** *grados de libertad*

Se generan 100 datos de una Student T con 5 grados de libertad

```
T   = 100;
K   = 5;
TST = RNDN(T,1) ./ SQRT(SUMC((RNDN(T,K)^2)')/K);
```

— *F Snedecor con* **m** *y* **n** *grados de libertad*

Se generan 100 datos de una F de Snedecor con m=7 y n=9 grados de libertad

```
T    = 100;
M    = 7;
N    = 9;
FSNE = (SUMC((RNDN(T,M)^2)')/M) ./
       (SUMC((RNDN(T,N)^2)')/N);
```

Pueden generarse valores de cualquier otra distribución asociada a la Normal usando formulaciones similares a las aquí expuestas.

CASO PRACTICO 3

Se dispone de los datos de índices de consumo de alimentos per cápita (Q_t), precios (deflactados) de los alimentos (P_t) y renta disponible (defactada) (Y_t) de los consumidores desde 1927 a 1941. Se pide construir un modelo de regresión en el cual se explique el consumo en función de las otras dos variables. La especificación adoptada será (se emplean logaritmos neperianos para interpretar los coeficientes como elasticidades):

$$\ln Q_t = \beta_0 + \beta_1 \ln P_t + \beta_3 \ln Y_t + e_t$$

Los datos son

Q_t	P_t	Y_t
88.9	91.7	57.7
88.9	92.0	59.3
89.1	93.1	62.0
88.7	90.9	56.3
88.0	82.3	52.7
85.9	76.3	44.4
86.0	78.3	43.8
87.1	84.3	47.8
85.4	88.1	52.1
88.5	88.0	58.0
88.4	88.4	59.8
88.6	83.5	55.9
91.7	82.4	60.3
93.3	83.0	64.1
95.1	86.2	73.7

La fuente es Maddala (1977) p. 116, en la siguiente página se presenta el programa de GAUSS que resuelve el problema, así como el resultado de las computaciones, calculando los contrastes sobre los residuos que se han presentado en las páginas previas.

```
/*
* Se crean los vectores de datos
*/

qt={88.9, 88.9, 89.1, 88.7, 88.0, 85.9, 86.0, 87.1, 85.4, 88.5, 88.4,
   88.6, 91.7, 93.3, 95.1};

pt={91.7, 92.0, 93.1, 90.9, 82.3, 76.3, 78.3, 84.3, 88.1, 88.0, 88.4,
   83.5, 82.4, 83.0, 86.2};
```

```
yt={57.7, 59.3, 62.0, 56.3, 52.7, 44.4, 43.8, 47.8, 52.1, 58.0, 59.8,
    55.9, 60.3, 64.1, 73.7};

/*
* Creación de la matriz global de datos con término constante
* datos en logaritmos naturales
*/

X = ONES(ROWS(PT),1)~LN(PT)~LN(YT);
Y = LN(QT);

/* Coeficientes de regresión */

N      = ROWS(X);
K1     = COLS(X);
XXI    = INV(X'X');
BETA   = XXI*(X'Y);
RESI   = Y - X*BETA;
SUMAC  = RESI'RESI;
VARI   = SUMAC/(N-K1);
COVV   = VARI*XXI;
STDE   = SQRT(DIAG(COVV);
SS     = DIAGRV(EYE(K1),STDE);
RR     = INV(SS)*COVV*INV(SS);
RDOS   = 1 - (SUMAC/(Y'Y - N*MEANC(Y)^2));

/*
* Inferencia
*/

TSTU = BETA ./ STDE;
TVAL = CDFTC(ABS(TSTU),N-K1);
FSNE = (RDOS*(N-K1)/((1-RDOS)*(K1+1));
FVAL = CDFFC(FSNE,K1+1,N-K1);

/* Contraste de White */

E2     = RESI^2;
X2     = X[.,2:3]^2;
X21    = X[.,2] .* X[.,3];
XX     = X~X2~X21;
COEF   = INV(XX'XX)*XX'E2;
SUMR   = (E2 - XX*COEF)'(E2 - XX*COEF);
WHITE  = N*(1 - (SUMR/(E2'E2 - N*MEANC(E2)^2)));
PWVAL  = CDFCHIC(WHITE,COLS(XX));
```



```
/* Contraste de Durbin-Watson */

DWSTAT=SUMC((RESI[2:N,1]-RESI[1:N-1,1])^2)/SUMAC;

/* Contraste de Breusch-Godfrey */

RESIX  = RESI[4:N,1];
P      = 3;
XXB    = X[4:N,.]~RESI[3:N-1,1]~RESI[2:N-2,1]~
         RESI[1:N-3,1];
COEF   = INV(XXB'XXB)*XXB'RESIX;
SUMR   = (RESIX - XXB*COEF)'(RESIX - XXB*COEF);
BRGO   = N*(1 - (SUMR/(RESIX'RESIX -
         N*MEANC(RESIX)^2)));
PBVAL  = CDFCHIC(BRGO,P);

/*
* Sección de resultados
*/
output file = out.pr3 on;
CLS;
format 10,5;

"Número de observaciones: " n;
"Grados de libertad: " n-k1;
print;
" Betas     S.Err.     Valores T     Valor p";
beta~stde~tstu~tval;
print;
"Matriz de Correlaciones de los regresores";
rr;
print;
" R cuadrado:                  " rdos;
" Desv Típica Residual:        " sqrt(vari);
" Estadístico F                " fsne     " Valor p "   fval;
" Contraste de White           " white    " Valor p "   pwval;
" Contraste Durbin-Watson      " dwstat;
" Contraste Breusch-Godfrey    " brgo     " Valor p "   pbval;
output off;
end;
```

