Práctica 5

Espacios vectoriales. Ortogonalización de Gram-Schmidt.

```
Clear["Global`*"]
```

(Basada en una práctica original de Antonio Palomares)

Dimensión de un subespacio vectorial. Vectores linealmente independientes.

Como vimos en la práctica 3, la orden RowReduce efectúa operaciones de filas para obtener una matriz equivalente en forma escalonada reducida. Si tenemos un conjunto de vectores, podemos considerar la matriz que tiene dichos vectores como filas. Hacer operaciones elementales con las filas de esa matriz sería lo mismo que hacer operaciones elementales con los vectores. Así pues la orden RowReduce aplicada a esa matriz nos daría una matriz cuyas filas representarían a otro conjunto de vectores tales que generan el mismo subespacio.

Por ejemplo, si tenemos los vectores v1 = {1, 2, 3}, v2 = {2, 4, 8} y v3 = {3, 6, 9}, podemos formar con ellos la siguiente matriz M1, colocando los vectores en las filas de la matriz.

```
M1 = {{1, 2, 3}, {2, 4, 8}, {3, 6, 11}}
```

Si hacemos la forma escalonada reducida de esta matriz, obtenemos la siguiente matriz.

```
M2 = RowReduce[M1];
MatrixForm[M2]
```

Como las filas de M2 se han obtenido a partir de las filas de M1 haciendo operaciones elementales, podemos decir que hay una combinación lineal de v1, v2 y v3 que da como resultado el vector {1, 2, 0}, que hay otra combinación de v1, v2 y v3 que da como resultado el vector {0, 0, 1}, independiente del anterior. Por último, también sabemos que hay una tercera combinación lineal de los vectores v1, v2 y v3 que da como resultado el vector {0,0,0}, lo cual implica que los tres vectores son linealmente dependientes.

Podemos ver que las dos primeras filas de M2 son linealmente independientes, con lo que podemos afirmar que hay dos vectores independientes que se obtienen mediante combinaciones lineales de v1, v2 y v3.

Además los vectores v1, v2 y v3 generan el mismo subespacio que los vectores {1, 2, 0} y {0, 0, 1}, por tanto los vectores v1, v2 y v3 generan un subespacio de dimensión 2.

El rango de una matriz es el número de filas linealmente independiente de una matriz, y es también igual al número de filas linealmente independientes de su forma escalonada reducida. Por eso la orden RowReduce nos permite saber la dimensión del subespacio generado por unos vectores.

Ejercicio: Determine si los vectores v1 = {1, 2, 3}, v2 = {1, 2, -3} y v3 = {2, 4, 3} son linealmente dependientes.

Ejercicio: Determine la dimensión del subespacio generado por los vectores v1 = {1, 2, -1, 2}, v2 = {1, 0, 1, -2}, v3 = {-1, -1, -1, 2} y v4 = {4, 4, 2, -4}.

Ecuaciones de un subespacio vectorial

A partir de una base de un subespacio, podemos hallar las ecuaciones paramétricas e implícitas del subespacio.

Por ejemplo, si U es el subespacio de $\mathbb{R}^4$ generado por los vectores {1, 0, 2, 0} y {1, 1, -1, 2}, un vector cualquiera de U se escribirá como {x, y, z, t} = a {1, 0, 2, 0}+ b {1, 1, -1, 2}. Si igualamos coordenada a coordenada, tendremos unas ecuaciones parámetricas de U, donde los parámetros son a y b :

x = a + b,
y = b,
z = 2a - b,
t = 2b.

Si en estas ecuaciones eliminamos los parámetros a y b tendremos unas ecuaciones implícitas. De la segunda ecuación tenemos que b = y, y de la primera ecuación tenemos que a = x - b = x - y. Sustituyendo estas expresiones de a y b en las ecuaciones tercera y cuarta, tenemos

z = 2x - 3y,
t = 2y,

que son las ecuaciones implícitas del subespacio U.

