INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

PROGRAMA DE POSGRADO EN ELECTRONICA, COMPUTACION, INFORMACION Y COMUNICACIONES.

Mínimos Cuadrados Parciales en el Control Estadístico Multivariado

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO ACADEMICO DE:

MAESTRO EN CIENCIAS CON ESPECIALIDAD EN ESTADISTICA APLICADA

POR:

Eduardo Sánchez Sanmiguel

MONTERREY, N. L. JULIO DE 2004

INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

PROGRAMA DE POSGRADO EN ELECTRONICA, COMPUTACION, INFORMACION Y COMUNICACIONES.

Mínimos Cuadrados Parciales en el Control Estadístico Multivariado

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO ACADEMICO DE:

MAESTRO EN CIENCIAS CON ESPECIALIDAD EN ESTADISTICA APLICADA

POR:

Eduardo Sánchez Sanmiguel

MONTERREY, N. L. JULIO DE 2004

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY

CAMPUS MONTERREY

PROGRAMA DE GRADUADOS EN ELECTRÓNICA,
COMPUTACIÓN, INFORMACIÓN Y COMUNICACIONES

Mínimos Cuadrados Parciales en el Control Estadístico Multivariado

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO ACADEMICO DE:

MAESTRO EN CIENCIAS CON ESPECIALIDAD EN ESTADÍSTICA APLICADA

POR:

Eduardo Sánchez Sanmiguel

MONTERREY, N.L. JULIO DE 2004

Mínimos Cuadrados Parciales en el Control Estadístico Multivariado

POR:

Eduardo Sánchez Sanmiguel

TESIS

Presentada al Programa de Graduados en Electrónica, Computación, Información y Comunicaciones.

Este trabajo es requisito parcial para obtener el grado de Maestro en Ciencias con especialidad en Estadística Aplicada

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Julio de 2004

Resumen.

El presente trabajo se desarrolla en el ámbito de la estadística multivariada. Pretende ser un estudio completo sobre la técnica emergente llamada PLS (partial least squares o proyection to latent structures) de manera que se evidencie su funcionamiento, abarcando desde su fundamento matemático hasta sus algoritmos de operación. Se implementan computacionalmente dichos algoritmos haciendo uso de un software estadístico de uso extendido (s-plus). Se compara la efectividad de ésta con otras técnicas estadísticas de mayor consolidación. También se implementa computacionalmente la adaptación de esta técnica en su aplicación a gráficos de control para procesos multivariados. Por último, se bosqueja la situación de dicha técnica en el caso extendido mencionado en la literatura como MPLS (multiway partial least squares).

Índice.

Dedicatoria iv
Agradecimientos v
Resumen vi
Índice vii
Lista de figuras y gráficas viii
Capítulo 1. Introducción 1
1.1 Antecedentes 1
1.2 Definición del problema 2
1.3 Justificación 2
1.4 Objetivo 2
1.5 Limitaciones y delimitaciones 3
1.6 Definición de términos y estructura de la tesis 3
Capítulo 2. Análisis de fundamentos 5
2.1 PCA 6
2.1.1 Proyecciones 6
2.1.2 Representación gráfica 7
2.1.3 Dos ventajas de PCA 10
2.1.4 Proceso de PCA 10
2.1.4 Sustento matemático de PCA 12
2.1.5 Aplicaciones de PCA en la construcción de gráficas de control 16
2.2 PLS 17
2.2.1 Construyendo el modelo PLS 20
2.2.2 Algoritmo de operación PLS 23
2.2.3 Número de componentes de PLS 26
2.2.4 Fundamento matemático PLS 28
2.2.5 Algoritmo de aplicación PLS 31
2.2.6 Aplicaciones de PLS en la construcción de gráficos de control 32
2.2.7 Aplicaciones PCA y PLS 33
2.2.8 MPLS 35
Capítulo 3. Análisis de resultados 37
3.1 Conclusiones y recomendaciones 50
Capítulo 4. Anexos 52
4.1 Rutinas programadas 52
4.2 Datos utilizados 74
Capítulo 5. Bibliografía 78
Vita 80

Lista de figuras y gráficas.

Figura 1. Dispersión de mediciones de concentración de reactivo..... 8

Figura 2. Direcciones latentes PCA 9

Figura 3. Representación en tres dimensiones del proceso PCA..... 11

Figura 4. Gráfica de control para scores en PCA..... 16

Figura 5. Criterio ||Fa||..... 26

Figura 6. Criterio PRESS..... 27

Figura 7. Elipse de confianza..... 32

Figura 8. Reactor fluidificado..... 34

Figura 9. MPLS..... 36

Figura 10. Proyecciones sobre direcciones latentes PCA..... 37

Figura 11. Comparación de proyecciones PCA y PLS..... 38

Figura 12. Número de dimensiones incluídas..... 39

Figura 13. Identificación de outliers..... 40

Figura 14. Contribución de las X's a las diferentes dimensiones..... 41

Figura 15. Contribución de las Y's a las diferentes dimensiones..... 41

Figura 16. Contribución al error de predicción..... 44

Figura 17. SPE..... 45

Figura 18. Elipses de confianza..... 46

Figura 19. Scores individuales..... 48

Tabla 1. Resultados de PLS en aplicación..... 47

1. Introducción.

La creciente demanda en la aplicación del control de procesos para el aseguramiento de la calidad industrial ha ido permeando cada vez más en México la aplicación de dichas prácticas. Estos desarrollos, que están bien establecidos en la actualidad en el área del monitoreo univariable, tuvieron sus inicios alrededor de 1925 [5] en el trabajo de Walter A. Shewhart en los Laboratorios Bell.

Sin embargo en los últimos años, debido a la cada vez mayor complejidad de los procesos industriales se ha visto la necesidad de extender estas aplicaciones para que contemplen casos multivariados, es decir, situaciones donde se estudie el efecto simultáneo de muchas variables. El advenimiento de la introducción de las computadoras directamente en los procesos industriales hace posible tener registros de muchos datos que pudieran aprovecharse para la situación descrita.

Lo anterior así como otras aplicaciones y desarrollos de la estadística multivariada ha fomentado el surgimiento reciente de la técnica denominada PLS. Esta técnica, que ha surgido principalmente en aplicaciones de la industria química, cuenta a la fecha con un buen número de aplicaciones exitosas; sin embargo, la forma en que se ha desarrollado fuera del ámbito estadístico y la forma como se ha aplicado de forma aislada por diferentes grupos de usuarios ha frenado su difusión y consolidación como una técnica estadística reconocida.

1.1 Antecedentes

El análisis de datos confinado a una sola variable es una materia fascinante que además se torna en algo muy retador cuando se extiende a situaciones donde intervienen varias variables. La necesidad de entender las relaciones entre muchas variables vuelve al análisis multivariado un campo inherentemente difícil, ya que la mente humana se ve saturada por una cantidad tremenda de datos. Las representaciones gráficas que son de gran ayuda en el entendimiento o en la generación de ideas, tampoco son un recurso disponible cuando sobrepasamos las dos o tres dimensiones. Además la cantidad impresionante de cálculos necesarios en este tipo de procedimientos, mantuvieron durante mucho tiempo "a raya" el desarrollo de la estadística multivariada.

El desarrollo de las computadoras vino a proporcionar el impulso que faltaba a esta importante materia. Este impulso se debió en primer término a la tremenda capacidad para realizar operaciones inequívocamente y en segundo término a la urgente necesidad de interpretar y aprovechar los "análisis" que de éstas se obtenían. Así que no quedó otro camino que redoblar esfuerzos con imaginación y entendimiento.

Comenzaron a consolidarse las técnicas estadísticas multivariadas: Inferencias sobre vectores de medias, pruebas de hipótesis de varias variables, modelos de regresión multivariada, etc.

Esta forma de trabajo y desarrollo de la estadística había de propiciar que muchos comenzaran a experimentar con datos y a obtener resultados útiles e interesantes que debían esperar para ser analizados por la comunidad matemática y estadística para ser incorporados de lleno al cuerpo de las técnicas estadísticas multivariadas. Este es el camino que han seguido, por ejemplo el análisis de componentes principales, y en el que se encuentra en proceso la técnica de mínimos cuadrados parciales, que constituye el objeto de este trabajo.

1.2 Definición del problema

La forma en que se ha dado el desarrollo de *partial least squares* (PLS, de ahora en adelante): por el trabajo aislado de individuos en distintos lugares del mundo como Canadá, Suecia, Noruega, Alemania, etc., ha generado una especie de controversia e inquietud alrededor de esta técnica. Primero por que su forma de operación se basa principalmente en la aplicación de un algoritmo computacional que ha dado buenos resultados pero del cual no se tiene un claro entendimiento de su funcionamiento; y en segundo término porque dicho algoritmo presenta variaciones entre los diferentes grupos de desarrollo. Así mismo, la falta de bibliografía de carácter conciliador tanto entre las partes mencionadas, así como de éstos con la comunidad estadística, contribuye a dicha inquietud. Es nuestra intención salvar, en la medida de lo posible, dadas las limitaciones propias de un servidor, con el presente trabajo, el escollo que se ha mencionado.

1.3 Justificación

Cada vez con mayor frecuencia se presentan situaciones, tanto en las aplicaciones de la industria nacional, como en el área de las ciencias sociales, donde se contemplan oportunidades para aplicar técnicas estadísticas que obtengan provecho de grandes cantidades de mediciones tomadas sobre varias variables que interactúan para un fin específico. Como en muchas de estas situaciones, el registro de dichas mediciones se da por "default", sería muy conveniente acercar los últimos desarrollos internacionales en dicha materia a los requerimientos propios de nuestro país.

1.4 Objetivo

El objetivo principal de esta investigación es presentar en un solo documento un panorama general, sólido, de la técnica de PLS, de manera que sea de utilidad tanto para la ciencia estadística como para la industria que requiere de la implementación de esta importante aplicación en sus procesos.

1.5 Limitaciones y delimitaciones

En este trabajo se incluye la presentación de los fundamentos matemáticos de PLS, sus algoritmos de operación, su adaptación a la construcción de gráficos de control, su extensión hacia MPLS, así como algunos ejemplos de aplicaciones.

La falta de credibilidad de muchos empresarios, en la aplicación de la ciencia para mejorar el desempeño de los procesos industriales del país, así como los problemas económicos de una buena parte de este sector y el celo con que se manejan muchos procesos de producción, no permitieron lograr una aplicación de la técnica de PLS a algún proceso real. Esta aplicación requeriría de detener el proceso para establecer condiciones iniciales y llevar un monitoreo muy cercano de materiales y de variables de entrada y salida del mismo.

1.6 Definición de términos y estructura de la tesis

Se conoce como PLS (partial least squares o projection to latent structures) a la técnica surgida en el ámbito químico, específicamente en Quimiometría, que sirve como modelo predictivo en situaciones donde hay muchas variables correlacionadas. PCA se refiere a la técnica de análisis de componentes principales, desarrollada a principios del siglo XX y cuyas aplicaciones se dan en el ámbito de la reducción de variables. MPLS se refiere a multiway partial least squares como una extensión de PLS para contemplar una dimensión extra en la aplicación de PLS que muchas veces se toma como el tiempo y se está comenzando a presentar con frecuencia en el ámbito de las aplicaciones industriales, sobre todo en el área química.

Como es común en el ámbito matemático, los vectores aparecen en negritas y las matrices en mayúsculas.

El cuerpo principal de este trabajo está estructurado en capítulos de la siguiente forma: En el capítulo 1, aparece esta introducción, que es un preámbulo al propio tema de desarrollo de este documento. En él se incluyen los antecedentes, justificaciones y objetivos del porqué elegí éste como mi tema de tesis. Aparece a continuación, como capítulo 2, el análisis de fundamentos. Entendemos este apartado como la exposición del estado actual del tema de PLS en la literatura consultada, a la vez que en él mismo se van delineando las aportaciones que busca lograr este trabajo. Dentro de este mismo capítulo presentamos el trabajo que realizamos con el análisis de los algoritmos de operación de PLS y de la programación de sus adaptaciones en la construcción de gráficas de control en el software s-plus. La forma esencial en que se desarrolla esto, es exponiendo primero los fundamentos del análisis de componentes principales, que es algo más conocido, mejor consolidado y de más clara interpretación; para después homologar en todo lo que sea posible la técnica de proyección a estructuras latentes. Por último en el capítulo 3 de análisis de resultados, se muestran claramente los puntos importantes que se encontraron en el capítulo 2 sobre la aportación que puede significar este trabajo a la técnica y se

presentan algunas conclusiones y recomendaciones de carácter general. Aparecen al final, anexos y bibliografía.

2. Análisis de fundamentos.

En el campo de las aplicaciones estadísticas cada vez se presentan más situaciones donde se tienen muchas variables incidiendo sobre un producto o muchas variables que deben ser consideradas para un fin específico. Para estudiar este tipo de situaciones se han desarrollado técnicas que forman el cuerpo del análisis multivariado en estadística.

Aunque la principal diferencia entre la estadística univariada y la estadística multivariada parece venir indicada directamente en su nombre, el número de variables presentes es una o más de una, el punto central es que la estadística multivariada se ocupa de situaciones donde estos conjuntos de variables se encuentran correlacionadas entre sí y por esto, no es lícito su estudio de forma individual (de manera separada) sino que más bien debe hacerse de forma conjunta.

Muchas de las técnicas del análisis multivariado se han desarrollado como una mera extensión de las técnicas univariadas, así tenemos por ejemplo las pruebas t o la regresión multivariada; sin embargo algunas otras técnicas "han nacido" puramente multivariadas; por ejemplo el análisis de componentes principales (PCA) y el análisis de factores (FA). Como una derivación de PCA y de la regresión multivariada nace la regresión de componentes principales (PCR) y últimamente la técnica de mínimos cuadrados parciales (PLS).

FA y PLS nacieron fuera del campo propio de la teoría estadística. FA mediante aplicaciones en el campo de la psicología y PLS en el área de la quimiometría (Chemometrics). Ésta es la rama de la química que se encarga de las aplicaciones de la estadística en esta importante rama industrial.

FA provocó mucha controversia en el transcurso de su historia. Sus primeros desarrollos aparecen a principios del siglo XX, en los intentos de Karl Pearson y Charles Spearman y otros, para definir y medir la inteligencia [2]. Estas tempranas aplicaciones psicológicas de interpretación un tanto subjetiva para los estadísticos, impidieron aceptar a ésta como una técnica estadística. Este problema ha quedado resuelto conforme se han estudiado sus propiedades y se ha establecido como una técnica de aplicación validada estadísticamente.