El resultado obtenido es el siguiente:

```
Número de observaciones:    15.00000
Grados de libertad:         12.00000
```

Betas	S.Err.	Valores T	Valor p
4.55492	0.20089	22.67386	0.00000
−0.23520	0.05338	−4.40649	0.00043
0.24324	0.02289	10.62572	0.00000

Matriz de correlaciones de los regresores

1.00000	−0.92509	0.20558
−0.92509	1.00000	−0.56168
0.20558	−0.56168	1.00000

R cuadrado:	0.90659		
Desv Típica Residual:	0.00980		
Estadístico F	8.27894	Valor p	0.00191
Contraste de White	8.91066	Valor p	0.17866
Contraste Durbin-Watson	1.71304		
Contraste Breusch-Godfrey	0.75386	Valor p	0.86047

Todos los coeficientes son altamente significativos y los estadísticos no señalan errores graves de especificación, aunque sería necesario un análisis más detallado de los residuos.

Referencias

Breusch, T. S. (1978). Testing for Autocorrelation in Dynamic Linear Models. *Australian Economic Papers*, 17, 334-355.

Godfrey, L. G. (1978). Testing Against General Autoregressive and Moving Average Models when the Regressors Include Lagged Dependent Variables. *Econométrica*, 46, 1293-1302.

Johnston, J. (1984). *Econometric Methods*. McGraw-Hill.

Judge, G., Hill, R. C., Griffiths, W. E., Lutkepohl, H. y Lee, T. (1988). *Introduction to the theory and practice of Econometrics*. Wiley.

Lindgren, B. W. (1976). *Statiscal Theory*. Mcmillan.

Maddala, G. S. (1977). *Econometrics*. McGraw-Hill.

Peña, D. (1986). *Estadística modelos y métodos 1. Fundamentos*. Alianza Universidad.

Rohatgi, V. K. (1976). *An Introduction to Probability Theory and Mathematical Statistics*. Wiley.

White, H. (1980). A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. *Econometrica*, 48, 817-838.

Apéndice
COMANDOS BASICOS GAUSS 2.0

OPERADORES

z = x;	OPERADOR DE ASIGNACION
y = s1 $+ a2;	CONCATENACION DE CARACTERES
y = x!;	FACTORIAL (3! = 3*2*1 = 6)
y = x′;	TRANSPOSICION DE MATRICES Y VECTORES
z = x * y;	MULTIPLICACION MATRICIAL
z = x *~ y;	PRODUCTO HORIZONTAL DIRECTO
z = x + y;	SUMA
z = x − y;	RESTA
y = x[.,j];	ASIGNA TODAS LAS FILAS DE LA COLUMNA j
z = x .* y;	MULTIPLICACION ELEMENTO A ELEMENTO
z = x .*. y;	PRODUCTO DE KRONECKER
z = x ./ y;	DIVISION ELEMENTO A ELEMENTO
z = x ^ y;	EXPONENCIACION ELEMENTO A ELEMENTO
z = x / y;	DIVISION O SOLUCION DE Ax=b x=b/A
y = x[i:k,j];	ASIGNA FILAS i a k DE LA COLUMNA j
z = x | y;	CONCATENACION VERTICAL
z = x ~ y;	CONCATENACION HORIZONTAL

EJEMPLOS

```
s1 = "fichero";
s2 = ".ext";
y  = s1 $+ s2 ;
print y ;    ————→ fichero.ext

let x[2, 2] = 1 2 3 4;
let y[2, 1] = 7 8;            7  14
z = x *~ y;   ————→          24  32

x = {1 2, 3 4};
y = {5, 6};                  1  2
z = x|y';     ————→          3  4
                             5  6

z = x~y;      ————→          1  2  5
                             3  4  6
```

OPERADORES DE COMPARACION LOGICA

z = x == y;	IGUAL A
z = x /= y;	NO IGUAL
z = x < y;	MENOR QUE
z = x <= y;	MENOR O IGUAL QUE
z = x > y;	MAYOR QUE
z = x >= y;	MAYOR O IGUAL QUE
z = x EQ y;	IGUAL A
z = x NE y;	NO IGUAL
z = x LE y;	MENOR O IGUAL
z = x LT y;	MENOR QUE
z = x GE y;	MAYOR O IGUAL
z = x GT y;	MAYOR QUE

EJEMPLOS

```
x = 5;
y = 6;
z = x == y;
print z;   ————→ 0.00000

w = x /= y;
print w;   ————→ 1.00000
```


Nota: Es totalmente equivalente el usar los símbolos aritméticos para comparaciones (+ > < /) que los símbolos alfabéticos (lt gt).

COMPARACIONES ELEMENTO A ELEMENTO

z = x .== y;	IGUAL A
z = x ./= y;	NO IGUAL
z = x .< y;	MENOR QUE
z = x .<= y;	MENOR O IGUAL QUE
z = x .> y;	MAYOR QUE
z = x .>= y;	MAYOR O IGUAL QUE
z = x .EQ y;	IGUAL QUE
z = x .NE y;	NO IGUAL
z = x .LE y;	MENOR O IGUAL QUE
z = x .LT y;	MENOR QUE
z = x .GE y;	MAYOR O IGUAL
z = x .GT y;	MAYOR QUE

EJEMPLOS

```
x = {1 2 3};
y = {4 1 6};
z = x .GT y;
print z;   ————→ 0.0 1.0 0.0

let x[2, 2] = 1 2 3 4 ;
let y[2, 2] = 5 6 1 2;
z = x .<= y;
print z;   ————→ 1.0  1.0
                 0.0  0.0
```