Ahora vamos a calcular las ecuaciones del subespacio U usando instrucciones de *Mathematica.*

Primero definimos el conjunto de vectores que generan U, y los parámetros, tantos como generadores, que se emplearán en las ecuaciones paramétricas.

```
generadores = {{1, 0, 2, 0}, {1, 1, -1, 2}}
```

```
parametros = {a, b}
```

Un vector general de U tendrá la forma

```
EcVectorial = {x, y, z, t} == Transpose[generadores].parametros
```

Para igualar coordenada a coordenada, desarrollamos la anterior ecuación con la orden LogicalExpand.

```
EcParametricas = LogicalExpand[EcVectorial]
```

Para obtener unas ecuaciones implícitas, eliminamos de estas ecuaciones los parámetros a y b con la orden Eliminate.

```
EcImplicitas = Eliminate[EcParametricas, parametros]
```

Supongamos ahora que en lugar de dar una base del subespacio U, damos un conjunto de generadores que no es linealmente independiente (en concreto el tercer vector es suma de los dos primeros).

```
generadores = {{1, 0, 2, 0}, {1, 1, -1, 2}, {2, 1, 1, 2}};
parametros = {a, b, c};
EcVectorial = {x, y, z, t} == Transpose[generadores].parametros
EcParametricas = LogicalExpand[EcVectorial]
EcImplicitas = Eliminate[EcParametricas, parametros]
```

Observamos que en las ecuaciones paramétricas aparecen tres parámetros aunque el subespacio es de dimensión 2 (obsérvese que si definimos otros dos parámetros a2=a-b y b2=b+c, las ecuaciones paramétricas se pueden expresar en función de a2 y b2 solamente) y que se obtienen las mismas ecuaciones implícitas que en el ejemplo anterior.

Si quisiéramos obtener unas ecuaciones paramétricas con sólo 2 parámetros, debemos obtener un sistema linealmente independiente que generen el mismo subespacio, es decir, una base del subespacio. Para hacerlo empleamos la orden RowReduce.

```
generadores = {{1, 0, 2, 0}, {1, 1, -1, 2}, {2, 1, 1, 2}};
generadoresR = RowReduce[generadores]
```

Observamos que el tercer vector de la salida de RowReduce es cero, y que bastan dos vectores para generar el mismo subespacio, sin embargo como RowReduce puede cambiar el orden de las filas, no podemos asegurar que los dos primeros vectores del sistema de generadores que hemos definido son los que generan el espacio, pero si podemos asegurar que los dos vectores independientes de la salida de RowReduce generan el subespacio.

Por tanto definimos una base con los dos primeros vectores de la salida de RowReduce, y a partir de ellos hallamos unas ecuaciones paramétricas con 2 parámetros, y las mismas ecuaciones implícitas.

```
base = {generadoresR[[1]], generadoresR[[2]]}
parametros = {a, b};
EcVectorial = {x, y, z, t} == Transpose[base].parametros
EcParametricas = LogicalExpand[EcVectorial]
EcImplicitas = Eliminate[EcParametricas, parametros]
```

Ejercicio: Halle unas ecuaciones implícitas y paramétricas del subespacio de $\mathbb{R}^4$ generado por los vectores {1, 0, 1, 0}, {0, 2, 0, 2} y

{1, 0, 0, 1}.

Base de un subespacio vectorial

Ya hemos visto en la primera sección de esta práctica cómo a partir de un conjunto de generadores de un subespacio podemos hallar, usando la orden RowReduce, una base de dicho subespacio. Ahora nos planteamos el problema de obtener una base de un subespacio definido por sus ecuaciones implícitas. En primer lugar definimos una lista con las ecuaciones implícitas que definen el subespacio.

```
Implicitas = {x + y + z == 0, x - 2 z == 0}
```

Usamos la orden Solve, para despejar unas variables en términos de las otras.

```
Sol = Solve[Implicitas, {x, y, z}]
```

La siguiente instrucción nos permite ver la forma que tiene un vector genérico del subespacio.

```
generico = {x, y, z} /. Sol[[1]]
```

La orden Solve no nos da soluciones para todas las variables y nos proporciona soluciones en términos de un parámetro z. Como un vector genérico tiene la forma {2z, -3z, z}=z*{2, -3, 1}, podemos afirmar que el subespacio vectorial que definen esas ecuaciones implícitas tiene una base formada por el vector {2, -3, 1}.