Con PLS sucede algo similar y en la actualidad se cuenta con una serie de aplicaciones exitosas de esta técnica en áreas de espectrometría y calibración química. Las aplicaciones que se han hecho por grupos de trabajo diferentes han dado surgimiento a diferentes "rutinas de operación" de la técnica. Hemos dicho rutinas de operación, porque la forma básica de aplicación de la técnica ha sido mediante un algoritmo que ha dado buenos resultados pero del cual no se tiene idea clara de su forma de operación, ni de interpretación; tampoco se tiene conocimiento certero de las propiedades estadísticas de los estimadores empleados por la técnica para hacer predicciones.

Los estadísticos se han comenzado entonces a dar a la tarea de sustentar la metodología, de unificar los diferentes criterios (algoritmos) y de esclarecer sus distribuciones y propiedades estadísticas.

Se puede ganar una buena cantidad de entendimiento acerca de PLS entendiendo la estructura de proyecciones que ocurre en PCA, por lo que a continuación, y con miras a evidenciar su desempeño, se expone este sustento teórico de la operación de PCA.

2.1 PCA

El análisis de componentes principales es un método de reducción de variables. Su premisa básica de operación es: si un conjunto de datos contiene correlaciones significativas, es posible encontrar un nuevo conjunto más pequeño de variables que contengan casi toda la información original.

Para realizar PCA (como para PLS) es necesario que las variables sean cuantitativas, por lo que cualquier variable cualitativa debe ser codificada numéricamente o descartada de forma previa al análisis. La codificación puede ser discreta o continua. Además de esto, en ciertas aplicaciones es benéfico preprocesar los datos centrándolos con respecto a sus medias (por variable) y escalándolos por sus desviaciones estándar, de manera que las "direcciones latentes" subyacentes en los datos no se vean afectados por condiciones de escalas de medida diferentes.

2.1.1 Proyecciones

Sea $\mathbf{p}$ un vector unitario anclado en el origen de un espacio M-dimensional con componentes $p_j \quad j = 1,...,M$; y sea $\mathbf{X}$ una matriz de dimensiones $N \times M$. Esta matriz, que gráficamente representa una nube de N puntos en las M dimensiones, contiene en su i-ésima fila las coordenadas del i-ésimo punto, mientras que la j-ésima columna almacena las mediciones de la variable j, realizadas sobre los N "objetos".

La proyección de este i-ésimo punto sobre $\mathbf{p}$ viene dada por el producto escalar:

$$t_i = x_i \mathbf{p}$$

Entonces, el vector de proyección de $\mathbf{X}$ en $\mathbf{p}$ está dado por:

$$\mathbf{t} = \mathbf{X}\mathbf{p}$$

Las componentes de cada proyección t_i en la dirección de los ejes coordenados están dadas por,

$$(proy\, x_i)_j = (t_i)_j = t_i p_j$$

y la matriz $\mathbf{tp}^T$ tendrá las componentes,

$$\begin{bmatrix} t_1p_1 & t_1p_2 & \cdots\cdots & t_1p_M \\ t_2p_1 & t_2p_2 & \cdots\cdots & t_2p_M \\ \vdots & \vdots & \vdots & \vdots \\ t_Np_1 & t_Np_2 & \cdots\cdots & t_Np_M \end{bmatrix}$$

es decir, las coordenadas (referidas a los M ejes originales) de las proyecciones de los puntos sobre el vector $\mathbf{p}$.

Esta última matriz representa un modelo de la matriz original $\mathbf{X}$ (que hemos dicho que almacena las coordenadas mismas de los N puntos). Lo anterior debería adecuarse si el vector $\mathbf{p}$ no es unitario.

2.1.2 Representación gráfica

PCA describe la localización y la forma de la nube de puntos en le espacio M-dimensional para un conjunto de objetos. Este proceso envuelve dos pasos; primero, la traslación de la nube de puntos al origen, y segundo, su rotación alrededor del origen. Lo anterior se considera a veces como una traslación y rotación de los ejes coordenados más que de los puntos que representan los datos.

La rotación alinea la primera componente principal en la dirección de mayor variabilidad del conjunto de datos. Este eje se deja fijo mientras se determina el segundo (y subsecuentes) eje de componentes principales.

Los ejes de componentes principales se conocen como variables latentes, cuyas direcciones en el espacio M-dimensional están dadas por los vectores $\mathbf{p}$, mientras que las localizaciones de los datos a lo largo de estos ejes están dadas por las magnitudes de los vectores $\mathbf{t}$. Estas magnitudes son conocidas como scores (respetaremos su nombre inglés por no haber en español una traducción que sea aceptada de forma general).

Quizás el proceder de PCA sea mejor visualizado con el ejemplo que se presenta a continuación [16]. El ejemplo es confinado a sólo dos dimensiones. Componentes principales es más "redituable" con una cantidad mayor de variables (más de cinco), pero para poder evidenciar su funcionamiento de forma gráfica, presentaremos primero un caso sencillo. Más adelante mostraremos el caso de tres dimensiones.

Tenemos dos métodos de medición de la concentración de un reactivo y se toman las mediciones de quince muestras con ambos métodos:

Method A	10	10.4	9.7	9.7	11.7	11	8.7	9.5	10.1	9.6	10.5	9.2	11.3	10.1	8.5
Method B	10.7	9.8	10	10.1	11.5	10.8	8.8	9.3	9.4	9.6	10.4	9	11.6	9.8	9.2

La representación gráfica de estos datos se muestra a continuación:

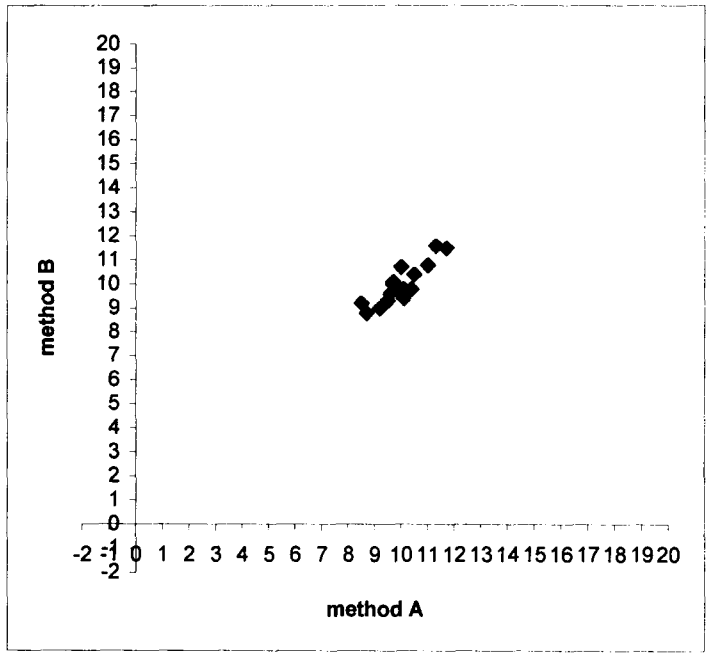


Figura 1

Observamos que los métodos de medición parecen estar relacionados entre sí de forma directa, por lo que, si separamos a las variables para estudiarlas univariadamente, perderíamos la información valiosa proporcionada por la relación existente entre ambos métodos de medición.

La matriz de varianzas y covarianzas de los datos anteriores aparece a continuación:

S =

0.79857143	0.67928571
0.67928571	0.73428571

Como sabemos, es en esta matriz donde está contenida información importante de las variables: su dispersión y sus relaciones entre sí.

Los eigenvalores y eigenvectores de la matriz **S** son:

EIGENVALORES =

l_1 y l_2

1.44647434
0.0863828

EIGENVECTORES =

$\mathbf{p}_1$	$\mathbf{p}_2$
0.723624808	0.69019355
0.69019355	-0.7236248

Los ejes originales de las variables serán "rotados" un ángulo de 43.65° (que viene dado por los cosenos inversos de $\mathbf{p}_1$ y $\mathbf{p}_2$) para ubicar en ellos las mediciones de las variables, referidas ahora a estos nuevos ejes coordenados. En la figura siguiente aparecen rotados y desplazados para cruzarse en el punto ($\text{promedio}_{\text{method A}}$, $\text{promedio}_{\text{method B}}$)

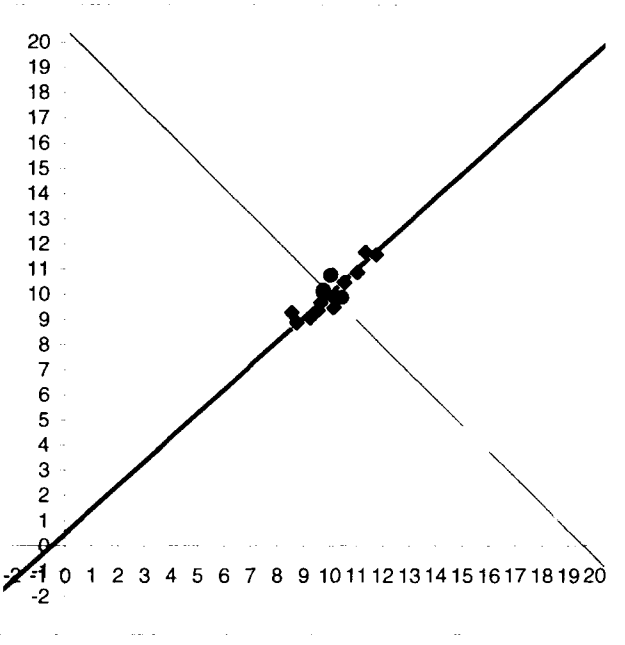


Figura 2

Las coordenadas de las mediciones, referidas a estos nuevos ejes (o sea, los scores) son:

X'	0.48	0.15	-0.22	-0.15	2.27	1.28	-1.77	-0.84	-0.34	-0.57	0.64	-1.27	2.05	-0.07	-1.64
Y'	0.51	-0.42	0.21	0.28	-0.09	-0.11	0.03	-0.16	-0.50	-0.01	-0.06	-0.17	0.26	-0.21	0.46

Las cuales son obtenidas mediante: $X' = \mathbf{P}^{\mathrm{T}}\left(X - \bar{X}\right)$ y $Y' = \mathbf{P}^{\mathrm{T}}\left(Y - \bar{Y}\right)$

2.1.3 Dos ventajas de PCA

Dos aspectos importantes a subrayar de esta técnica es, primero que "transforma" un conjunto de mediciones de variables correlacionadas en un nuevo conjunto de mediciones de variables independientes. Matricialmente:

$$\mathbf{P}^{\mathrm{T}}\mathbf{S}\mathbf{P} = \mathbf{L}$$

Donde $\mathbf{S}$ es la matriz de covarianzas de las variables originales correlacionadas y $\mathbf{L}$ es la matriz de covarianzas de las nuevas variables independientes. $\mathbf{L}$ es una matriz diagonal cuyos elementos distintos de cero (en la diagonal) son los eigenvalores de $\mathbf{S}$, mientras que $\mathbf{P}$ es la matriz formada por los eigenvectores de $\mathbf{S}$ correspondientes a los eigenvalores que forman $\mathbf{L}$. Las varianzas de las "variables" en $\mathbf{L}$ son las mismas que las varianzas de X' mencionada antes. En la forma electrónica acompañante de este material se presenta una pequeña rutina en Excel para mostrar que esta propiedad es válida para cualquier matriz $\mathbf{P}$ (no tiene que ser necesariamente la formada por los eigenvectores de $\mathbf{S}$). También se incluye el programa utilizado aquí para encontrar las componentes principales de un conjunto de mediciones y representarlas gráficamente.

Como segundo punto importante podemos decir que, en muchos casos, se puede prescindir de una (o más) de las componentes porque los "puntos" yacen sobre una sola (o sobre unas pocas) dimensión y sin mucha pérdida de información (y sí con mucho ahorro o simplificación) esto conduce a interpretaciones simplificadas de problemas, en donde la ganancia en conocimiento está directamente relacionada con la simplificación lograda.

2.1.4 Proceso de PCA

En forma resumida, PCA opera de la siguiente manera (vea la figura 3[10], más abajo):

a) Tenemos un conjunto de datos referenciados al sistema $\mathbf{x_1}$-$\mathbf{x_2}$-$\mathbf{x_3}$

b) La primer dirección latente (nuevo eje de referencia) será aquella en la que se presente la variabilidad mayor de los datos ($\mathbf{t_1}$ = PC1 en la figura)

c) La siguiente dirección latente será aquella que tenga la variación mayor, después de la presentada en PC1 y además, restringida a que sea ortogonal a la primer dirección latente ($\mathbf{t_2}$ = PC2 en la figura)

d) Se procede secuencialmente hasta obtener el número de dimensiones latentes que explique completamente la variabilidad original del conjunto de

datos (que será igual al número de ejes en el sistema original; tres, en la figura).

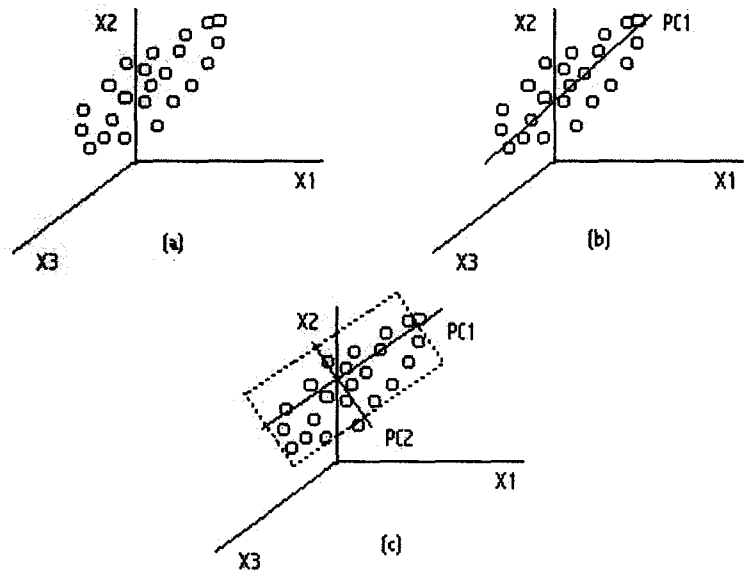


Figura 3

Obsérvese en la figura la forma en que están ubicados los ejes con respecto a la magnitud de la variabilidad presente en los datos. Esto redunda en que las "nuevas variables" ($\mathbf{t's}$) son independientes unas de otras (véase, como la nube de puntos no refleja ninguna tendencia, cuando se referencia dicho conjunto a los nuevos ejes).