OPERADORES DE COMPARACION
DE CARACTERES ALFABETICOS

y=s1 $==s2;	FRASE s1 IGUAL A s2 (mismo contenido)
y=s1 $/= s2;	FRASE s1 NO IGUAL A s2
y=s1 $< s2;	FRASE s1 MAS CORTA QUE s2
y=s1 $<=s2;	FRASE s1 MAS CORTA O IGUAL QUE s2
y=s1 $> s2;	FRASE s1 MAS LARGA QUE s2
y=s1 $>=s2;	FRASE s1 MAS LARGA O IGUAL QUE s2

OPERACIONES ELEMENTO A ELEMENTO

y=s1 .$==s2;	FRASE s1 IGUAL A s2
y=s1 .$/= s2;	FRASE s1 NO IGUAL A s2
y=s1 .$< s2;	FRASE s1 MAS CORTA QUE s2
y=s1 .$<=s2;	FRASE s1 MAS CORTA O IGUAL QUE s2
y=s1 .$> s2;	FRASE s1 MAS LARGA QUE s2
y=s1 .$>=s2;	FRASE s1 MAS LARGA O IGUAL QUE s2

EJEMPLOS

```
s1 = "luisito";
s2 = "luis";
y  = s1 $== s2;
print y; ——————→ 0.0

y = s1 $>= s2;
print y; ——————→ 1.0  (luisito mayor o igual que luis)

s1 = {"pepe" "juan"};
let s2[1, 2] = "luisito" "juanito";
z = s1 .$<= s2;
print z; ——————→ 0.0  1.0
print $s1;
        pepe        juan
```

Nota: Para imprimir matrices de caracteres es necesario indicarlo en el comando print con el signo $ previo al nombre de la variable.

OPERADORES LOGICOS

z = x AND y;	INTERSECCION
z = x EQV y;	EQUIVALENCIA
z = x NOT y;	COMPLEMENTO
z = x OR y;	DISJUNCION
z = x XOR y;	ELECCION EXCLUSIVA

OPERACIONES ELEMENTO A ELEMENTO

z = x .AND y;	INTERSECCION
z = x .EQV y;	EQUIVALENCIA
z = x .NOT y;	COMPLEMENTO
z = x .OR y;	DISJUNCION
z = x .XOR y;	ELECCION EXCLUSIVA

Nota: Los operadores lógicos realizan operaciones Booleanas sobre valores numéricos. Un valor no cero se considera VERDADERO, es decir, se asigna el uno (1) y un valor cero como FALSO o sea 0.

Para tablas de comparación de resultados véase el manual original de GAUSS 2.0, el apartado 4.4.

COMANDOS Y FUNCIONES GENERALES DE GAUSS

(ALGUNO DE LOS COMANDOS PRECEDIDOS
POR * PUEDEN NO ESTAR DISPONIBLES
EN LA STUDENT VERSION)

y = ABS(x);	VALOR ABSOLUTO
y = ARCSIN(x);	ARCO SENO
y = ARCTAN(x);	ARCO TANGENTE
z = ARCTAN2(y,x);	ARCO TANGENTE DE y/x
*{M,P} = BASE10(x);	DESCOMPONE UN NUMERO EN SU BASE Y POTENCIA DE 10. EJ: X=20 M=2 P=1 (M*10^P=X)
*y = BESSELJ(n,x);	FUNCION DE BESSEL TIPO UNO
*y = BESSELY(n,x);	FUNCION DE BESSEL TIPO DOS
BREAK;	TERMINA UN BUCLE DIRECTAMENTE
CALL func(arg_list);	LLAMADA A FUNCION CUANDO NO SE NECESITA UN VALOR DE RETORNO
CALLEXE func(Args);	LLAMADA A FUNCION NO ESCRITA EN GAUSS (P. EJ. EN FORTRAN)
y = CDFBETA(x,a,b);	FUNCION DE DISTRIBUCION ACUMULADA (FDA) DE LA FUNCION BETA
*c = CDFBVN(h,k,r);	FDA DE LA NORMAL BIVARIANTE ESTANDAR
y = CDFCHIC(x,n);	COMPLEMENTO DE LA FDA DE UNA CHI-CUADRADO
y = CDFFC(x,n1,n2);	COMPLEMENTO DE LA FDA DE UNA F
*y=CDFFNC(x,v1,v2,d);	INTEGRAL BAJO UNA F NO CENTRAL
*g=CDFGAM(x,intlim);	FUNCION GAMMA INCOMPLETA
y = CDFN(x);	FDA DE UNA NORMAL(0, 1)