Veamos otro ejemplo de un subespacio de $\mathbb{R}^4$ definido por dos ecuaciones implícitas.

```
Implicitas = {x + y - z + t == 0, x - t == 0};
Sol = Solve[Implicitas, {x, y, z, t}]

generico = {x, y, z, t} /. Sol[[1]]
```

Un vector genérico del subespacio tiene la forma {t, -2t + z, z, t}={t, -2t, 0, t}+{0, z, z, 0}=t*{1, -2, 0, 1}+z*{0, 1, 1, 0}. Por tanto tenemos dos vectores que generan el subespacio. Para obtener una base a partir de estos generados, usamos las siguientes instrucciones.

```
generadores = {{1, -2, 0, 1}, {0, 1, 1, 0}};
RowReduce[generadores]
```

Como los vectores {1, -2, 0, 1} y {0, 1, 1, 0} son independientes, ya tenemos una base del subespacio vectorial, aunque otra base sería la formada por los vectores {1,0,2,1} y {0,1,1,0}.

Ejercicio: Halle un sistema de generadores del subespacio de $\mathbb{R}^4$ dado por las ecuaciones implícitas
x + y + z + t = 0,
x + 2z - t = 0,
y + z + 2t = 0.

Pertenencia de un vector a un subespacio vectorial

Dado un vector y un sistema de generadores de un subespacio vectorial, nos planteamos en esta sección si el vector pertenece o no pertenece al subespacio vectorial. Estudiaremos el problema de tres formas diferentes.

La primera forma de abordar el problema consiste en comprobar si el vector en cuestión se puede poner como combinación lineal de los generadores.

Supongamos que el subespacio vectorial U está generado por los vectores {1, 1, 1}, {1, 0, -1} y {0, 1, 2}, y queremos saber si el vector {1, 2, 3} pertenece a U.

En primer lugar definimos el conjunto de vectores que generan el subespacio U.

```
generadores = {{1, 1, 1}, {1, 0, -1}, {0, 1, 2}}
vector = {1, 2, 3}
```

Si el vector pertenece al subespacio vectorial, se podrá poner como una combinación lineal de los generadores, es decir {1, 2, 3}=a*{1, 1, 1}+b*{1, 0, -1}+c*{0, 1, 2}. Así pues formamos una combinación lineal con los vectores generadores.

```
comblineal = Transpose[generadores].{a, b, c}
```

Ahora comprobamos si hay valores de a, b y c que hacen que esta combinación lineal sea igual al vector dado {1, 2, 3}.

```
Solve[comblineal == vector, {a, b, c}]
```

En este caso, como la orden Solve nos da soluciones (de hecho infinitas soluciones), resulta que sí se puede poner el vector {1, 2, 3} como combinación lineal de los vectores generadores de U. Por ejemplo si hacemos c = 0, y por tanto a = 2 y b = -1, tenemos una combinación lineal que nos da el vector {1, 2, 3}.

```
Transpose[generadores].{2, -1, 0}
```

Veamos ahora si el vector {1, 2, 2} está en el mismo subespacio U.

```
generadores = {{1, 1, 1}, {1, 0, -1}, {0, 1, 2}};
vector = {1, 2, 2};
comblineal = Transpose[generadores].{a, b, c};
Solve[comblineal == vector, {a, b, c}]
```

En este caso el sistema no tiene solución, por lo que no es posible encontrar una combinación lineal de los generadores que dé el vector {1, 2, 2}, es decir, el vector {1, 2, 2} no pertenece a U.

Ejercicio: ¿Pertenece el vector {4, -2, 2} al subespacio vectorial generado por los vectores {1, 0, 1} y {1, -1, 0}?

Otra forma de saber si un vector pertenece al subespacio generado por un conjunto de vectores, es comprobar si el vector es linealmente dependiente de los generadores. Si el vector es linealmente independiente de los generadores entonces no pertenece al subespacio generado por ese conjunto de vectores.