Más aún, como ya se mencionó antes, el espacio generado por las mediciones puede estar descrito en forma aproximada por un número de dimensiones menor a la del sistema de referencia original (en la figura, el plano generado por $\mathbf{t}_1$ = PC1 y $\mathbf{t}_2$ =PC2 aproxima el espacio en el que están ubicados los datos, $\mathbf{x_1}$-$\mathbf{x_2}$-$\mathbf{x_3}$).

En conjuntos de datos de variables altamente correlacionadas, el subespacio donde "yacen" estas mediciones tiene dimensionalidad mucho menor que la del sistema de referencia original (numero de variables) y, entonces PCA se vuelve una poderosa técnica de reducción de datos, amén de las buenas propiedades que ofrece la nueva representación de las mediciones. Esto es, si comenzamos en un conjunto de N mediciones en M variables, tendremos casi la misma cantidad de información, ahora dada en un conjunto de N mediciones en A nuevas variables, donde A puede ser mucho menor que M ($A << M$).

Los ejes que describen la mayor variabilidad de un conjunto de datos vienen dados por los eigenvectores de la matriz de varianzas y covarianzas de los variables originales (vea la demostración más adelante). De *forma más precisa*,

podemos decir que los ángulos referidos a los ejes originales, de estas nuevas direcciones, vienen dados por el coseno inverso de los componentes de los eigenvectores. Esto, aunado al hecho de que las varianzas de los datos en la direcciones de los nuevos ejes, están dadas directamente por los eigenvalores correspondientes, sugiere el cálculo de estos valores y vectores propios como la forma operativa "natural" para implementar PCA.

Entonces, los nuevos ejes de referencia o variables latentes encontrados por PCA ($\mathbf{t}_1, \mathbf{t}_2$, etc.), son las combinaciones lineales de los elementos que forman cada fila de la matriz $\mathbf{X}$. Esto es, $\mathbf{Xp}_i$, donde $\mathbf{p}_i$ son los eigenvectores de $\mathbf{X}^T\mathbf{X}$ correspondientes a los eigenvalores mayores; esto es $\mathbf{t}_i = \mathbf{Xp}_i$.

Esto, según se expuso antes, también se puede ver como las proyecciones de la matriz $\mathbf{X}$ sobre el vector $\mathbf{p}_i$.

Así, las componentes principales definen "el plano" de mayor variabilidad y los vectores de carga (o loadings como se conocen por su nombre en inglés) $\mathbf{p}_i$, asociados con estas componentes principales definen la localización del plano, en términos de las variables originales, mientras que cada observación es localizada en este plano vía sus scores ($\mathbf{t}_i$). El score es la distancia desde el origen del plano a lo largo de cada componente principal, y es calculado como el producto del vector de cargas y la observación $\mathbf{x_1}$-$\mathbf{x_2}$-$\mathbf{x_3}$. La distancia perpendicular desde cada observación al plano, es el residual para aquella observación.

2.1.5 Sustento matemático de la técnica de PCA

Hemos visto en el apartado de proyecciones que si $\mathbf{X}$ es la matriz de N datos M-dimensionales:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots\cdots & x_{1M} \\ x_{21} & x_{22} & \cdots\cdots & x_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots\cdots & x_{NM} \end{bmatrix}$$

la matriz $\mathbf{tp}^T$ =

$$\begin{bmatrix} t_1p_1 & t_1p_2 & \cdots\cdots & t_1p_M \\ t_2p_1 & t_2p_2 & \cdots\cdots & t_2p_M \\ \vdots & \vdots & \vdots & \vdots \\ t_Np_1 & t_Np_2 & \cdots\cdots & t_Np_M \end{bmatrix}$$

tiene como elementos a las coordenadas de las proyecciones de $\mathbf{X}$ sobre el vector unitario $\mathbf{p}$. Estas coordenadas están referenciadas a los M ejes en que se midieron los datos de la matriz $\mathbf{X}$.

Cada elemento $t_i p_j$ es una aproximación (un modelo) de x_{ij}. De igual forma, la matriz $\mathbf{tp}^{\mathrm{T}}$ es un modelo de $\mathbf{X}$.

No hemos restringido al vector $\mathbf{p}$ a que tenga alguna dirección específica, pero podemos restringirlo a la dirección que cumpla que $\mathbf{tp}^{\mathrm{T}}$ sea un buen modelo de $\mathbf{X}$. Esto podría definirse como que la suma de las "discrepancias" entre los elementos de $\mathbf{X}$ y de $\mathbf{tp}^{\mathrm{T}}$ sea lo más pequeña posible. Es decir, buscamos minimizar la suma de los errores (cuadráticos, porque no nos interesa penalizar alguna dirección específica en estas discrepancias o errores).

Sea entonces $\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x_{ij}-t_i p_j\right)^2$ la función que buscamos minimizar para tener en $\mathbf{tp}^{\mathrm{T}}$ un buen modelo de $\mathbf{X}$. Como tenemos la restricción que el vector $\mathbf{p}$ sea unitario, aplicamos multiplicadores de Lagrange:

$$f=\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x_{ij}-t_i p_j\right)^2-\lambda\left(\sum_{j=1}^{M}p_j{}^2-1\right) \tag{1}$$

que, derivando parcialmente con respecto a t_k, produce:

$$\frac{\partial f}{\partial t_k}=\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\partial}{\partial t_k}\left(x_{ij}-t_i p_j\right)^2=-2\sum_{j=1}^{M}\left(x_{kj}-t_k p_j\right)p_j$$

desaparece la primer sumatoria porque sólo "sobrevive" el término k, de la misma.

$$\frac{\partial f}{\partial t_k}=-2\left(\sum_{j=1}^{M}x_{kj}p_j-\sum_{j=1}^{M}t_k p_j{}^2\right)=-2\left(\sum_{j=1}^{M}x_{kj}p_j-t_k\sum_{j=1}^{M}p_j{}^2\right)=-2\left(\sum_{j=1}^{M}x_{kj}p_j-t_k\right)$$

por ser $\mathbf{p}$ un vector unitario. Después, igualando a cero y despejando (para encontrar el óptimo):

$$t_k=\sum_{j=1}^{M}x_{kj}p_j \tag{2}$$

esto es; el elemento k-ésimo de $\mathbf{p}$ resulta de multiplicar la k-ésima fila de $\mathbf{X}$ por el vector unitario $\mathbf{p}$. Lo cual es concordante con lo que ya sabíamos.

Enseguida repetimos el proceso, pero ahora derivando parcialmente con respecto a p_h:

$$\frac{\partial f}{\partial p_h}=\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{\partial}{\partial p_h}\left(x_{ij}-t_i p_j\right)^2-\lambda\left(\sum_{j=1}^{M}\frac{\partial}{\partial p_h}p_j{}^2-\frac{\partial}{\partial p_h}1\right)=-2\sum_{i=1}^{N}\left(x_{ih}-t_i p_h\right)t_i-2\lambda p_h$$

nuevamente, las dos sumatorias con respecto a j desaparecen como antes

$$\frac{\partial f}{\partial p_h} = -2\left(\sum_{i=1}^{N} x_{ih}t_i - p_h\sum_{i=1}^{N} t_i^{\,2} + \lambda p_h\right) = -2\left[\sum_{i=1}^{N} x_{ih}t_i - p_h\left(\sum_{i=1}^{N} t_i^{\,2} - \lambda\right)\right]$$

que, al igualar a cero y despejar, resulta en:

$$\sum_{i=1}^{N} x_{ih}t_i = p_h\left(\sum_{i=1}^{N} t_i^{\,2} - \lambda\right) \qquad (3)$$

El artilugio siguiente es netamente para eliminar al multiplicador lambda:

Multiplicamos por el término p_h a ambos lados de la igualdad y después sumamos sobre h, también a ambos lados para obtener:

$$p_h\sum_{i=1}^{N} x_{ih}t_i = p_h^{\,2}\left(\sum_{i=1}^{N} t_i^{\,2} - \lambda\right) \rightarrow \sum_{h=1}^{M} p_h\sum_{i=1}^{N} x_{ih}t_i = \sum_{h=1}^{M} p_h^{\,2}\left(\sum_{i=1}^{N} t_i^{\,2} - \lambda\right)$$

y, reordenando:

$$\sum_{i=1}^{N} t_i\sum_{h=1}^{M} x_{ih}p_h = \sum_{h=1}^{M} p_h^{\,2}\left(\sum_{i=1}^{N} t_i^{\,2} - \lambda\right) \rightarrow \sum_{i=1}^{N} t_i^{\,2} = \sum_{i=1}^{N} t_i^{\,2} - \lambda \rightarrow \lambda = 0$$

en virtud de la ecuación (2) y de que $\mathbf{p}$ es unitario.

Por este último resultado, se simplifica la ecuación (3) en:

$$\sum_{i=1}^{N} x_{ih}t_i = p_h\sum_{i=1}^{N} t_i^{\,2}$$

Si sustituimos (2) en (3), $\sum_{i=1}^{N} x_{ih}\sum_{j=1}^{M} x_{ij}p_j = p_h\sum_{i=1}^{N} t_i^{\,2}$ que podemos escribir como:

$\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ih}x_{ij}p_j = p_h\sum_{i=1}^{N} t_i^{\,2}$ ó $\sum_{j=1}^{M} p_j\sum_{i=1}^{N} x_{ih}x_{ij} = p_h\sum_{i=1}^{N} t_i^{\,2}$ en donde reconocemos el

término $\sum_{i=1}^{N} x_{ih}x_{ij}$ como el elemento hj-ésimo de la matriz $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Si representamos

a este elemento como C_{hj}, entonces tenemos:

$$\sum_{j=1}^{M} C_{hj} p_j = p_h \sum_{i=1}^{N} t_i^2$$

al variar j en la primera sumatoria, estamos multiplicando toda la fila h de $\mathbf{X}^T\mathbf{X}$ por el vector $\mathbf{p}$, y el resultado de esto es la componente h del mismo vector $\mathbf{p}$. Como esto vale para cada componente h, se puede escribir en forma matricial como:

$$\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{p} = \left(\sum_{i=1}^{N} t_i^2\right)\mathbf{p}$$

Es decir, el vector $\mathbf{p}$ debe ser el eigenvector de $\mathbf{X}^T\mathbf{X}$ correspondiente al eigenvalor $\sum_{i=1}^{N} t_i^2$, que además es el eigenvalor de valor más grande, por lo que se verá enseguida.

Ya hemos dicho que la cantidad que buscamos minimizar, es: $\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x_{ij} - t_i p_j\right)^2$

Si desarrollamos el binomio, tendremos:

$$\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x_{ij} - t_i p_j\right)^2 = \sum_{i=1}^{N}\sum_{j=1}^{M}\left(x_{ij}^2 - 2x_{ij} t_i p_j + t_i^2 p_j^2\right) = \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij}^2 - 2\sum_{i=1}^{N} t_i \sum_{j=1}^{M} x_{ij} p_j +$$

$$\sum_{i=1}^{N} t_i^2 \sum_{j=1}^{M} p_j^2 = \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij}^2 - 2\sum_{i=1}^{N} t_i t_i + \sum_{i=1}^{N} t_i^2 = \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij}^2 - \sum_{i=1}^{N} t_i^2$$

en donde hemos aplicado, la ecuación (2) y el hecho de que $\mathbf{p}$ es unitario.
Este último desarrollo nos dice que tal cantidad se minimizará cuando la última sumatoria (que ya encontramos que es un eigenvalor) sea mayor.

En forma analítica, al repetir el proceso que en la demostración se llevó a cabo para un componente específico de los vectores $\mathbf{t}$ y $\mathbf{p}$; lo que ha sucedido, es que la matriz $\mathbf{X}$ ha sido aproximada por:

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \ldots + \mathbf{t}_A\mathbf{p}_A^T + \mathbf{E} \qquad (4)$$

donde $\mathbf{E}$ es la matriz de residuales después de haber ajustado sólo A dimensiones de las M originales.

En forma idealizada A se debería escoger de tal forma que no quede información significativa en $\mathbf{E}$. Por ejemplo, si $\mathbf{X}$ es una matriz de observaciones de las variables de un proceso industrial, $\mathbf{E}$ podría representar *error aleatorio* (*de*

medición, por ejemplo) y al adicionar otra componente $A-1$, se "ajustaría" algo de este error, incrementándose así el error de predicción, lo cual no es deseable.

Hay varias formas de seleccionar A (el número de dimensiones adecuadas a considerar). Métodos que van desde los más sencillos (forma gráfica) hasta aquellos más precisos en los que intervienen una gran cantidad de cálculos, como son los conocidos: "crossvalidation", "jacknife" y "bootstrap".

2.1.6 Aplicaciones de PCA en la construcción de gráficas de control

Al trabajar PCA, como un proceso de reducción de variables, encaja perfectamente en la necesidad que se tiene de procesos que construyan gráficos de control para situaciones de muchas variables relacionadas. Así se ha estado empleando desde algunos años. Además de reducir el número de variables que se requeriría monitorear, la otra cualidad de PCA, específicamente que sus "variables" resultantes son no correlacionadas, es la que ayuda mucho para que las gráficas de control operen casi como una extensión directa de cómo lo hacen las famosas cartas de control univariadas de Shewhart. Véase a continuación la gráfica de control individual sobre uno de los scores encontrados por PCA.

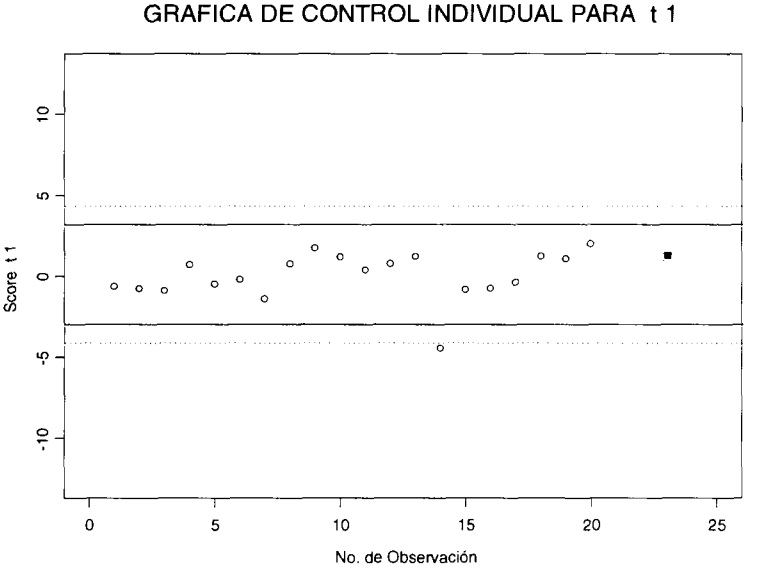


Figura 4

Si se ha determinado en una situación dada, que más de una dirección específica (componente principal) es importante para explicar la mayor parte de la variabilidad de un proceso, se pueden manejar de forma separada, cartas de control individuales para dichos scores.