y = CDFNC(x);	COMPLENTO DE LA FDA N(0, 1)
y = CDFTC(x,n);	COMPLEMENTO DE LA T DE STUDENT
*y = CDFTNC(x,v,d);	INTEGRAL BAJO UNA T NO CENTRAL
*y = CDFTVN(x1,x2,x3,rho12,rho23,rho31);	FDA NORMAL TRIVARIANTE
y = CDIR(s);	OBTIENE EL DIRECTORIO DE TRABAJO
y = CEIL(x);	REDONDEA NUMEROS REALES A ENTEROS
*{yr,yi} = CFFT(xr,xi);	COMPUTA LA TRANSFORMADA RAPIDA DE FOURIER (FFT) PARA NUMEROS COMPLEJOS
*{yr,yi} = CFFTI(xr,xi);	INVERSA FFT COMPLEJA
y = CHOL(x);	DESCOMPOSICION DE CHOLESKY
y = CHRS(x);	CONVIERTE UN VECTOR DE CODIGOS ASCII EN CARACTERES
CLEAR x, y;	IGUALA A CERO MATRICES
CLEARG a,b,c;	IGUALA A CERO MATRICES GLOBALES
y = CLSE (handle);	CIERRA UN FICHERO GAUSS
CLOSEALL f1, f2;	CIERRA TODOS LOS FICHEROS GAUSS ABIERTOS
CLS;	LIMPIA LA PANTALLA
y = CINT(x);	IDENTICO QUE ROUND
*{zr,zi} = CMADD(xr,xi,yr,yi);	SUMA ELEMENTO A ELEMENTO DE DOS MATRICES COMPLEJAS
*{zr,zi}=CMDIV(xr,xi,yr,yi);	IDEM DIVISION
*{zr,zi} = CMEMULT(xr,xi,yr,yi);	IDEM MULTIPLICACION
*{zr,zi} = CMINV(xr,xi);	INVERSA MATRIZ COMP.
*{zr,zi} = CMMULT(xr,xi,yr,yi);	MULTIPLICACION DE MATRICES COMPLEJAS
*{xr,xi} = CMSOLN(br,bi,AR,AI);	SOLUCION DE UNA ECUACION LINEAL COMPLEJA
*{zr,zi} = CMSUB(xr,xi,yr,yi);	RESTA EQUIVALENTE A CMADD

*{zr,zi} = CMTRANS(xr,xi);	TRASPUESTA DE LA CONJUGADA COMPLEJA
y = CODE(e,v);	RECODIFICA DATOS
y = COLOR(cv);	FIJA VALORES PARA COLOR
y = COLS(x);	NUMERO DE COLUMNAS POR MATRIZ
y = COLSF(fh);	NUMERO DE COLUMNAS EN FICHERO
*{zr,zi} = COMPLEX(x);	CONVIERTE UNA MATRIZ REAL EN DOS COMPLEJAS
*{z1r,z1i,z2r,z2i} = COMPLEX2(x1,x2);	CONVIERTE DOS MATRICES REALES EN DOS PARES DE COMPLEJAS
x = CON(n,k);	PIDE DATOS NUMERICOS DEL TECLADO
*c = COND(x);	NUMERO DE CONDICIONAMIENTO
c = CONS;	PIDE DATOS ALFABETICOS AL TECLADO
CONTINUE:	VUELVE AL PRINCIPIO DE UN DO
c = CONV(b,x,first,last);	CONVOLUCION DE DOS VECTORES
y = CORELEFT;	WORKSPACE DISPONIBLE
*cx = CORRM(m);	MATRIZ DE CORRELACION BASADA EN LA MATRIZ DE MOMENTOS
*cx = CORRVC(vc);	MATRIZ DE CORRELACION BASADA EN LA MATRIZ DE COVARIANZAS
*cx = CORRX(x);	MATRIZ DE CORRELACION BASADA EN LA MATRIZ DE DATOS ORIGINALES
y = COS(x);	COSENO
*y = COSH(x);	COSENO HIPERBOLICO
c = COUNTS(x,v);	FRECUENCIAS DE LOS ELEMENTOS DE UN VECTOR
*C = COUNTWTS(x,v,w);	FRECUENCIAS PONDERADAS DE LOS ELEMENTOS DE UN VECTOR
CREATE fh = file WITH vnames,col,typ;	CREA FICHERO DE DATOS
*Z = CROSSPRD(x,y);	MATRIZ DE PRODUCTOS CRUZADOS
y = CROUT(x);	DESCOMPOSICION LU DE CROUT

y = CROUTP(x);	IDEM CROUT CON FILA PIVOTE
y = CSRCOL;	POSICION (COLUMNA) DEL CURSOR
y = CSRLIN;	POSICION (FILA) DEL CURSOR
*y = CUMPRODC(x);	PRODUCTOS ACUMULADOS DE LAS COLUMNAS DE UNA MATRIZ
*y = CUMSUMC(x);	SUMAS ACUMULADAS DE LAS COLUMNAS DE UNA MATRIZ
y = DATE;	FECHA (4×1 vector)
*str = DATESTR(d);	FECHA (CARECTERES)
*daynum = DAYINYR(dt);	NUMERO DE DIA EN EL AÑO
*{yr,yi} = DCFFT(xr,xi);	FFT COMPLEJA DISCRETA
*{yr,yi} = DCFFTI(xr,xi);	FFT INVERSA COMPLEJA DISCRETA
DECLARE x = {1 2, 3 4};	INICIALIZA MATRICES (COMPILAR)
DELETE /msfp s1,s2,s3;	BORRA DE LA TABLA DE SIMBOLOS
*y = DELIF(x,e);	BORRA FILAS DE UNA MATRIZ
*y = DESIGN(x);	CREA MATRIZ BINARIA (0, 1)
y = DET(x);	COMPUTA EL DETERMINANTE DE X
y = DETL;	DETERMINANTE DE LA ULTIMA DESCOMPOSICION DE MATRICES
y = DFREE(drive);	ESPACIO DISPONIBLE EN DISCO
y = DIAG(x);	DIAGONAL PRINCIPAL DE X
y = DIAGRV(x,v);	INSERTA EL VECTOR v EN LA DIAGONAL PRINCIPAL DE X
DISABLE;	DESACTIVA EL OPERADOR NDP
DO UNTIL expression;	BUCLE (ACABA EN ENDO)
DO WHILE expression;	BUCLE (ACABA EN ENDO)
DOS cmd;	EJECUTA UN COMANDO D.O.S.
DOS ^str;	EJECUTA UN COMANDO D.O.S. DEFINIDO EN str
*{vnam,mean,var,std,min,max,valid,mis} = DSTAT(dataset,vars);	ESTADISTICAS DESCRIPTIVAS
*y = DUMMY(x,v);	VARIABLES FICTICIAS
*y = DUMMYBR(x,v);	VA. FICTICIAS LIMITADAS DERECHA
*y = DUMMYDN(x,v,p);	VA. FICTICIAS ELIMINA ULTIMA
ED filename;	ACCEDE A EDITOR EXTERNO