Como en el caso anterior, definimos el conjunto de generadores del subespacio U, y definimos el vector que queremos comprobar si pertenece al conjunto de generadores.

```
generadores = {{1, 1, 1}, {1, 0, -1}, {0, 1, 2}}
vector = {1, 2, 3}
```

Con la orden RowReduce vemos cuántos vectores linealmente independientes hay en U, es decir, cuál es la dimensión de U.

```
RowReduce[generadores]
```

Ahora consideramos el conjunto de los generadores de U junto con el vector {1, 2, 3}

```
ampliado = Append[generadores, vector]
```

De nuevo con la orden RowReduce comprobamos cuál es la dimensión del subespacio generado por este conjunto ampliado.

```
RowReduce[ampliado]
```

Vemos cómo la dimensión sigue siendo 2, es decir el vector {1, 2, 3} que hemos añadido no aumenta la dimensión, por lo tanto es un vector linealmente dependiente de los generadores de U, y podemos afirmar que el vector pertenece al subespacio.

Ejercicio: ¿Pertenece el vector {0, -3, 2} al subespacio vectorial generado por los vectores {1, 1, 1} y {1, -1, 0}?

Por último, supongamos que tenemos las ecuaciones implícitas de un subespacio vectorial y un vector del que queremos saber su pertenencia o no al subespacio. Si el vector cumple las ecuaciones podremos afirmar que pertenece al subespacio.

```
ecuaciones = {x + y + z == 0, x - z == 0}
vector = {1, 2, 3}
```

La siguiente instrucción sirve para comprobar si el vector cumple o no las ecuaciones

```
ecuaciones /. {x -> 1, y -> 2, z -> 3}
```

En este caso el vector no cumple ninguna de las dos ecuaciones, y no pertenece al subespacio vectorial.

Comprobamos ahora si el vector {1, 1, -2} pertenece al subespacio

```
ecuaciones /. {x -> 1, y -> 1, z -> -2}
```

En este caso el vector cumple una de las ecuaciones pero no la otra, así que este vector tampoco pertenece al subespacio vectorial.

Ejercicio: Compruebe si el vector {1, 2, 1, 0} pertenece al subespacio dado por las ecuaciones x - y + z = 0, x - y - t = 0.

Matriz de cambio de base.

En esta sección veremos dos maneras de plantear la obtención de la matriz de cambio de base. La primera consiste en poner los vectores de una base como combinación lineal de los vectores de la otra base, para lo que habrá que resolver un sistema de ecuaciones, y la segunda pasa por la base canónica y emplea propiedades de la matriz de cambio de base.

Supongamos que B1 = {v1, v2, ..., vn} es una base de un espacio vectorial y B2 = {w1, w2, ..., wn} es otra base del mismo espacio vectorial. Un vector v de ese espacio vectorial podrá ponerse como combinación lineal de los vectores de B1, por ejemplo v = a1 v1 + a2 v2 + ... + an vn, y también podrá ponerse como combinación lineal de los vectores de B2, por ejemplo v = b1 w1 + b2 w2 + ... + bn wn. Buscamos una matriz P que nos permita hallar los números a1, a2, ..., an si conocemos los números b1, b2, ..., bn, en concreto si a = {a1, a2, ..., an} y b = {b1, b2, ..., bn}, buscamos la matriz P que cumple que a = P.b.

Esta matriz P se llama matriz de cambio de base de B2 a B1, y se sabe que tiene por columnas las coordenadas de los vectores de B2 en la base B1. Por ejemplo, si ponemos el vector w1 en términos de los vectores de B1, tendremos una combinación lineal como w1 = p11 v1+p21 v2+... + pn1 vn, donde los números {p11, p21, ..., pn1} son la primera columna de la matriz P.

Si en la combinación w1 = p11 v1+p21 v2+...+pn1 vn, ponemos los vectores como columnas e igualamos coordenada a coordenada, tendremos un sistema lineal de ecuaciones donde las incógnitas son p11, p21, ..., pn1, y la matriz de coeficientes está formada por las coordenadas de los vectores de B1 colocados por columnas.

$$\begin{pmatrix} . & . & & . \\ . & . & & . \\ \mathrm{v1} & \mathrm{v2} & . & \mathrm{vn} \\ . & . & & . \\ . & . & & . \end{pmatrix} . \begin{pmatrix} \mathrm{p11} \\ \mathrm{p21} \\ . \\ . \\ \mathrm{pn1} \end{pmatrix} = \begin{pmatrix} . \\ . \\ \mathrm{w1} \\ . \\ . \end{pmatrix}$$