2.2 PLS

Partial Least Squares es una técnica de reciente surgimiento; sus primeras aplicaciones aparecen en la década de los 90. Es ésta, una técnica de predicción sobre variables que son direcciones latentes, similares a las componentes principales de PCA. El poder predictivo de PLS fue aprovechado en situaciones industriales donde se requería monitoreo de procesos pero sus mediciones de las variables de calidad o salida del proceso, eran difíciles de lograrse de forma rápida, anulando esto, totalmente la búsqueda de un sistema de control en línea. Fue en las aplicaciones de este tipo que la técnica comenzó a cobrar una relevancia marcada en cienrtos ámbitos de aplicación; uno de los cuales es la producción de la industria química y sus procesos tipo batches. Esto se comentará más adelante.

La regresión lineal simple es una técnica estadística muy conocida y de amplia aplicación. Se utiliza para formar modelos predicativos entre una pareja de variables que están relacionadas en forma lineal o mediante la regresión no lineal, si la correlación entre las variables no es proporcional. Se sabe que la regresión no lineal es una extensión casi directa de la lineal y, de igual forma las regresiones lineal y no lineal múltiples, con una variable respuesta *Y*, son de fácil entendimiento cuando se ha comprendido la regresión lineal simple básica.

Si bien es cierto que en muchas aplicaciones que se hacen de la regresión clásica de mínimos cuadrados, no se tiene mucho cuidado en la verificación del cumplimiento de los supuestos de normalidad y varianzas constantes, principalmente; existe un campo de aplicación grande en áreas químicas como espectrometría y calibración, donde es claro que el cumplimiento de los supuestos mencionados muy rara vez se da. Así, en dichas aplicaciones se han empleado tradicionalmente regresiones Ridge o algunas otras adaptaciones. Además siendo éste un campo fértil para las técnicas de reducción de variables, demandaba alguna técnica multivariada que pudiera servir a la vez para formar modelos predictivos.

PLS nace en este contexto como una variante de PCR (regresión de componentes principales) que ya se aplicaba para dichos fines. La regresión de componentes principales, aunque cumple con los dos requisitos mencionados, pues reduce dimensiones y puede formar modelos predictivos, no probó ser tan eficiente en este campo de aplicación como PLS. Las direcciones de las variables latentes en PLS no son las direcciones de mayor variabilidad sino las direcciones más relacionadas entre las variables respuestas y explicativas.

Por otra parte, en regresión multivariada, el método más simple sería ajustar cada variable *Y* independientemente sobre el conjunto de regresoras, aunque esto podría resultar en conjuntos diferentes de variables independientes para cada regresión. PLS ofrece un tratamiento más sofisticado de este problema.

La regresión será efectiva sólo si la variable Y está correlacionada con al menos una de las variables X. Es importante conservar el número de variables X usadas en regresión, tan pequeño como sea posible. Para este fin ayuda que

las variables X no estén correlacionadas entre sí. PLS es una técnica de regresión que logra estas dos propiedades.

En PCA se busca el espacio de vectores latentes que explique la mayor cantidad de variabilidad de una sola matriz de datos ($\mathbf{X}$). Con frecuencia, también en el control de procesos industriales, podemos identificar un segundo grupo de variables ($\mathbf{Y}$), las cuales son de gran importancia (variables de calidad o productividad) y que nos gustaría incluir en un esquema de monitoreo del desempeño del proceso. Desafortunadamente estas variables son, por lo general, medidas con una frecuencia mucho menor que la de las variables del proceso ($\mathbf{X}$), así que sería deseable poder usar la información contenida en $\mathbf{X}$, para PREDECIR, MONITOREAR y DETECTAR cambios en las variables de salida ($\mathbf{Y}$).

Regresión lineal múltiple (MLR) es el método más común para desarrollar modelos estadísticos multivariados, pero es bien sabido que MLR puede tener problemas serios al tratar con conjuntos de datos que presentan problemas de colinealidad, conduciendo a parámetros de estimación muy imprecisos y/o a predicciones pobres. En muchas aplicaciones, subconjuntos de variables independientes son escogidos para eliminar problemas de este tipo, pero procediendo de esta forma se puede perder información contenida en las variables que no se tomaron en cuenta.

Un método que podría ser usado para estabilizar los coeficientes de regresión en estos casos, es el de Regresión Ridge [17], pero este método no reduce la dimensionalidad del problema (como lo hace PCA) y se vuelve en extremo problemático para casos con grandes dimensiones (muchas variables).

Regresión de Componentes Principales (PCR) es un método que efectúa la regresión de cada una de las $Y's$, sobre las componentes principales de la matriz $\mathbf{X}$; en esta forma, esta técnica resuelve tanto el problema de dimensión como el de colinealidad.

PCR trata cada variable y individualmente; cuando (como ocurre muchas veces) el espacio Y consiste de variables altamente correlacionadas (las variables y aportan poca o ninguna información individualmente, pero por el contrario, son altamente informativas como un grupo), esto puede conducir a resultados no concluyentes.

Además, el espacio definido por las componentes principales de $\mathbf{X}$, como ya se comentó, es solo el espacio que exhibe la mayor variación en las $X's$ y, no necesariamente el espacio que es más predictivo de $\mathbf{Y}$ [7].

El método de Proyección a Estructuras Latentes como también se conoce a PLS, parece manejar mejor los problemas anteriores. Conceptualmente, PLS es similar a PCA, excepto que reduce simultáneamente las dimensiones de los espacios X y Y para encontrar los vectores latentes, que están mas altamente correlacionados, en ambos espacios. De hecho, esta situación es manejada también por el clásico análisis de correlación canónica, pero la diferencia se da porque partial least squres intrínsecamente determina el número de direcciones latentes adecuadas como parte inherente de la misma técnica.

Los espacios de vectores latentes en PLS son encontrados usualmente, por un método iterativo, pero se puede mostrar que los vectores de carga en PLS, son los eigenvectores de $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$, donde $(\mathbf{Y}^T\mathbf{X})$ es la matriz de varianzas y covarianzas entre $\mathbf{X}$ y $\mathbf{Y}$ (Así como en PCA, los pesos o cargas están dados por los eigenvectores de $\mathbf{X}^T\mathbf{X}$).

El algoritmo de cálculo de PLS, es una adecuación del algoritmo NIPALS (para encontrar eigenvalores y eigenvectores), el cual se presenta a continuación, para un buen entendimiento de PLS. El nombre NIPALS proviene de Non-linear Iterative Partial Least Squares.

NIPALS no calcula todas las componentes principales a la vez, sino que calcula $\mathbf{t}_1$ y $\mathbf{p}_1{}^T$ (los scores y los pesos) desde la matriz $\mathbf{X}$; entonces el producto externo de $\mathbf{t}_1\,\mathbf{p}_1{}^T$ es sustraído de $\mathbf{X}$ y el residual $\mathbf{E}_1$ es calculado (ver enseguida); este residual es usado para calcular $\mathbf{t}_2$ y $\mathbf{p}_2{}^T$:

$$\mathbf{E}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1{}^T$$
$$\mathbf{E}_2 = \mathbf{X} - \mathbf{t}_2\mathbf{p}_2{}^T$$
$$\vdots$$
$$\mathbf{E}_A = \mathbf{X} - \mathbf{t}_A\mathbf{p}_A{}^T$$

Algoritmo NIPALS para PCA:

(1) tome el vector $\mathbf{x}_j$ de $\mathbf{X}$ y llame a este $\mathbf{t}_a$; $\mathbf{t}_a = \mathbf{x}_j$

(2) calcule $\mathbf{p}^T{}_a$: $\mathbf{p}^T{}_a = \dfrac{\mathbf{t}^T{}_a\mathbf{X}}{\mathbf{t}^T{}_a\mathbf{t}_a}$

(3) normalice $\mathbf{p}^T{}_a$ a tener longitud 1: $\mathbf{p}^T{}_a = \dfrac{\mathbf{p}^T{}_a}{\left\|\mathbf{p}^T{}_a\right\|}$

(4) calcule $\mathbf{t}_a$: $\mathbf{t}_a = \dfrac{\mathbf{X}\mathbf{p}_a}{\mathbf{p}^T{}_a\mathbf{p}_a}$

(5) compare el $\mathbf{t}_a$ usado en 2, con aquel obtenido en 4. Si son iguales, parar (ya que la iteración ha convergido); si difieren, regrese al paso 2

Después de que se encuentra la primera componente (a), se reemplaza $\mathbf{X}$ en los pasos 2 y 4 por su residual $(\mathbf{X} - \mathbf{t}_1\mathbf{p}^{T}_1)$

*note que en el paso 4, se está dividiendo entre la norma cuadrada de un vector unitario. Esto es inecesario; se incluye aquí porque asi aprece en la nomenclatura tradicional de dicho algoritmo. Lo mismo ocurre en el algoritmo PLS mostrado más adelante.

2.2.1 Construyendo el modelo PLS

Un modelo sencillo consistiría en la aplicación de PCA tanto en la matriz $\mathbf{X}$, como en la matriz $\mathbf{Y}$, para obtener:

$$\mathbf{X} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^{T} + \mathbf{E} = \mathbf{T}\mathbf{P}^{T} + \mathbf{E}$$

$$\mathbf{Y} = \sum_{a=1}^{A} \mathbf{u}_a \mathbf{q}_a^{T} + \mathbf{F} = \mathbf{U}\mathbf{Q}^{T} + \mathbf{F}$$

y entonces realizar una regresión entre los scores de $\mathbf{X}$ y de $\mathbf{Y}$ ($\mathbf{t}$ y $\mathbf{u}$)

$$\hat{\mathbf{u}}_a = \hat{b}_a \mathbf{t}_a$$

(en su forma lineal)

donde $\hat{b}_a = \dfrac{\mathbf{u}_a^{T}\mathbf{t}_a^{T}}{\mathbf{t}_a^{T}\mathbf{t}_a}$

aquí, los $\hat{b}_a$ juegan el papel de los coeficientes de regresión en los modelos MLR y PCR. Este modelo, sin embargo, no es el mejor posible. La razón es que las componentes principales fueron calculadas para ambos bloques en forma separada, de tal forma que ellos tienen una relación débil entre sí; sería mejor dar a cada bloque información del otro, de tal forma que componentes ligeramente rotados, resultaran en un mejor ajuste al momento de la regresión. Escribiendo los dos modelos en forma algorítmica, podemos ver la forma en que podrían obtener información uno del otro:

PARA EL BLOQUE X	PARA EL BLOQUE Y
(1) tome $\mathbf{t}$ = algún x_j	tome $\mathbf{u}$ = algún y_j

(2) $\mathbf{p}^T = \frac{\mathbf{t}^T\mathbf{X}}{\mathbf{t}^T\mathbf{t}}$ $\left(= \frac{\mathbf{u}^T\mathbf{X}}{\mathbf{u}^T\mathbf{u}}\right)^*$ $\mathbf{q}^T = \frac{\mathbf{u}^T\mathbf{X}}{\mathbf{u}^T\mathbf{u}}$ $\left(= \frac{\mathbf{t}^T\mathbf{X}}{\mathbf{t}^T\mathbf{t}}\right)^*$

(3) $\mathbf{p}^T = \frac{\mathbf{p}^T}{\|\mathbf{p}^T\|}$ $\mathbf{q}^T = \frac{\mathbf{q}^T}{\|\mathbf{q}^T\|}$

(4) $\mathbf{t} = \frac{\mathbf{Xp}}{\mathbf{p}^T\mathbf{p}}$ $\mathbf{u} = \frac{\mathbf{Yq}}{\mathbf{q}^T\mathbf{q}}$

(5) compare $\mathbf{t}$ en 2 y 4
si son iguales, detenga el proceso

compare $\mathbf{u}$ en 2 y 4
si son iguales, detenga el proceso

Mejorando la relación entre bloques

La forma en que pueden compartir información ambos bloques es dejando que $\mathbf{t}$ y $\mathbf{u}$ cambien posiciones en el paso 2* (ver los términos entre paréntesis). Entonces un solo algoritmo puede ser escrito en secuencia a partir de estos dos.

Este algoritmo usualmente converge muy rápido al tomar componentes rotados de los bloques $\mathbf{X}$ y $\mathbf{Y}$

Ortogonalizando los scores de X

(1) tome $\mathbf{u}$ = algún y_j

(2) $\mathbf{p}^T = \frac{\mathbf{u}^T\mathbf{X}}{\mathbf{u}^T\mathbf{u}}$ $\left(\mathbf{w} = \frac{\mathbf{u}^T\mathbf{X}}{\mathbf{u}^T\mathbf{u}}\right)^{**}$

(3) $\mathbf{p}^T = \frac{\mathbf{p}^T}{\|\mathbf{p}^T\|}$ $\left(\mathbf{w}^T = \frac{\mathbf{w}^T}{\|\mathbf{w}^T\|}\right)^{**}$

(4) $\mathbf{t} = \frac{\mathbf{Xp}}{\mathbf{p}^T\mathbf{p}}$ $\left(\mathbf{t} = \frac{\mathbf{Xw}}{\mathbf{w}^T\mathbf{w}}\right)^{**}$

(5) $\mathbf{q}^T = \frac{\mathbf{t}^T\mathbf{Y}}{\mathbf{t}^T\mathbf{t}}$

(6) $\mathbf{q}^T = \frac{\mathbf{q}^T}{\|\mathbf{q}^T\|}$

(7) $\mathbf{u} = \frac{\mathbf{Yq}}{\mathbf{q}^T\mathbf{q}}$

(8) compare $\mathbf{t}$ en 4 con la del paso precedente en la iteración; si son iguales (dentro de un cierto rango de error de redondeo), pare; en caso contrario, regrese al paso 2.