EDIT filename;	EDITA FICHERO CON EDITOR GAUSS
y = EDITM(x);	EDITA UNA MATRIZ
*{var,vai} = EIGCG(xr,xi);	AUTOVALORES MATRIZ COMPLEJA
*{var,vai,ver,vei} = EIGCG2(xr,xi);	AUTOVALORES Y AUTOVECTORES DE UNA MATRIZ COMPLEJA
*va = EIGCH(xr,xi);	AUTOVALORES MATRIZ HERMITIANA COMPLEJA
*{var,vai,ver,vei} = EIGCH2(xr,xi);	AUTOVALORES Y AUTOVECTORES DE UNA MATRIZ HERMITIANA
*{var,vai} = EIGRG(x);	AUTOVALORES MATRIZ REAL
*{var,vai,ver,vei} = EIGRG2(x);	AUTOVALORES/VECTORES MATRIZ REAL
*va = EIGRS(x);	AUTOVALORES MATRIZ REAL SIMETRICA
*{va,ve} = EIGRS2(x);	AUTOVALORES/VECTORES REAL SIMETRICA
ELSE;	PARTE DE LA CLAUSURA IF
ELSEIF expression;	PARTE DE LA CLAUSULA IF
ENABLE;	ACTIVA OPERADOR NDP
END;	PARA UN PROGRAMA Y CIERRA FICHERO
ENDIF;	FIN DE CLAUSULA IF
ENDO;	FIN DE BUCLE DO
ENDP;	FIN DE DEFINICION DE PROC.
y = ENVGET(s);	OBTIENE UNA CARACTERISTICA DEL ENTORNO D.O.S. DE TRABAJO
y = EOF(fh);	COMPRUEBA FIN DE FICHERO
y = ERF(x);	FUNCION DE ERROR GAUSIANA
y = ERFC(x);	COMPLEMENTO DE ERF
y = ERROR(x);	CODIGO DE ERROR DEFINIDO POR EL USUARIO
ERRORLOG str;	MENSAJE AL FICHERO DE ERRORES
*days = ETDAYS(ds,de);	TIEMPO ENTRE FECHAS
*hs = ETHSEC(ds,de);	ETDAYS EN CIENTOS DE SEGUNDOS

*str = ETSTR(hs);	PONE ETHSEC EN CARACTERES
*n = EXCTSMPL(in,out,percent);	TOMA UNA MUESTRA ALEATORIA
y = EXEC(prgm,comline);	EJECUTA UN PROGRAMA EXE
y = EXP(x);	FUNCION EXPONENCIAL
EXTERNAL matrix x;	DECLARA X COMO EXTERNA
y = EYE(n);	CREA MATRIZ IDENTIDAD n×n
*{yr,yi} = FFT(x);	FFT REAL
*y = FFTI(xr,xi);	INVERSA FFT REAL
y = FILES(s,a);	OBTIENE NOMBRES DE FICHEROS
y = FIX(x);	TRUNCA UN NUMERO REAL (TRUNC)
y = FLOOR(x);	REDONDEA UN NUMERO REAL
FN cat(x,y) = x*inv(y);	FUNCION DEFINIDA POR EL USUARIO
FORMAT {/mf} {/jnt} {f,p};	FORMATO DE IMPRESION DE MATRICES
y = FTOCV(x,field,prec);	CONVIERTE UNA MATRIZ DE NUMEROS EN OTRA DE CARACTERES
y = FTOS(x,fmat,field,prec);	FTOVC PARA ESCALARES
y = GAMMA(x);	FUNCION GAMMA
y = GETF(file, mode);	CARGA UN FICHERO ASCII O BINARIO
y = GETNAME(dataset);	CARGA NOMBRES DE VARIABLES DE UN FICHERO
GOSUB label(x,y);	SALTA A UNA SUBRUTINA
GOTO label(x,y);	SALTA A UNA ETIQUETA
*g = GRADP(&f,x0);	GRADIENTE DE f(x)
GRAPH x,y;	GRAFICO DE DOS VECTORES
*h = HESSP(&f,x0);	HESSIANO DE f(x)
y = HSEC;	CIENTOS DE SEGUNDOS DESDE MEDIANOCHE
IF expression;	ELECCION CONDICIONAL
*zi = IMAG(xr,xi);	PARTE IMAGINARIA DE UN PAR DE MATRICES COMPLEJAS L
#INCLUDE filename;	COMPILA OTRO FICHERO EN ESE PUNTO