Para hallar la columna {p11, p21, ..., pn1}, que ya hemos dicho que es la primera columna de P, debemos resolver el sistema lineal, donde la matriz de coeficientes es la traspuesta de B1, y el término independiente es el vector w1. Es decir, la primera columna de P es

```
LinearSolve[ Transpose[B1] , w1]
```

y como w1 es el primer vector de la base B2, w1 = B2 [[1]], la primera columna de P es

```
LinearSolve[ Transpose[B1] , B2[[1]] ]
```

La matriz P se puede obtener colocando por columnas las soluciones de estos sistemas lineales. En el programa que sigue, ponemos todas estas soluciones en una tabla T, y luego hacemos su traspuesta para poner las soluciones por columnas.

```
B1 = {{1, 1, 1}, {1, 1, 0}, {1, 0, 0}}

B2 = {{1, 0, 1}, {0, 1, 1}, {0, 1, 0}}

n = Dimensions[B1][[1]]

T = Table[ LinearSolve[Transpose[B1] , B2[[i]] ], {i, 1, n}]

P = Transpose[T]
```

Una vez calculada P, la matriz de cambio de base de B2 a B1, vamos a aplicarla para obtener las coordenadas de un vector en la base B1.

Si queremos hallar las coordenadas del vector w1 = {1, 0, 1} en la base B1, debemos tener en cuenta que las coordenadas del vector w1 en la base B2 = {w1, w2, w3} son {1, 0, 0}. Al multiplicar la matriz P por la columna {1, 0, 0}, se obtiene

```
P.{1, 0, 0}
```

que son las coordenadas del vector en la base B1, es decir que w1=1 v1-1v2+1 v3. Comprobamos que esta combinación lineal da el vector w1 = {1, 0, 1}.

```
1 * {1, 1, 1} - 1 * {1, 1, 0} + 1 * {1, 0, 0}
```

Ejercicio: Hallar la matriz de cambio de base de B2 = {{1, 2, 1}, {0, 1, 2}, {2, 1, 0}} a B1={{1, 0, -1}, {-1, 1, 0}, {1, 1, 0}}.

Ahora vamos a obtener la matriz P pasando por la base canónica BC, usando ciertas propiedades de las matrices de cambio de base. Si tenemos las coordenadas de un vector en la base B2, usando una matriz de cambio de base (de B2 a BC) podemos tener sus coordenadas en la base canónica, y usando otra matriz de cambio de base (de BC a B1), tendremos sus coordenadas en la base B1.

Como las matrices se aplican a los vectores multiplicando la matriz por la izquierda, se tiene la siguiente relación entre las matrices de cambio de base,

```
P=Matriz de B2 a B1 = (Matriz de BC a B1).(Matriz de B2 a BC)
```

Ahora, la matriz de cambio de base de B2 a la base canónica tiene por columnas las coordenadas de los vectores de B2 en la base canónica, así pues basta con poner los vectores de B2 por columnas.

```
DeB2aBC = Transpose[B2]
```

La matriz de cambio de base de la base canónica a la base B1 es la inversa de la matriz de cambio de base de la base B1 a la base canónica.

```
DeBCaB1 = Inverse[ Transpose[B1] ]
```

```
DeB2aB1 = Inverse[ Transpose[B1] ] . Transpose[B2]
```

Esta es la matriz P buscada

```
P = Inverse[ Transpose[B1] ] . Transpose[B2]
```

Ejercicio: Considere las bases B1 = {{1, 1, 2}, {1, 0, 3}, {0, 0, 1}} y B2 = {{1, 0, 2}, {-1, 2, 0}, {0, 1, 0}}, halle la matriz de cambio de base de B2 a la base canónica, y también la matriz de cambio de base de la canónica a B1. Multiplique convenientemente las matrices halladas para obtener la matriz de cambio de base de B2 a B1.