Hay aún un problema; el algoritmo no da valores $\mathbf{t}$ ortogonales (como lo hace PCA y que es clave para el monitoreo). La razón es, que el orden de cálculos que se usa en PCA ha sido cambiado; usemos entonces un vector de paso $\mathbf{w}^T$** en lugar de $\mathbf{p}^T$ (ver fórmulas entre paréntesis).
Se puede incluir un paso extra después de la convergencia, para obtener valores de $\mathbf{t}$, ortogonales:

$$\mathbf{p}^T = \frac{\mathbf{t}^T\mathbf{X}}{\mathbf{t}^T\mathbf{t}}$$

y, que al normalizar $\mathbf{p}^T = \frac{\mathbf{p}^T}{\|\mathbf{p}^T\|}$, requiere del ajuste de $\mathbf{t}$: $\mathbf{t}^T = \frac{\mathbf{Xp}}{\mathbf{p}^T\mathbf{p}}$

Pero esto es equivalente a realizar una sola multiplicación escalar con la norma de $\mathbf{p}^T$; esto es,

$$\mathbf{t} = \mathbf{t}\|\mathbf{p}^T\|$$

Por otro lado los $\mathbf{w}$'s serán usados en predicciones, por lo que deben ser también reescalados:

$$\mathbf{w} = \mathbf{w}\|\mathbf{p}^T\|$$

Ahora sí, $\mathbf{t}$ puede usarse en la regresión, y los residuales se calculan de:

$$\mathbf{E} = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1^T \qquad \text{y} \qquad \mathbf{F} = \mathbf{Y} - \mathbf{u}_1\mathbf{q}_1^T$$

En forma general:

$$\mathbf{E}_h = \mathbf{E}_{h-1} - \mathbf{t}_h \mathbf{p}_h^T; \quad \mathbf{X} = \mathbf{E}_0$$
$$\mathbf{F}_h = \mathbf{F}_{h-1} - \mathbf{u}_h \mathbf{q}_h^T; \quad \mathbf{Y} = \mathbf{F}_0$$

y, para encontrar la relación que será usada para predicción, $\mathbf{u}_h$ se reemplaza por su estimador

$\hat{\mathbf{u}}_h = \hat{b}_h \mathbf{t}_h$ y entonces:

$$\mathbf{F}_h = \mathbf{F}_{h-1} - b_h \mathbf{t}_h \mathbf{q}_h^T$$

y el propósito es hacer que $\|\mathbf{F}_h\|$ sea pequeña.

El algoritmo completo, en forma terminada, se presenta enseguida.

2.2.2 Algoritmo NIPALS para PLS

(1) tome $\mathbf{u}$ inicial = cualquier y_j

(2) $\mathbf{w} = \dfrac{\mathbf{u}^T \mathbf{X}}{\mathbf{u}^T \mathbf{u}}$

(3) $\mathbf{w}^T = \dfrac{\mathbf{w}^T}{\|\mathbf{w}^T\|}$ (normalización)

(4) $\mathbf{t} = \dfrac{\mathbf{X}\mathbf{w}}{\mathbf{w}^T \mathbf{w}}$

(5) $\mathbf{q}^T = \dfrac{\mathbf{t}^T \mathbf{Y}}{\mathbf{t}^T \mathbf{t}}$

(6) $\mathbf{q}^T = \dfrac{\mathbf{q}^T}{\|\mathbf{q}^T\|}$ (normalización)

(7) $\mathbf{u} = \dfrac{\mathbf{Y}\mathbf{q}}{\mathbf{q}^T \mathbf{q}}$

(8) Compare la convergencia de $\mathbf{t}$ del paso 4 con su valor previo; si son iguales continúe en el paso 9, si no es así, regrese al paso 2.

(9) $\mathbf{p}^T = \frac{\mathbf{t}^T\mathbf{X}}{\mathbf{t}^T\mathbf{t}}$

(10) $\mathbf{p}^T = \frac{\mathbf{p}^T}{\|\mathbf{p}^T\|}$ (normalización)

(11) $\mathbf{t} = \mathbf{t}\|\mathbf{p}^T\|$

(12) $\mathbf{w} = \mathbf{w}\|\mathbf{p}^T\|$

(13) $b = \frac{\mathbf{u}^T\mathbf{t}}{\mathbf{t}^T\mathbf{t}}$

Como punto final "actualice" las matrices $\mathbf{X}$ y $\mathbf{Y}$, restándoles el producto vectorial de $\mathbf{t}$ con $\mathbf{p}$, y regrese para calcular la siguiente componente.

Resumen del algoritmo PLS:

- Existen 2 relaciones externas en los bloques $\mathbf{X}$ y $\mathbf{Y}$, de la forma:

  $\mathbf{X} = \mathbf{TP}^T + \mathbf{E}$ y $\mathbf{Y} = \mathbf{UQ}^T + \mathbf{F}$

- Existe una relación interna:

  $\hat{\mathbf{u}}_h = \hat{b}_h \mathbf{t}_h$

- La relación mixta es: $\mathbf{Y} = \mathbf{TBQ}^T + \mathbf{F}$, donde $\|\mathbf{F}\|$ debe ser minimizada

- En el algoritmo iterativo, los bloques intercambian scores, lo cual da una mejor relación interna

- Con el fin de obtener los scores de $\mathbf{X}$ ($\mathbf{t}$) ortogonales, es necesario introducir los pesos ($\mathbf{w}$).

Propiedades de los factores PLS

Las principales propiedades de los términos envueltos en PLS son las siguientes:

- Las cantidades $\mathbf{p}_a^T$ y $\mathbf{q}_a^T$ tienen longitud unitaria para cada a: $\left\|\mathbf{p}_a^T\right\| = 1$ y $\left\|\mathbf{q}_a^T\right\| = 1$

- $\mathbf{t}_a$ y $\mathbf{u}_a$ son centrados alrededor de cero: $\sum_{i=1}^{N} t_{ai} = \sum_{i=1}^{N} u_{ai} = 0$

- Los $\mathbf{w}_a$ son ortogonales: $\mathbf{w}_i^T \mathbf{w}_i = \partial_{ij} \left\|\mathbf{w}_i^T\right\|^2$

- Los $\mathbf{t}_a$ son ortogonales

Predicción

La parte importante de cualquier regresión es su uso para predecir el bloque dependiente a partir del bloque independiente. Esto se logra en PLS descomponiendo el bloque $\mathbf{X}$ y construyendo el bloque $\mathbf{Y}$. Para este propósito, $\mathbf{p}^T$, $\mathbf{q}^T$ y $\mathbf{w}^T$ y b, de la parte de la construcción del modelo, son guardados para cada factor PLS (cada dimensión a).

Para el bloque $\mathbf{X}$, $\mathbf{t}$ se estima multiplicando $\mathbf{X}$ por $\mathbf{w}$, como en la etapa de construcción del modelo:

$$\hat{\mathbf{t}}_a = \mathbf{E}_{a-1}\mathbf{w}_a$$

$$\mathbf{E}_a = \mathbf{E}_{a-1} - \hat{\mathbf{t}}_a \mathbf{p}_a^T$$

con $\mathbf{X} = \mathbf{E}_0$

Para el bloque $\mathbf{Y}$:

$$\mathbf{Y} = \sum_{a=1}^{A} b_a \hat{\mathbf{t}}_a \mathbf{q}_a^T$$

La suma es sobre todas las dimensiones que se quieran incluir en el PLS.

Si el modelo intrínseco para la relación entre $\mathbf{X}$ y $\mathbf{Y}$ es un modelo lineal. El número de componentes necesario para describir este modelo es igual a la dimensionalidad del modelo. Modelos no lineales requieren componentes extras para describir la no linealidad.

2.2.3 Número de componentes en PLS

El número de componentes a ser usado es una propiedad muy importante de un modelo PLS.

Aunque es posible calcular tantas componentes PLS como el rango de la matriz $\mathbf{X}$, normalmente no todas son usadas. La razón principal de esto, es que los datos medidos nunca están libres de error y algunas de las más pequeñas componentes, solo describen "ruido" y, como ya hemos dicho, es deseable eliminar estas pequeñas componentes porque ellas conllevan los problemas de multicolinealidad. Además de que se muestra la potencia de la reducción de datos, facilitando con esto la interpretación de la situación estudiada.

Significa entonces, que debe haber uno o varios criterios para decidir cuando parar. Un posible criterio es que $\|\mathbf{F}_a\|$ sea pequeña (ver sección 2.2.1). La figura 5 muestra una gráfica de $\|\mathbf{F}_a\|$ vs. el número de componentes.

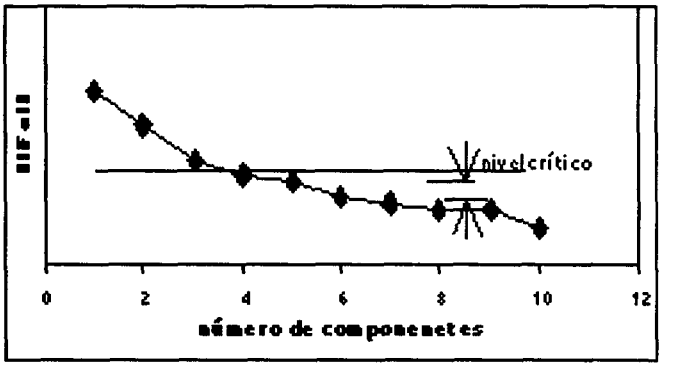


Figura 5

Es posible escoger un nivel crítico y parar cuando $\|\mathbf{F}_a\|$ "caiga" por debajo de este nivel. Otra posibilidad, es tomar la diferencia entre $\|\mathbf{F}_a\|$ y $\|\mathbf{F}_{a-1}\|$, y parar cuando éste es pequeño comparado a algún error de medición previamente establecido.

Se puede usar también el análisis de varianza (donde la prueba F es sobre la relación interna del modelo) para validar el modelo. En este caso, se usa la prueba F en la regresión lineal como un criterio para saber cuando detenerse en el número de dimensiones a considerar.

Los métodos mencionados son adecuados para la etapa de la construcción del modelo en PLS. Si se desea hacer predicción, otra clase de criterios son más adecuados para establecer el número de componentes necesarias. "crossvalidation" es un procedimiento para probar la *consistencia interna de un*

modelo de regresión, cuando este se está construyendo y se lleva a cabo de la siguiente forma:

Una fracción del conjunto de datos (observaciones m-dimensionales) son omitidas en la construcción del modelo; esta fracción puede variar desde un único objeto (boostrap) hasta la mitad de los objetos

$\mathbf{X}_m$ <----> $\mathbf{Y}_m$
$\mathbf{X}_t$ -------> $\mathbf{Y}_t$

Las etapas sucesivas son:

1) El modelo es construido entre $\mathbf{X}_m$ y $\mathbf{Y}_m$
2) Con los parámetros del modelo, $\mathbf{Y}_t$ es predicha a partir de $\mathbf{X}_t$
3) Se calcula la suma de cuadrados $\mathbf{ss_a}$ de $\left(Yt - \hat{Y}t\right)^2$ para cada dimensión
4) $\boldsymbol{ss_a}$ es acumulado en $\mathbf{SS_a}$ para cada dimensión a
5) La fracción que fue omitida es reinstalada y otra fracción del mismo tamaño es omitida
6) Se vuelve al paso 1

Se repite este procedimiento, hasta que todos los objetos han estado en $\mathbf{X_t}$ una vez. Finalmente $\mathbf{SS_a}$ es dividido por el tamaño de la matriz **Y**; esto resulta en **PRESSa** (suma de cuadrados de los residuales en la predicción). Se puede graficar **PRESSa** contra el número de dimensiones, para escoger el que produzca el **PRESS** mínimo (ver Figura 6) o bien podemos comparar la raíz cuadrada de **PRESS** con la desviación estándar del bloque **Y** de la dimensión previa, como sigue:

$$\mathbf{XVAL} = \frac{(\mathrm{PRESS})^{1/2}}{S_Y}$$

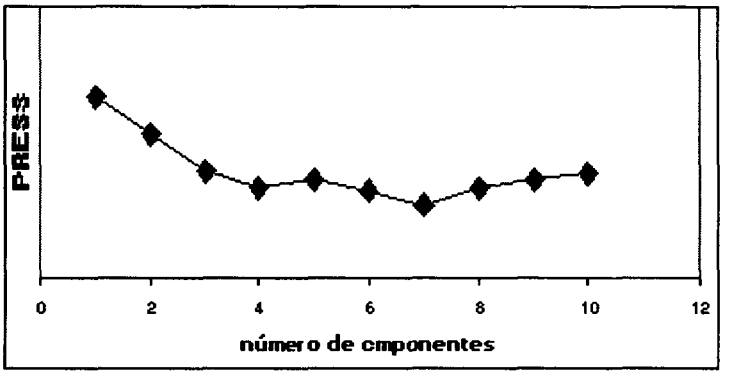


Figura 6

Si **XVAL**$\geq$1, la dimensión actual no tiene propiedades predictivas.

Si XVAL<1, la dimensión actual es útil y la siguiente debe probarse.

2.2.4 Fundamento matemático de PLS

Sean $\mathbf{X}$ y $\mathbf{Y}$ las matrices de N datos, M y P-dimensionales, respectivamente:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix} \qquad \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1P} \\ y_{21} & y_{22} & \cdots & y_{2P} \\ \vdots & \vdots & \vdots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NP} \end{bmatrix}$$

Sea $\mathbf{w}$ un vector arbitrario unitario en el M-espacio. Llamemos entonces $\mathbf{t}$ al vector de proyección de $\mathbf{X}$ sobre $\mathbf{w}$. Esto es, $\mathbf{t} = \mathbf{X}\,\mathbf{w}$.

De forma análoga, sea $\mathbf{c}$ un vector arbitrario unitario en el P-espacio. Llamemos entonces $\mathbf{u}$ al vector de proyección de $\mathbf{Y}$ sobre $\mathbf{c}$. Esto es, $\mathbf{u} = \mathbf{Y}\,\mathbf{c}$

Si en este momento fuéramos a buscar los ejes de componentes principales de $\mathbf{X}$, sólo tendríamos que forzar al vector $\mathbf{w}$ a ser aquel que minimice los residuos (como vimos para PCA), o equivalentemente (como puede probarse sin mucha dificultad) a ser aquel que maximice la longitud de $\mathbf{t}$ (es decir, que tenga la proyección máxima). Lo mismo haríamos con $\mathbf{Y}$, $\mathbf{u}$ y $\mathbf{c}$. De hecho, esto es lo que hace la técnica de PCR: obtiene las componentes principales de $\mathbf{X}$ y de $\mathbf{Y}$ por separado y procede con la regresión de las componentes principales de $\mathbf{Y}$ sobre las de $\mathbf{X}$. En PLS, sin embargo hay una variante: se buscan $\mathbf{w}$ y $\mathbf{c}$ tales que el producto interno entre $\mathbf{t}$ y $\mathbf{u}$ sea máximo para asegurar que las componentes que se obtienen de $\mathbf{X}$ y $\mathbf{Y}$ estén lo más correlacionadas que sea posible.