z = INDCV(what,where);	COTEJA UN VECTOR DE CARACTERES CON OTRO
*y = INDEXCAT(x,v);	INDICE DE ELEMENTOS POR CATEGORIAS
*{name, indx} = INDICES(dataset,vars);	NOMBRES DE VARIABLES DEL PROCESO
*{name1,indx1,name2,indx2} = INDICES2(dataset,var1,var2);	VARIABLES DEPENDIENTES E INDEPENDIENTES DEL PROCESO
*y = INTQUAD1(&F,xl);	INTEGRAL DE f(x)
*y = INTQUAD2(&f,xl,yl);	INTEGRAL DE f(x,y)
*y = INTQUAD3(&f,xl,yl,zl);	INTEGRAL DE f(x,y,z)
y = INT(x);	REDONDEA (VEASE FLOOR)
*INTRLEAV(infile1,infile2,outfile,keyvar,keytyp);	COMBINA DOS VECTORES ORDENADOS
*y = INTRSECT(v1,v2,flag);	INTERSECCION DE 2 VECTORES
*y = INTSIMP(&f,xl,tol);	INTEGRA CON EL METODO SIMPSON
y = INV(x);	INVERSA DE UNA MATRIZ
y = INVPD(x);	INVERSA UNA MATRIZ SIMETRICA POSITIVA DEFINIDA
y = ISMISS(x);	CONTROLA VALORES AUSENTES
y = KEY;	CODIGO ASCII DE UNA TECLA
LET x = {1 2, 3 4};	CREA UNA MATRIZ (2×2)
LIBRARY lib1,lib2;	ACTIVA LIBRERIAS ESPECIALES
LINE x,y;	DIBUJA UNA LINEA
#LINESOFF;	NO NUMERA LINEAS DE PROGRAMA
#LINESON;	NUMERO LINEAS DE PROGRAMA
y = LN(x);	LOGARITMO NEPERIANO (NATURAL)
*y = LNFACT(x);	LN DE LA FUNCION FACTORIAL
LOAD x,y[ ]=file;	CARGA UNA MATRIZ O UN FICHERO ASCII DE DATOS
*y = LOADD(dataset);	CARGA UN FICHERO DE DATOS
LOADEXE buf = file;	CARGA UNA FUNCION ESCRITA EN OTRO LENGUAJE (v.gr. FORTRAN)

LOADF f;	CARGA UNA FUNCION DEL USUARIO
LOADM x;	CARGA UNA MATRIZ
LOADP p;	CARGA UN PROC. DEL USUARIO
LOADS s;	CARGA UNA RISTRA DE CARACTERES
LOCAL x,y;	DECLARA COMO LOCALES A x,y
LOCATE row,col;	POSICION DEL CURSOR
y = LOG(x);	LOGARITMO DECIMAL (BASE 10)
y = LOWER(s);	CAMBIA A MINUSCULAS UNA RISTRA
*L = LOWMAT(x);	MATRIZ TRIANGULAR INFERIOR
*L = LOWMAT1(x);	MATRIZ TRIANGULAR INFERIOR CON DIAGONAL PRINCIPAL DE UNOS
y = LPOS;	POSICION DE INICIO DE IMPRESION
LPRINT {/mf /jnt} expression;	IMPRIMIR EN LA IMPRESORA
LPWIDTH m;	FIJA ANCHO DE CARRO DE IMPRESORA
LSHOW x;	IMPRIME LISTA DE SIMBOLOS ACTIVOS
*MAKEVARS(x,vnames,xnames);	CREA VECTORES DESDE UNA MATRIZ
y = MAXC(x);	MAXIMO DE CADA COLUMNA
y = MAXINDC(x);	INDICE DE LOS MAXIMOS DE CADA COLUMNA
y = MEANC(x);	MEDIA DE CADA COLUMNA
*MERGEBY(infile1,infile2,outfile,keytyp);	FUSION CON CLAVES
*x = MERGEVAR(vnames);	CREA MATRIZ DESDE VECTORES
y = MINC(x);	MINIMO DE CADA COLUMNA
y = MININDC(x);	INDICE DE LOS MINIMOS DE CADA COLUMNA
y = MISS(x,v);	CONVIERTE VALORES NUMERICOS A CLAVE DE «MISSING» (AUSENTES)
*y = MISSEX(x,e);	IDEM ANTERIOR CON CLAVES 1/0
y = MISSRV(x,v);	INVERSO DE MISS
y = MOMENT(x,d);	MATRIZ DE PRODUCTOS CRUZADOS (X'X)
*m = MOMENTD(dataset,vars);	MOMENT DE UN GRUPO DE DATOS

MSYM str;	FIJA EL SIMBOLO PARA «MISSING DATA» (DATOS AUSENTES)
y = NDPCHK(mask);	COMPRUEBA EL STATUS DE NDP
NDPCLEX;	ACTUALIZA LOS CODIGOS NDP
y = NDPCNTRL(new,mask);	FIJA Y OBTIENE CODIGO NDP
NEW {nos {,mps}};	ELIMINA TABLA DE SIMBOLOS (TODOS)
*b = NULL(x);	COMPUTA UNA BASE ORTONORMAL PARA EL ESPACIO NULO DE UNA MATRIZ
*{vname,m,b,stb,vc,stderr,sigma,cx,rsq,resid,dwstat} = OLS(dataset,depvar,indvars);	REGRESION MCO
*b = OLSQR(y,x);	MCO CON DESCOMPOSICION QR
*{b,r,p} = OLSQR2(y,x);	MCO Y OTROS RESULTADOS CON QR
y = ONES(n,k);	MATRIZ DE UNOS
OPEN fh = file {FOR mode};	ABRE UN FICHERO GAUSS
*y = ORTH(x);	BASE ORTONORMAL PARA UNA COLUMNA DE UNA MATRIZ
OUTPUT {FILE = file} {ON/RESET/OFF};	CONTROLA UNIDAD AUXILIAR PARA SALIDA DE RESULTADOS
OUTWIDTH m;	ANCHO DEL OUTPUT AUXILIAR
y = PACKR(x);	ELIMINA FILAS CON «MISSING»
*PAUSE(sec);	ESPERAR sec SEGUNDOS
*PAUSE(sec);	ESPERAR sec SEGUNDOS
y = PDFN(x);	FUNCION DE DENSIDAD GAUSSIANA
y = PI;	3.1416...
*y = PINV(x);	INVERSA GENERALIZADA DE MOORE
PLOT x,y;	GRAFICA DE DOS VECTORES
PLOTSYM m;	FIJA SIMBOLOS USADOS POR PLOT
*c = POLYCHAR(x)	POLINOMIO CARACTERISTICO
*y = POLYEVAL(x,c);	EVALUACION DE POLINOMIO
*y = POLYINT(xa,ya,x);	INTERPOLACION POLINOMIAL
*c = POLYMAKE(r);	COEFICIENTES DEL POLINOMO DADAS LAS RAICES DEL MISMO
*y = POLYMAT(x,p);	MATRIZ DE LAS POTENCIAS DE X DESDE 1 A P