Ortogonalización de Gram-Schmidt

Dada una base $\{v_1, v_2, ..., v_n\}$, el método de Gram-Schmidt permite construir una base ortogonal $\{w_1, w_2, ..., w_n\}$. Antes de escribir las fórmulas del método, recordamos que el producto escalar de dos vectores <v,w> se puede calcular multiplicando los vectores con un punto, y que la norma de un vector v al cuadrado ||v||^2 se puede calcular como ||v||^2 = <v,v> = v.v usando también el punto. Las fórmulas que permiten calcular $w_1, w_2, ..., w_n$ a partir de $v_1, v_2, ..., v_n$ son las siguientes:

$$w_1 = v_1$$

$$w_2 = v_2 - \frac{v_2 . w_1}{w_1 . w_1} w_1$$

...

Si ya hemos calculado los vectores $w_1, w_2, ..., w_{i-1}$

$$w_i = v_i - \frac{v_i . w_1}{w_1 . w_1} w_1 - \frac{v_i . w_2}{w_2 . w_2} w_2 - ... - \frac{v_i . w_{i-1}}{w_{i-1} . w_{i-1}} w_{i-1}$$

Como ejemplo, aplicamos el método que acabamos de ver a una base formada por tres vectores.

```
B = {{1, 1, 1}, {1, 0, 1}, {0, 1, 1}}
```

La siguiente instrucción define una lista donde se guardarán los vectores BO[[1]], BO[[2]], BO[[3]].

```
BO = Table[0, {i, 1, 3}, {j, 1, 3}]
```

Ahora utilizamos las fórmulas del método de Gram-Schmidt para construir la base ortogonal.

```
BO[[1]] = B[[1]]
```

$$BO[[2]] = B[[2]] - \frac{B[[2]].BO[[1]]}{BO[[1]].BO[[1]]} BO[[1]]$$

$$BO[[3]] = B[[3]] - \frac{B[[3]].BO[[1]]}{BO[[1]].BO[[1]]} BO[[1]] - \frac{B[[3]].BO[[2]]}{BO[[2]].BO[[2]]} BO[[2]]$$

Para comprobar que forman una base podemos calcular el determinante de BO, y para comprobar que forman una base ortogonal podemos calcular los productos escalares de los vectores de la base, o multiplicar BO por su traspuesta.

```
Det[BO]
```

```
BO[[1]].BO[[2]]
```

```
BO[[1]].BO[[3]]
```

```
BO[[2]].BO[[3]]

BO.Transpose[BO]
```

Si necesitamos una base ortonormal, debemos dividir cada uno de los vectores de la base ortogonal entre su norma.

$$\frac{\text{BO[[1]]}}{\sqrt{\text{BO[[1]].BO[[1]]}}}$$

$$\frac{\text{BO[[2]]}}{\sqrt{\text{BO[[2]].BO[[2]]}}}$$

$$\frac{\text{BO[[3]]}}{\sqrt{\text{BO[[3]].BO[[3]]}}}$$

Ejercicio: A partir de la base B = {{1, 0, 1}, {2, 1, 3}, {-2, 0, -3}} halle una base ortogonal. A partir de la base ortogonal obtenida, halle una base ortonormal.

Ya hemos visto cómo se pueden aplicar las fórmulas generales para ortogonalizar una base formada por tres vectores. Ahora programaremos estas fórmulas para ortogonalizar una base formada por n vectores. Comenzamos definiendo una lista B con los vectores de la base.

```
B = {{1, 0, 1, -1}, {0, 1, 2, 0}, {3, -1, 1, 1}, {-2, 2, 0, 3}}

Det[B]

n = Length[B]
```

Definimos la lista que contendrá la base ortogonal BO.

```
BO = Table[0, {i, 1, n}, {k, 1, n}]
```

El siguiente bucle calcula los vectores ortogonales según el método de Gram-Schmidt.

```
For[i = 1, i <= n, i = i + 1,
    BO[[i]] = B[[i]] - Sum[ (B[[i]].BO[[j]]) / (BO[[j]].BO[[j]]) BO[[j]], {j, 1, i - 1}]
   ]

BO
```

Comprobamos que los vectores de BO son ortogonales.

```
BO.Transpose[BO]
```

Hallamos una base ortonormal dividiendo cada uno de los vectores de la base BO por su norma.

```
For[i = 1, i <= n, i = i + 1,
    BO[[i]] = BO[[i]] / Sqrt[BO[[i]].BO[[i]]]
]
```

Por último, podemos comprobar que BO es una base ortonormal.

```
BO.Transpose[BO]
```