Sabemos que el producto interno de $\mathbf{t}$ y $\mathbf{u}$, está dado por: $\mathbf{t}^T\mathbf{u}$ o, en forma de sumatoria: $\sum_{i=1}^{N} t_i u_i$. De manera que esta es nuestra función a maximizar, sujeta a las restricciones de que $\mathbf{w}$ y $\mathbf{c}$ sean unitarios.
Utilizando nuevamente multiplicadores de Lagrange:

$$\sum_{i=1}^{N} t_i u_i - \lambda_1\left(\sum_{j=1}^{N} w_j{}^2 - 1\right) - \lambda_2\left(\sum_{k=1}^{P} c_k{}^2 - 1\right) \tag{5}$$

sustituyendo $\mathbf{t}$ y $\mathbf{u}$ como las proyecciones de $\mathbf{X}$ y $\mathbf{Y}$ sobre $\mathbf{w}$ y $\mathbf{c}$,

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{M} x_{ij} w_j\right)\left(\sum_{k=1}^{P} y_{ik} c_k\right) - \lambda_1\left(\sum_{j=1}^{N} w_j{}^2 - 1\right) - \lambda_2\left(\sum_{k=1}^{P} c_k{}^2 - 1\right)$$

que reordenando,

$$\sum_{j=1}^{M}\sum_{k=1}^{P} w_j c_k \left(\sum_{i=1}^{N} x_{ij} y_{ik}\right) - \lambda_1\left(\sum_{j=1}^{N} w_j{}^2 - 1\right) - \lambda_2\left(\sum_{k=1}^{P} c_k{}^2 - 1\right)$$

el término $\sum_{i=1}^{N} x_{ij} y_{ik}$ es el elemento jk de la matriz $\mathbf{X}^{\mathrm{T}}\mathbf{Y}$. Si llamamos a este término, $(X'Y)_{jk}$, tendremos: la función objetivo:

$$\sum_{j=1}^{M}\sum_{k=1}^{P} (X'Y)_{jk} w_j c_k - \lambda_1\left(\sum_{j=1}^{N} w_j{}^2 - 1\right) - \lambda_2\left(\sum_{k=1}^{P} c_k{}^2 - 1\right)$$

Derivando esta función con respecto a w_m primero,

$\sum_{k=1}^{P}(X'Y)_{mk} c_k - \lambda_1(2w_m)$, igualando a cero y despejando

$$\sum_{k=1}^{P}(X'Y)_{mk} c_k = 2\lambda_1 w_m \qquad (6)$$

Siguiendo el mismo proceso al derivar con respecto a c_p,

$$\sum_{j=1}^{M}(X'Y)_{jp} w_j - \lambda_2(2c_p) \rightarrow \sum_{j=1}^{M}(X'Y)_{jp} w_j = 2\lambda_2 c_p \qquad (7)$$

multiplicamos (6) por w_m : $w_m \sum_{k=1}^{P}(X'Y)_{mk} c_k = 2\lambda_1 w_m{}^2$ y aplicamos sumatoria sobre w,

$\sum_{m=1}^{M} w_m \sum_{k=1}^{P}(X'Y)_{mk} c_k = 2\lambda_1 \sum_{m=1}^{M} w_m{}^2 = 2\lambda_1$.

Similarmente sobre (7), $\sum_{p=1}^{P} c_p \sum_{j=1}^{M}(X'Y)_{jp} w_j = 2\lambda_2 \sum_{p=1}^{P} c_p{}^2 = 2\lambda_2$

los lados izquierdos de las últimas dos ecuaciones son iguales e iguales a $\mathbf{t}^T\mathbf{u}$, entonces,

$\mathbf{t}^T\mathbf{u} = 2\lambda_1 = 2\lambda_2$

Si sustituimos w de (6) en (7), $\sum_{j=1}^{M}(X'Y)_{jp}\left[(2\lambda_1)^{-1}\sum_{k=1}^{P}(X'Y)_{jk} c_k\right] = 2\lambda_2 c_p$, que reordenando,

$$\sum_{k=1}^{P}\sum_{j=1}^{M}(X'Y)_{jp}(X'Y)_{jk}c_k = (2\lambda_1)2\lambda_2 c_p = (\mathbf{t}^T\mathbf{u})^2 c_p \qquad (8)$$

donde reconocemos al término $\sum_{j=1}^{M}(X'Y)_{jp}(X'Y)_{jk}$ como el elemento pk de la matriz $(\mathbf{X}^T\mathbf{Y})^T(\mathbf{X}^T\mathbf{Y})$, por lo que (8) indica que la fila p de la matriz $(\mathbf{X}^T\mathbf{Y})^T(\mathbf{X}^T\mathbf{Y})$ al multiplicarse por el vector $\mathbf{c}$, resulta en un múltiplo de la componente p del mismo vector $\mathbf{c}$. Como esto vale para todos los componentes de $\mathbf{c}$, lo podemos escribir como:

$$(\mathbf{X}^T\mathbf{Y})^T(\mathbf{X}^T\mathbf{Y})\mathbf{c} = (\mathbf{t}^T\mathbf{u})^2\,\mathbf{c}$$

Lo cual implica que el vector $\mathbf{c}$ es el eigenvector de la matriz $[(\mathbf{X}^T\mathbf{Y})^T(\mathbf{X}^T\mathbf{Y})]$ correspondiente al eigenvalor $(\mathbf{t}^T\mathbf{u})^2$, que además debe ser el mayor eigenvalor por que justamente es esto lo que queremos maximizar.

Por un proceso similar se puede encontrar que $\mathbf{w}$ es el eigenvector de la matriz $[(\mathbf{Y}^T\mathbf{X})^T(\mathbf{Y}^T\mathbf{X})]$ correspondiente al mayor eigenvalor. Si sólo se quiere obtener $\mathbf{w}$, ya conociendo $\mathbf{c}$, no es necesario volver a calcular otros eigenvalores ya que de (6) y (7) se obtiene: $\mathbf{w}$ = (eigenvalor)$^{-1/2}$ $(\mathbf{X}^T\mathbf{Y})\mathbf{c}$.

Por medio de este proceso estamos encontrando las características de los vectores donde se proyectarán la matriz $\mathbf{X}$ y la matriz $\mathbf{Y}$. Estos vectores serán respectivamente $\mathbf{t}$ y $\mathbf{u}$. Los vectores $\mathbf{w}$ y $\mathbf{c}$ se utilizaron solo para encontrar las direcciones con correlación mayor en los conjuntos de datos. Así que, ahora nuevamente modelamos a $\mathbf{X}$ con la multiplicación de 2 vectores $\mathbf{t}$ y, $\mathbf{p}$ que no hemos encontrado aún, pero que igual que en PCA está dado por $\frac{\mathbf{X}^T\mathbf{t}}{\|\mathbf{t}\|^2}$. De igual forma procede en el caso de $\mathbf{Y}$, y tenemos entonces:

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \ldots + \mathbf{t}_A\mathbf{p}_A^T \qquad (9)$$

$$\mathbf{Y} = \mathbf{u}_1\mathbf{q}_1^T + \mathbf{u}_2\mathbf{q}_2^T + \ldots + \mathbf{u}_A\mathbf{q}_A^T \qquad (10)$$

Para el propósito de hacer predicciones no nos sirve esta representación de $\mathbf{Y}$ porque el vector $\mathbf{u}$ no se conoce para nuevos valores de $\mathbf{X}$ (porque necesitaríamos los valores de $\mathbf{Y}$ y si las tuviéramos, las predicciones carecerían de sentido). Podemos cambiar a $\mathbf{u}$ por $\mathbf{t}$ en la ecuación (10); esta era la finalidad perseguida al forzar a que $\mathbf{t}$ y $\mathbf{u}$ estuvieran lo más correlacionados posible. Entonces,

$$\hat{\mathbf{Y}} = \mathbf{t}_1\mathbf{r}_1^{\mathrm{T}} + \mathbf{t}_2\mathbf{r}_2^{\mathrm{T}} + \ldots + \mathbf{t}_A\mathbf{r}_A^{\mathrm{T}} \qquad (11)$$

donde $\mathbf{r}$ está dado por: $\dfrac{\mathbf{Y}^{\mathrm{T}}\mathbf{t}}{\|\mathbf{t}\|^2}$

2.2.5 Algoritmos de operación de PLS

(1) Comienzo: Establezca $\mathbf{u}$ igual a una columna de $\mathbf{Y}$

(2) $\mathbf{w}^{\mathrm{T}} = \dfrac{\mathbf{u}^{\mathrm{T}}\mathbf{X}}{\mathbf{u}^{\mathrm{T}}\mathbf{u}}$

(3) $\mathbf{w}^{\mathrm{T}} = \dfrac{\mathbf{w}}{\|\mathbf{w}^{\mathrm{T}}\|}$ normalizar $\mathbf{w}$ a longitud unitaria

(4) $\mathbf{t} = \dfrac{\mathbf{X}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{w}}$

(5) $\mathbf{q}^{\mathrm{T}} = \dfrac{\mathbf{t}^{\mathrm{T}}\mathbf{Y}}{\mathbf{t}^{\mathrm{T}}\mathbf{t}}$

(6) $\mathbf{q} = \dfrac{\mathbf{q}}{\|\mathbf{q}\|}$ normalizar $\mathbf{q}$ a longitud unitaria

(7) $\mathbf{u} = \dfrac{\mathbf{Y}\mathbf{q}}{\mathbf{q}^{\mathrm{T}}\mathbf{q}}$

(8) Checar la convergencia; si converge ir a 9, si no converge, regresar a 2

(9) $\mathbf{p} = \dfrac{\mathbf{X}^{\mathrm{T}}\mathbf{t}}{\mathbf{t}^{\mathrm{T}}\mathbf{t}}$

(10) $b = \dfrac{\mathbf{u}^{\mathrm{T}}\mathbf{t}}{\mathbf{t}^{\mathrm{T}}\mathbf{t}}$

(11) $\mathbf{E} = \mathbf{X} - \mathbf{t}\mathbf{p}^{\mathrm{T}}$; $\mathbf{F} = \mathbf{Y} - \mathbf{u}\mathbf{q}^{\mathrm{T}}$

(12) $\mathbf{X} = \mathbf{E}$; $\mathbf{Y} = \mathbf{F}$ y volver a 2.

*The Canadian Journal of Chemical engineering, Vol. 69, February, 1991.

2.2.6 Aplicaciones de PLS en la construcción de gráficos de control

Existen muchas situaciones (sobre todo en la industria química) en las que la medición de las variables de calidad de un proceso no se puede tener de forma inmediata al terminar la producción del mismo. Todos los productos en los cuales intervienen reacciones químicas o aquellos cuyas variables importantes son concentraciones de alguna sustancia, medicines de pH, etc., requieren para su determinación de un análisis que con frecuencia no se lleva a cabo en la misma planta, o en el mejor de los casos, demanda una cantidad de tiempo para su medición que hace prohibitiva la implementación del control de procesos estándar en las líneas de producción. En estos casos se puede aprovechar el poder predictivo de PLS para hacer monitoreo en línea sobre las estimaciones de dichas cantidades utilizando las variables de entrada ($X's$) a través de sus scores y del modelo PLS generado con conjuntos de datos de prueba, tomados en condiciones estables del proceso, de la misma forma que se acostumbra en las gráficas de control típicas de Shewhart. Aquí, tenemos la ventaja de que se pueden aplicar gráficas separadas por cada variable (dimensión o score) ya que tenemos la certeza de que dichas gráficas son independientes ya que en el modelo PLS, de la misma forma que PCA, genera componentes y scores ortogonales.

Además de llevar el monitoreo individual como ya se mencionó, se pueden monitorear también elipses de control en la que se incluyen parejas de scores como en la siguiente (observe el score de una nueva observación en el cuadrado sólido):

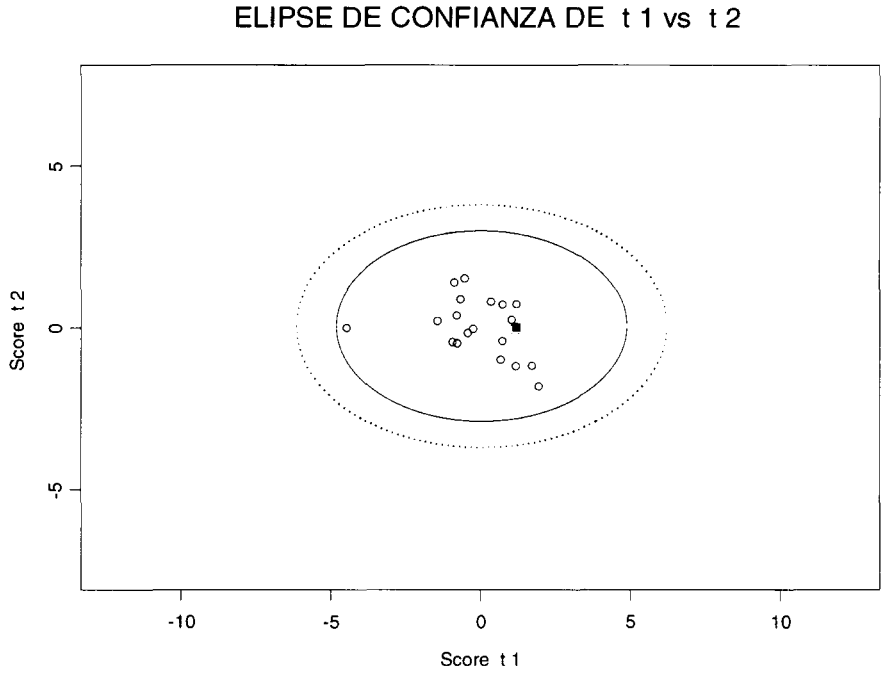


Figura 7

aunado a esto, se puede llevar un registro que puede ser visual o analítico de las predicciones mencionadas para estas nuevas observaciones. La rutina de gráficas de control que aparece en el anexo, tiene salidas como la siguiente para un conjunto de predicciones:

```
              Y real Y predicho
Observacion 1    162  143.29081
Observacion 2    110  124.72690
Observacion 3    101  111.08624
Observacion 4    105  141.24771



                   L.INF. REAL LIM.SUP.
Observación 1   31.7658358  162 254.8158
Observación 2   15.0009813  110 234.4528
Observación 3   -0.3324171  101 222.5049
Observación 4   27.7335239  105 254.7619
PORCENTAJE DE INTERVALOS DE CONFIANZA QUE NO CONTIENEN A SU VALOR REAL:
                                                                      0


              L.INF. PREDICHO LIM.SUP.
Observación 1 41.4224 158.6654 275.9085
```

En donde, fácilmente, se podría llevar un control del proceso en base a predicciones de $\mathbf{Y}$. Claro que la efectividad aquí depende también de la bondad del modelo y de la longitud del intervalo de predicción (el ejemplo mostrado, en donde parecería pobre la efectividad de la predicción, se logró con un conjunto base de veinte datos que provienen de un ejemplo tomado de [1]. Las variables que se miden son, para $\mathbf{Y}$: MENTÓN, INCORPORACION, SALTO (variables psicológicas). Las variables en $\mathbf{X}$, son: PESO, CINTURA Y PULSO (variables físicas). El ejemplo es famoso en la literatura por haber sido incorporado por el paquete computacional SAS en sus manuales, lo que lo hace fácilmente ubicable y por presentar en una cantidad pequeña de datos (que lo vuelve fácilmente interpretable) una estructura adecuada para evidenciar relacionaes multivariables. Los datos son autoría del Dr. A.C. Linnerud de la NCSU (North Carolina State University).