*c = POLYMULT(c1,c2);	MULTIPLICACION DE POLINIMIOS
*r = POLYROOT(c);	RAICES DEL POLINOMIO DADOS LOS COEFICIENTES
POP b; POP a;	MATRIZ DE REFERENCIA USADA POR GOSUB Y GOTO
PRCSN m;	FIJA PRECISION
PRINT {/mf /jnt} expression;	IMPRIME MATRIZ
PRINTDOS s;	IMPRIME RISTRA DE CARACTERES EN LA IMPRESORA D.O.S.
y = PRINTFM(x,mask,fmt);	IMPRIME MATRIZ FORMATEADA
PROC (nrets) = name(args);	INICIA DEFINICION DE PROC.
y = PRODC(x);	PRODUCTO DE COLUMNAS
*{q, r} = QR(x);	DESCOMPOSICION QR X = Q * R
*{q,r,e} = QR1(x);	DESCOMPOSICION QR X[.,E]=Q*R
*{q,r,e} = QR2(x,pvt);	IDEM X[.,E] = Q * R
*k = RANK(x);	RANGO DE UNA MATRIZ
*y = RANKINDX(x,flag);	INDICE DE RANGO DE UN VECTOR
y = READR(fh,r);	LECTURA DESDE UN FICHERO GAUSS
*zr = REAL(xr,xi);	OBTIENE LA PARTE REAL DE UNA MATRIZ COMPLEJA (xr,xi)
*y = RECODE(x,e,v);	RECODIFICA DATOS
y = RECSERAR(x,y0,a);	GENERA SERIES AR(P)
y = RECSERCP(x,z);	GENERA SERIES RECURSIVAS QUE CONTIENEN MULTIPLICACIONES
*y = RECSERRC(x,z);	GENERA SERIES RECURSIVAS QUE CONTIENEN DIVISIONES
y = RESHAPE(x,r,c);	REFORMULA UNA MATRIZ
RETP(x,y,z);	RETORNO DESDE UN PROCEDIMIENTO
RETURN(x,y,z);	RETORNO DESDE UNA SUBRUTINA
y = REV(x);	CAMBIA EL ORDEN DE LAS FILAS
RNDCON c;	CAMBIA EL ORIGEN DEL GENERADOR DE NUMEROS ALEATORIOS (GNA)
RNDMOD m;	CAMBIA EL MODULO DEL GNA
RNDMULT a;	CAMBIA EL MULTIPLICADOR DEL GNA

y = RNDN(n,k);	MATRIZ DE NUMEROS ALEATORIOS N(0, 1)
y = RNDNS(n,k,seed);	MATRIZ DE NUMEROS ALEATORIOS N(0, 1) INCLUYENDO SEMILLA
RNDSEED seed;	CAMBIA LA SEMILLA DEL GNA
y = RNDU(n,k);	MATRIZ DE NUMEROS ALEATORIOS U(0,1)
y = RNDUS(n,k,seed);	MATRIZ DE NUMEROS ALEATORIOS U(0, 1) INCLUYENDO SEMILLA
y = ROTATER(x,r);	MUEVE LAS FILAS DE UNA MATRIZ DE IZQUIERDA A DERECHA
y = ROUND(x);	REDONDEA AL ENTERO MAS PROXIMO
y = ROWS(x);	NUMERO DE FILAS EN UNA MATRIZ
y = ROWSF(fh);	NUMERO DE FILAS EN FICHERO GAUSS
*y = RREF(x);	COMPUTA LA FORMA REDUCIDA (EN ESCALON) DE UNA MATRIZ
RUN filename;	EJECUTA UN PROGRAMA
SAVE x,y;	SALVA MATRICES, ETC.
*y = SAVED(x,dataset,vnames);	SALVA UNA MATRIZ EN UN FICHERO
y = SCALERR(x);	CONTROLA CODIGO DE ERROR ESCALAR
y = SCALMISS(x);	CONTROLA SI UN ESCALAR CONTIENE «MISSING VALUES»
SCREEN;	CONTROLA SALIDAS A PANTALLA
SCROLL v;	CONTROLA PARTE DE LA PANTALLA
y = SEEKR(fh,r);	MUEVE EL PUNTERO A UNA FILA DETERMINADA DEL FICHERO DE DATOS
*y = SELIF(x,e);	SELECCIONA FILAS DE UNA MATRIZ MEDIANTE CODIGOS 1/0
y = SEQA(start,inc,n);	CREA PROGRESION ARITMETICA
y = SEQM(start,inc,n);	CREA PROGRESION GEOMETRICA
*y = SETDIF(v1,v2,flag);	UNICO ELEMENTO DE v1 QUE NO ESTA EN v2