Para ortogonalizar una base y luego obtener una base ortonormal, puede usarse la orden GramSchmidt que se encuentra en un paquete de *Mathematica.*

El paquete debe cargarse antes de usarse, sólo antes de usarse, y sólo una vez.

```
Needs["LinearAlgebra`Orthogonalization`"]

B = {{1, 1, 1}, {1, 0, 1}, {0, 1, 1}}
```

La siguiente instrucción permite obtener una base ortonormal a partir de la base B.

```
GramSchmidt[B]
```

Ejercicio: Usando el paquete adecuado de *Mathematica*, obtenga una base ortonormal a partir de la base B = {{1, -3, 2, 1}, {0, 0, 1, 1}, {0, 1, 0, 1}, {2, 1, 1, 2}}. Compruebe que la base obtenida es ortonormal.

Aproximación por mínimos cuadrados.

Ajuste lineal por mínimos cuadrados

Supongamos que tenemos un conjunto de puntos del plano (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) y queremos hallar una recta de la forma $y = a\,x + b$ que pase por esos puntos del plano o que pase "lo más cerca posible".

Si los puntos estuvieran en una línea recta, podrían encontrarse valores a y b que serían una solución exacta del siguiente sistema lineal :

$a\,x_1 + b = y_1$,
$a\,x_2 + b = y_2$,
...
$a\,x_n + b = y_n$.

Este sistema se puede expresar de forma matricial, si definimos

$$A = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \cdots & \cdots \\ x_n & 1 \end{pmatrix}, \quad X = \begin{pmatrix} a \\ b \end{pmatrix} \text{ y } B = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}$$

entonces el sistema planteado se escribiría A.X = B.

En el caso en que los puntos no estén en una línea recta, no será posible encontrar valores de a y b que hagan que A.X sea igual a B, pero nos planteamos el problema de encontrar valores a y b que hagan que A.X esté "cerca" de B, en el sentido de que la norma || A.X-B || sea la menor posible.

Este problema, llamado "ajuste lineal por mínimos cuadrados", puede plantearse y resolverse de forma rigurosa usando los conceptos de proyección ortogonal y el teorema de la mejor aproximación, de donde puede deducirse que la solución X de este problema verifica que $A^t.A.\ X = A^t.B$, que es un sistema compatible determinado siempre que dos valores x_i sean distintos, de este sistema podría hallarse X, por ejemplo con una orden LinearSolve.

Veamos un ejemplo. Definimos primero los valores de x_i e y_i que se van a ajustar.

```
x = {1, 2.1, 3, 4, 6}
y = {2.7, 4.8, 6.7, 9.8, 12.4}
n = Dimensions[x][[1]]
```

La siguiente instrucción sirve para dibujar los puntos (x_i, y_i).

```
Puntos = ListPlot[Transpose[{x, y}], PlotStyle -> PointSize[0.02] ]
```

Siguiendo la notación que hemos usado al plantear el problema, definimos la matriz A y el vector B.

```
A = Table[ {x[[i]], 1} , {i, 1, n}]
```

```
B = y
```

Los valores de a y b que estamos buscando, son las coordenadas del siguiente vector, que obtenemos resolviendo un sistema usando la siguiente orden.

```
X = LinearSolve[Transpose[A].A, Transpose[A].B]
```

```
a = X[[1]]
b = X[[2]]
```

Con las siguientes instrucciones definimos la recta del ajuste lineal y la comparamos con los datos del problema.

```
f[x_] := a x + b
Recta = Plot[f[x], {x, 0, 7}]
```

```
Show[Recta, Puntos]
```

Ejercicio: Para estudiar la deformación de una viga delgada en voladizo, se ha construido un modelo en el laboratorio al que se le han aplicado distintas fuerzas verticales (F) en el extremo libre, y se ha medido la flecha (d) en el extremo libre. Se sabe que hay una relación entre los valores de la fuerza F, y del desplazamiento d, que viene dado por la fórmula d=$\frac{F L^3}{3 E I}$, donde L es la longitud de la barra (en el experimento igual a 0.4 metros), y el producto E*I se conoce como 'rigidez' de la barra.

a) Considere los siguientes datos experimentales y halle la recta d=a F+b que mejor ajusta los datos en el sentido de los mínimos cuadrados.