2.2.7 APLICACIONES PCA O PLS

Las técnicas descritas hasta el momento, han tenido una gran repercusión en los últimos años en que se han adaptado para funcionar en el monitoreo de control de procesos. PCA se utiliza en situaciones donde las variables de entrada de un proceso son muchas y están muy correlacionadas, mientras que la motivación principal para utilizar PLS es el hecho de tener muchas variables de calidad o de salida y que además son en extremo importantes. Mencionamos a continuación, unos ejemplos de la literatura para tener mejor idea sobre esto:

Reactor de cama fluidificado [10]

Proceso químico donde se producen metano, propano, etano, hidrógeno y butano. El esquema del reactor se presenta enseguida:

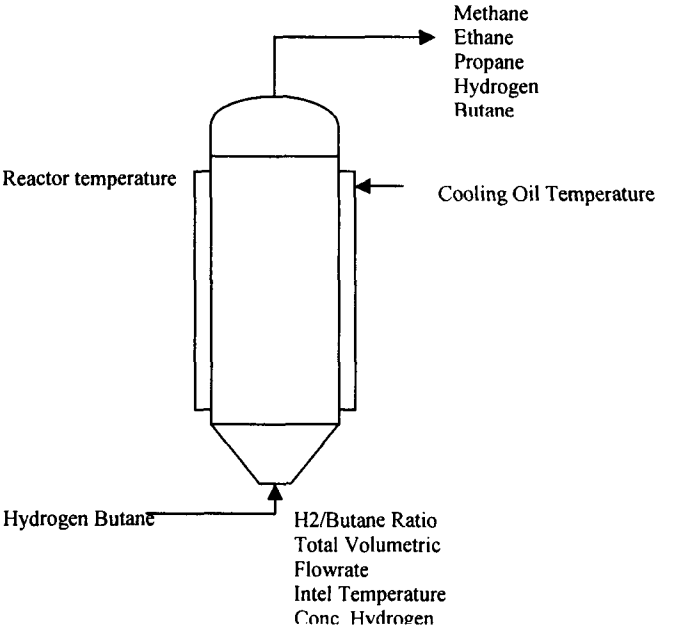


Figura 8

	First L.V.		Second L.V.	
Variables	Loading	Cum. % Expl.	Loading	Cum. % Expl.
Butane concentration	0.3002	29.5	-0.461	80.5
Hydrogen concentration	-0.2957	39.2	0.4696	81.36
Ratio	-0.3054	32.4	0.4951	88.34
Volumetric Flowrate	-0.1008	0	0.085	4.5
T-Inlet	0.1099	9	-0.1893	12
T-Cooling oil	0.586	56	0.4385	99.05
T-Reactor	0.6416	76.5	0.3131	98
Total % Explained of X		23.7		52.7
Methane selectivity	0.4607	83.7	0.4353	91
Ethane selectivity	-0.4046	65.3	-0.6862	82.6
Propane selectivity	-0.4594	83.2	-0.4127	89.8
Butane conversion	0.4471	78.74	0.255	81
Hydrogen conversion	0.4617	84	-0.3229	88
Total % Explained of Y		78.8		86.5
Regression coeficient (b)		1.265		0.4188

Results of the PLS Calculation for the Fluidized Bed Reactor Showing the Loadings for Each Latent Vector (LV) and the Cumulativa Percent of Variation Explained

Tabla 1

Ejemplos de aplicaciones de PCA aparecen en investigación de mercados en donde esta técnica se aplica con la finalidad de reducir datos de forma previa a un análisis de regresión o un análisis de clusters.

En economía, no son escasas las situaciones de un gran número de variables, tanto predictoras como respuesta, correlacionadas y que son sujeto de aplicación de PCA principalmente, aunque también se ha aplicado PLS en ellas.

2.2.8 MPLS

En la literatura de aplicaciones comienzan a aparecer situaciones que se modelan mediante una extensión de la técnica PLS, denominda MPLS (multiway partial least squres), o análogamente MPCA expandiendo PCA.

Estas modificaciones surgen, pensando en implementar las técnicas originales, en el caso de procesos que se desarrollan en "batches" o lotes. En este tipo de situaciones, es muy recomendable considerar a una misma variable medida a lo largo del tiempo que consume el batch, como si fueran diferentes variables, ya que sus valores son muy diferentes en los diferentes "instantes" en que se podría dividir teóricamente el proceso. Estas técnicas sugieren "desdoblar" a la matriz (si se aplica MPCA, o matrices si se aplica MPLS) de datos en varios instantes de tiempo (pueden ser cada cinco minutos, etc.), donde la partición depende mucho de las características específicas de cada caso, para posteriormente aplicar la técnica base, por ejemplo PLS a esta "gran matriz" de variables (el número de dimensiones M se multiplica por el número de intervalos de tiempo en el que se parte el proceso).

Siempre que se disponga de un sistema de toma de mediciones en línea es factible pensar en este tipo de aplicación. Sería muy difícil poder implementarse con una forma manual de toma de mediciones y registro. Afortundamente, la automatización a la que están tendiendo los procedimientos en la actualidad, mediante el control numérico y adaptación de equipo computacional directamente en las líneas de proceso, van habilitando a las líneas de producción, el poder ser monitoreados de esta forma.

En la literatura mencionada en la bibliografía, se pueden encontrar ejemplos de aplicaciones de este tipo. Ver [11] a [14].

La siguiente figura, tomada de [14] nos dará una idea de lo que se está planteando antes:

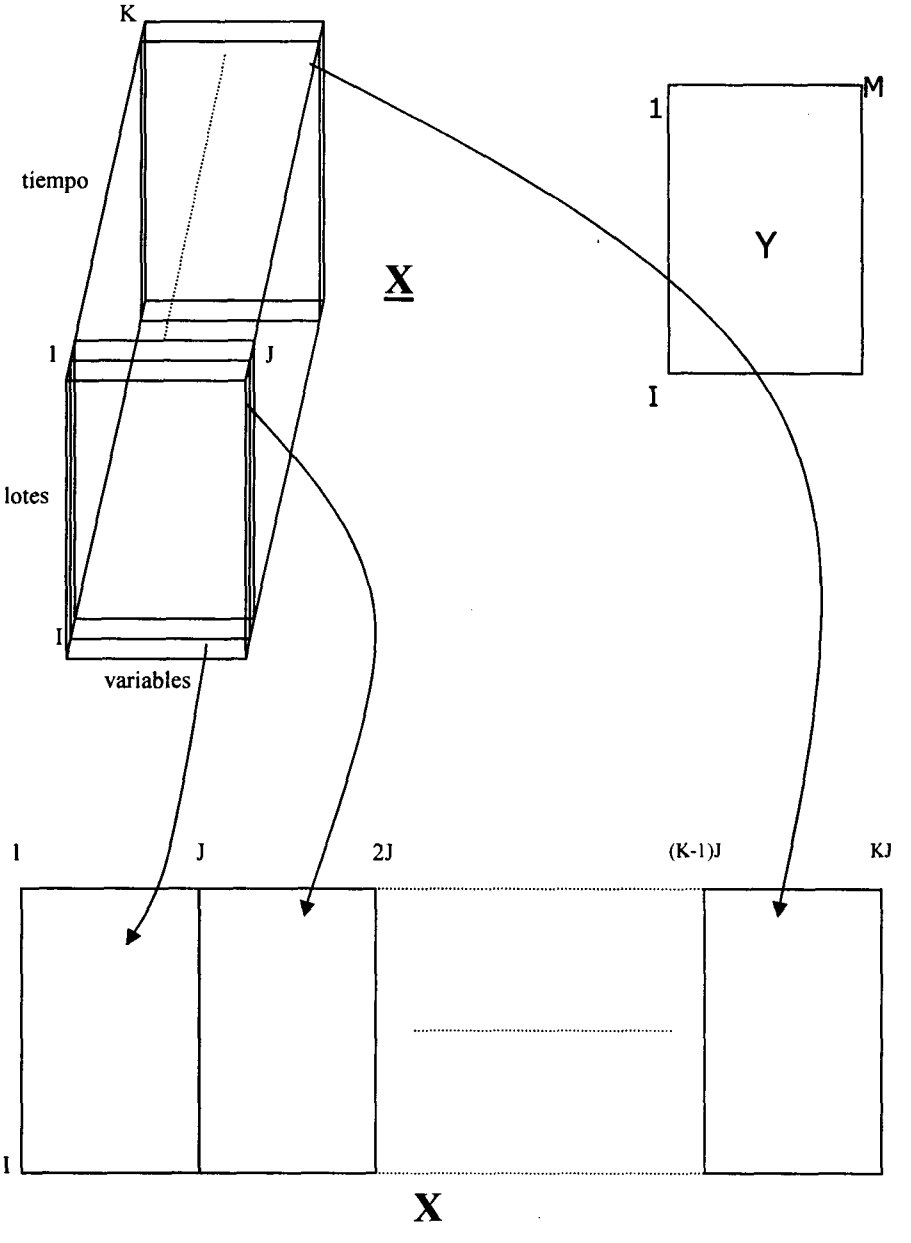


Figura 9

3. Análisis de resultados.

El proceso de componentes principales proyecta un conjunto de datos sobre unas nuevas direcciones llamadas direcciones latentes. Estas direcciones, para esta técnica, son aquellas donde los datos presentan una variabilidad mayor.

Esto se puede ver en la siguiente gráfica. (las proyecciones son las que aparecen en sólido y con forma de rombo)

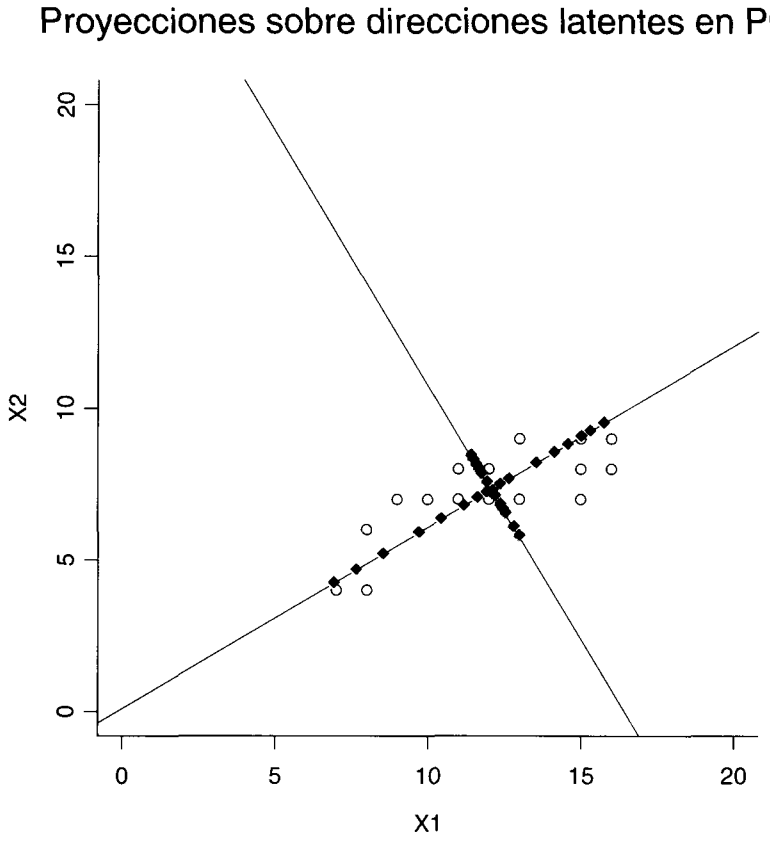


Figura 10

La rutina (en s-plus) que genera este resultado gráfico, así como el conjunto de datos utilizado, se incluyen en el apéndice. La rutina es PCA-GRAFICO y el conjunto de datos es DOSX.

El proceso denominado mínimos cuadrados parciales, proyecta también un conjunto de mediciones sobre unas nuevas direcciones. A diferencia de componentes principales, estas direcciones no son aquellas donde los datos tienen variabilidad mayor sino más bien aquellas en las que estas mediciones están mayormente correlacionadas con las direcciones latentes de otro conjunto de datos (las Y's). La gráfica siguiente muestra la comparación en las direcciones latentes de componentes principales y las direcciones de mínimos cuadrados parciales. Los ejes punteados (en los que no se muestran las proyecciones para tener mayor claridad en la figura) son las direcciones encontradas en el proceso de "partial least squares". El ángulo que "giran" los ejes $\mathbf{x_1}$-$\mathbf{x_2}$ originales en las componentes principales es de 47 grados por 43 grados de los ejes en mínimos cuadrados parciales.