*nvec = SETVARS(dataset);	OBTIENE NOMBRES EN EL GRUPO DE DATOS Y CREA MATRICES CON ELLOS
y=SETVMODE(mode);	FIJA EL MODO VIDEO (QUICKGRAPHICS)
y = SHIFTR(x,s,f);	MUEVE FILAS EN UNA MATRIZ
SHOW x;	MUESTRA TABLA DE SIMBOLOS
y = SIN(x);	SENO
*y = SINH(x);	SENO HIPERBOLICO
y = SOLPD(b,A);	RESUELVE ECUACIONES POSITIVO DEFINIDAS
y = SORTC(x,c);	ORDENA LAS FILAS DE X CON RESPECTO A UNA COLUMNA
y = SORTCC(x,c);	IDEM SORTC PARA VARIABLES ALFABETICAS
*SORTD(infile,outfile,keyvar,keytyp);	ORDENA UN FICHERO
*y = SORTHC(x,c);	COMO SORTC CON CLAVE
*y = SORTHCC(x,c);	COMO SORTCC CON CLAVE
*ind = SORTIND(x);	INDICE PARA ORDENACION NUMERICA
*ind = SORTHINDC(x);	IDEM PARA ALFABETICAS
*y = SORTMC(x,v);	ORDENA UNA MATRIZ MEDIANTE MULTIPLES COLUMNAS
y = SQRT(x);	RAIZ CUADRADA
y = STDC(x);	DESVIACION TIPICA DE COLUMNAS
y = STOF(s);	CONVIERTE CARACTER ALFABETICO A VALOR EN COMA FLOTANTE
STOP;	DETIENE EJECUCION
y = STRINDX(where,what,start);	ENCUENTRA EL INDICE DE UNA FRASE CON OTRA COMUN EN PARTE
y = STRLEN(s);	LONGITUD DE UNA FRASE
y = STRSECT(str,strt,len);	EXTRAE SUBFRASE DE UNA FRASE
y = SUBMAT(x,r,c);	SUBMATRIZ
*y = SUBSCAT(x,v,s);	SUSTITUYE VALORES POR CATEGORIAS

*y = SUBSTUTE(x,e,v);	SUSTITUYE VALORES MEDIANTE CLAVE 1/0
y = SUMC(x);	SUMA DE COLUMNAS
*s = SVD(x);	VALORES SINGULARES
*{u,s,v} = SVD1(x);	DESCOMPOSICION DE VALOR SINGULAR X = U * S * V' L
*{u,s,v} = SVD2(x);	DESCOMPOSICION DE VALOR SINGULAR (COMPACTO) X = U * S * V' L
SYSTEM x;	SALIR DE GAUSS A D.O.S.
y = TAN(x);	TANGENTE
*y = TANH(x);	TANGENTE HIPERBOLICA
y = TIME;	HORA ACTUAL EN UN 4×1 vector
*ts = TIMESTR(t);	FORMATEA LA HORA EN FORMA NUMERICA A FORMA ALFABETICA
*t = TOEPLITZ(x);	MATRIZ TOEPLITZ
TRACE new,mask;	FIJA MODO DE TRACE PARA DEPURACION (DEBUG)
TRAP new,mask;	FIJA SIGNO DE AVISO
y = TRAPCK(m);	CONTROLA SIGNO DE AVISO
y = TRIMR(x,t,b);	ELIMINA FILAS DE LAS PARTES INFERIOR Y SUPERIOR DE UNA MATRIZ
y = TRUNC(x);	TRUNCA NUMERO REAL
t = TYPE(x);	MUESTRA LA TABLA DE SIMBOLOS
y = TYPECV(s);	MUESTRA LA TABLA DE SIMBOLOS USANDO NOMBRES DE VARIABLES
y = TYPEF(fh);	TIPO DE DATOS DE UN GRUPO
*y = UNION(v1,v2,flag);	UNION DE DOS VECTORES
*index = UNIQUINDX(x,flag);	INDICES DE VALORES UNICOS
*y = UNIQUE(x,flag);	MUESTRA VALORES UNICOS
*u = UPMAT(x);	MATRIZ TRIANGULAR SUPERIOR
*u = UPMAT1(x);	MATRIZ TRIANGULAR SUPERIOR CON DIAGONAL DE UNOS
y = UPPER(s);	PONE UNA FRASE EN MAYUSCULAS
y = VALS(s);	TRADUCE LAS LETRAS DE UNA FRASE A SUS VALORES ASCII

y = VARGET(s);	OBTIENE UNA VARIABLE USANDO SU NOMBRE COMO ARGUMENTO
y = VARPUT(x,s);	ASIGNA x A LA MATRIZ MENCIONADA EN LA FRASE S
*y = VARTYPE(names);	OBTIENE EL TIPO DE DATOS; SI NUMERICOS=1 SI CARACTER=0
*vc = VCM(m);	MATRIZ DE VARIANZAS COVARIANZAS DE LA MATRIZ DE MOMENTOS
*vc = VCX(x);	IDEM DE LA MATRIZ DE DATOS
y = VEC(x);	CREA UN VECTOR SUPERPONIENDO SUCESIVAMENTE LAS COLUMNAS DE x
*WAIT;	ESPERA HASTA PULSAR TECLA
*WAITC;	COMO WAIT PERO SUPRIME TODAS LAS INSTRUCCIONES PENDIENTES
y = WRITER(fh,x);	ESCRIBE UNA MATRIZ
y = ZEROS(n,k);	MATRIZ DE CEROS