F (newtons)	0	0.5	1	1.5	3	4
d (metros)	0	0.045	0.089	0.13	0.28	0.35

b) Determine la rigidez de la barra asumiendo que b es despreciable y que a = $\frac{L^3}{3 E I}$. (Sol: E*I=0.2397 N m^2)

Ajuste parabólico por mínimos cuadrados

Nos planteamos ahora el problema de hallar una parábola y = a x^2 + b x + c que ajuste un conjunto de puntos del plano (x_1, y_1), (x_2, y_2), ..., (x_n, y_n).

Si los puntos estuvieran sobre una parábola, podrían encontrarse valores a, b y c que serían una solución exacta del siguiente sistema lineal:

a x_1^2 + b x_1 + c = y_1,
a x_2^2 + b x_2 + c = y_2,
...
a x_n^2 + b x_n + c = y_n.

Este sistema se puede expresar como A.X = B de forma matricial, definiendo

$$A = \begin{pmatrix} x_1{}^\wedge 2 & x_1 & 1 \\ x_2{}^\wedge 2 & x_2 & 1 \\ ... & ... & ... \\ x_n{}^\wedge 2 & x_n & 1 \end{pmatrix}, \quad X = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \text{ y } B = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{pmatrix}.$$

Si los puntos no se encuentran todos sobre una parábola, el sistema anterior no tendrá una solución, pero podremos hallar una solución X que minimize || A.X - B || resolviendo el sistema $A^t.A.\ X = A^t.B$, que es un sistema compatible determinado siempre que haya tres valores x_i distintos.

Como ejemplo ajustamos una parábola a los mismos datos del ejemplo anterior, observando la forma en que se define ahora la matriz A.

```
x = {1, 2.1, 3, 4, 6}
y = {2.7, 4.8, 6.7, 9.8, 12.4}
n = Dimensions[x][[1]]

Puntos = ListPlot[Transpose[{x, y}], PlotStyle -> PointSize[0.02] ]

A = Table[ {x[[i]]^2, x[[i]], 1} , {i, 1, n}]

B = y

X = LinearSolve[Transpose[A].A, Transpose[A].B]

a = X[[1]]
b = X[[2]]
c = X[[3]]

f[x_] := a x^2 + b x + c
Parabola = Plot[f[x], {x, 0, 7}]

Show[Parabola, Puntos]
```

Ejercicio: Halle la parábola que mejor ajuste los siguientes puntos, en el sentido de los mínimos cuadrados: (-1, -4.5), (0, -3), (2.5, 17.2) (3, 25) y (4.5, 50).

Ejercicios

1- Sea U el subespacio vectorial generado por los vectores u1 = {1, 1, -1}, u2 = {0, 1, 1}, u3 = {1, 2, 0}.
a) Halle la dimensión de U, y deduzca si los vectores u1, u2 y u3 son linealmente independientes.
b) Obtenga unas ecuaciones implícitas de U.
c) Obtenga unas ecuaciones paramétricas de U usando sólo dos parámetros.
d) Determine si el vector {1, 0, 0} pertenece a U.

2- Halle una base del subespacio de $\mathbb{R}^4$ dado por las siguientes ecuaciones implícitas

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

3- Sea u el vector que tiene coordenadas {1, 2, 3} en la base B1={{1, 1, -1}, {0, 1, 1},{1, 0, 0}}, y sea v el vector que tiene coordenadas {1, 3, 0} en la base B2={{1, 0, 1}, {1, 1, 0}, {0, 1, 1}}. Determine si u y v son el mismo vector.

4- Usando las fórmulas del método de Gram-Schmidt, obtenga una base ortogonal a partir de los vectores de la base B = {{1, 1, 0}, {1, -2, 0}, {0, 1, -1}}. Emplee el paquete correspondiente de *Mathematica* para obtener una base ortonormal a partir de B.

5- Halle la recta y la parábola que mejor ajustan, en el sentido de los mínimos cuadrados, los puntos (-2, 16), (4.5, 59), (5, 73), (7, 140), (9, 235), (15, 660). Represente conjuntamente los puntos dados, la recta y la parábola.