Proyecciones PCA, y ejes PLS juntos para comparación

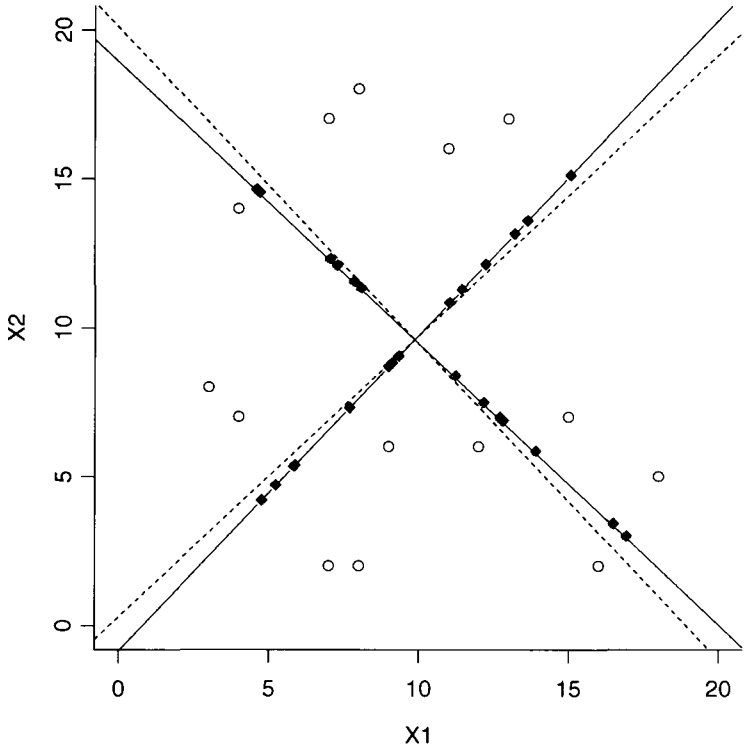


Figura 11

La rutina (en s-plus) que genera este resultado gráfico y el conjunto de datos, se incluyen en el apéndice. La rutina es PCA-PLS COMP GRAFICO y el conjunto de datos es CUATROX y CUATROY.

Como se puede ver aquí, tanto los ejes PCA como los PLS se pueden considerar como los ejes originales x_1-x_2; desplazados al centro de medias y rotados.

A continuación se presenta la salida de la rutina de criterios de detención (CRIT-DETEN).
El gráfico corresponde al criterio del "tamaño" de la matriz de residuales F en la dimensión A. La matriz de residuales cuando una única dimensión latente es considerada, tiene un "tamaño", medido como la suma de los valores absolutos, alrededor de 6.8 unidades, mientras que si se retienen dos dimensiones o variables latentes, este indicador baja a un valor de 6.6. Para tres dimensiones tenemos un valor de 6.4. Entre más pequeño sea este valor, se tendrá un error de predicción menor al uitilzar PLS, Para este caso, por ejemplo, valdría la pena quedarse sólo con la primera dirección latente ya que las otras dos no aportan mejora sustancial a las predicciones, o dicho de otro modo, la dimensionalidad esencial en el conjunto de datos LINNX y LINNY (ver apéndice) es uno.

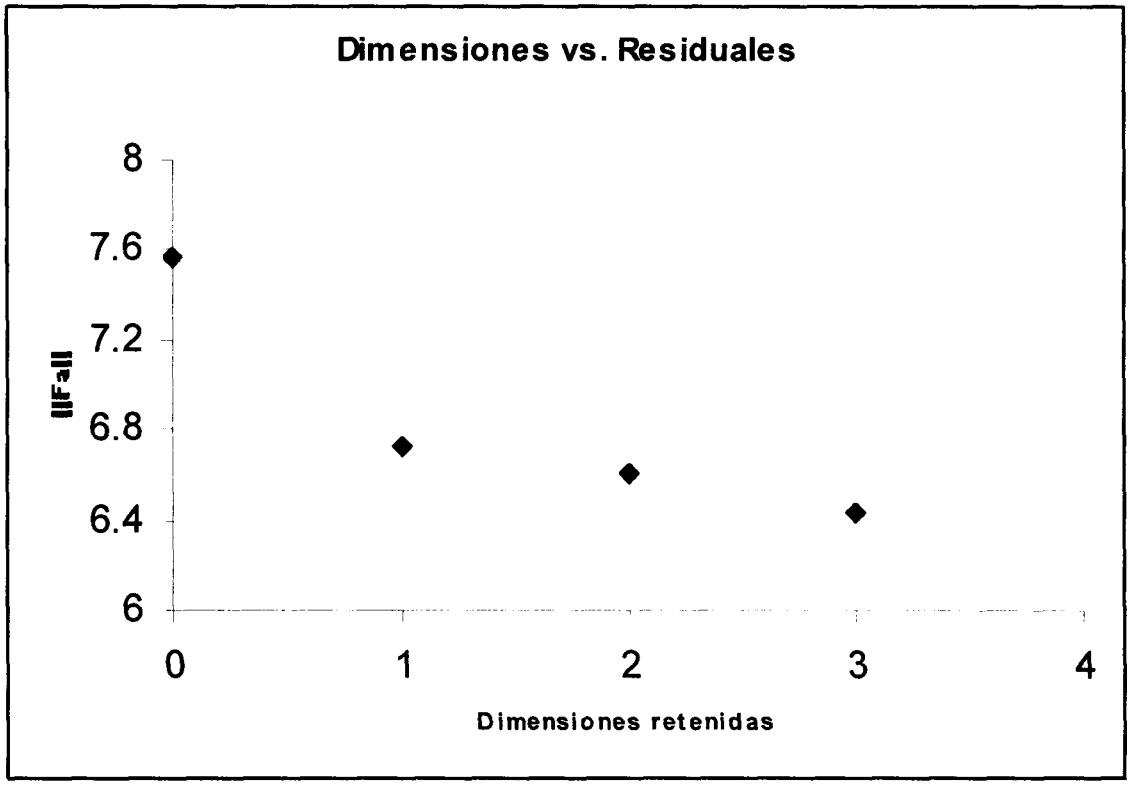


Figura 12

Recuerde que tenemos además de este criterio, los de PRESS y crossvalidation. Este es uno de los puntos críticos de PLS (como de PCA) y es motivo de conflicto entre diferentes preferencias.

El programa de eliminación de outliers (ELIMIN-OUT) tiene como salida, diagramas de dispersión de los scores calculados del conjunto de datos de prueba (para esta gráfica se utilizaron nuevamente las bases LINNX y LINNY). Al encontrar con PLS un modelo predictivo, es importante descartar observaciones atípicas antes de que vayan a sesgar las estimaciones de los parámetros del modelo. Estos "outliers" no pueden descartarse de las mediciones directas de **X** y **Y** sino más bien de los scores que son los que juegan el papel importante al cambiar el sistema de referencia de medición en los datos cuando se construyen las direcciones latentes. La rutina programada en s-plus aprovecha los gráficos interactivos de dicho paquete para identificar fácilmente a los datos mediante un clic sobre el punto "sospechoso". Aquí solo se está mostrando un diagrama de dispersión pero se deben revisar el diagrama correspondiente a cada pareja de scores (tres, en el caso de tres variables, utilizado aquí).

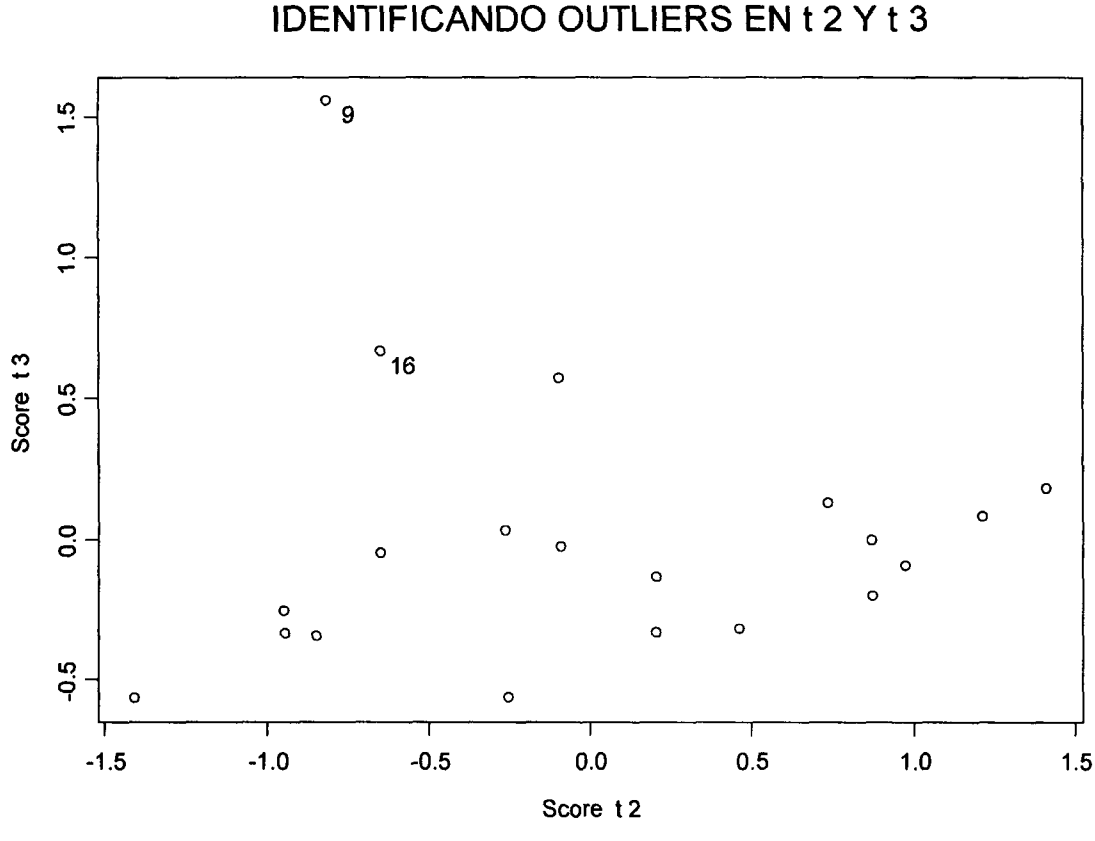


Figura 13

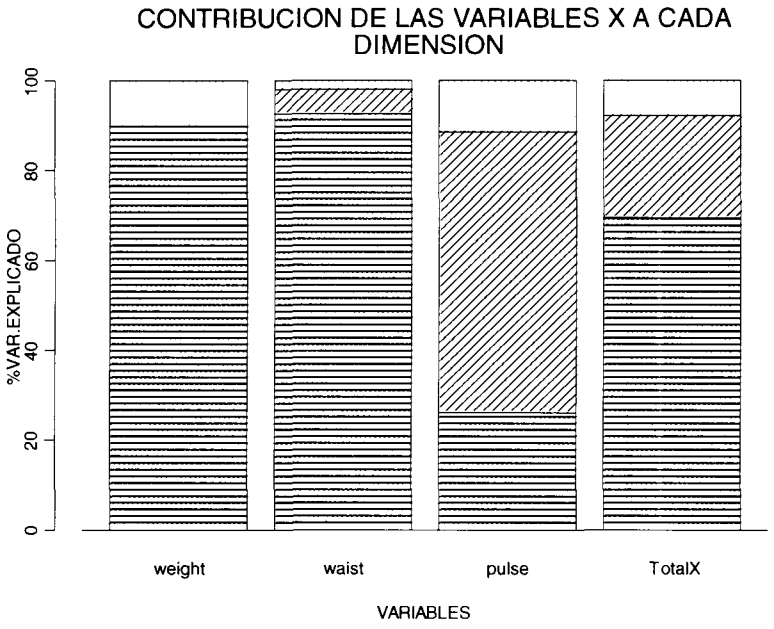


Figura 14

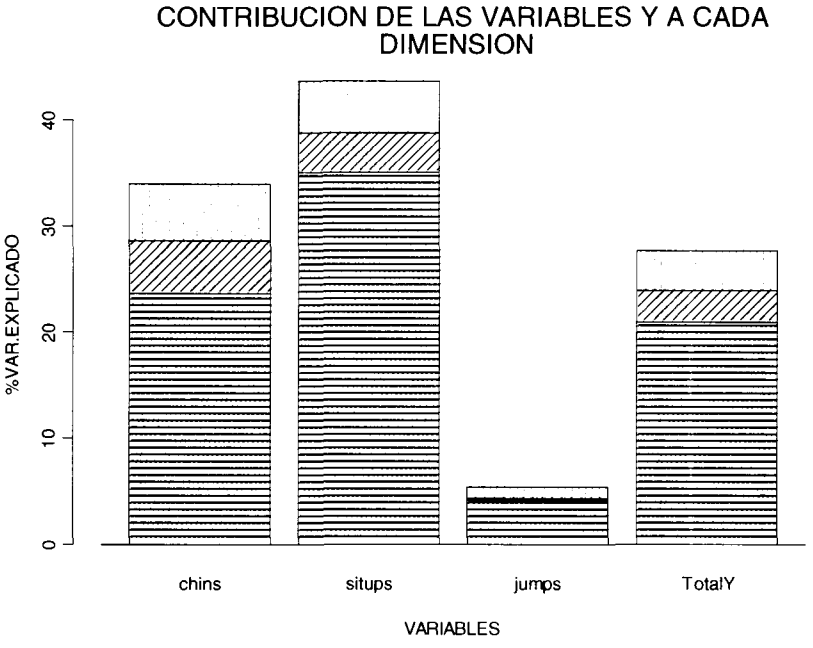


Figura 15

Las figuras 14 y 15 anteriores pueden considerarse una buena contribución de este trabajo en la interpretación de PLS, ya que en ellas se exhibe una representación que plasma de forma muy clara la participación de las variables **X** y **Y** a cada una de las dimensiones latentes en ambos espacios **X** y **Y** en R^3. Encontramos, por ejemplo, que todas las variables, tanto de X como de Y, contribuyen a la primera componente PLS (que se muestra con franjas horizontales), aunque las de mayor consideración son las variables *weight* y *waist* en **X**, y *jumps* en **Y** (cabe hacer la aclaración que esta la primer componente domina la variable *jumps*, aunque lo más probable es que no sea representativa con respecto al total, ya que es la que menos representa, con un porcentaje inferior al 5%). Para la segunda dirección latente (en franjas diagonales), la variable que más contribuye en **X** es la variable *pulse*. Viendo las franjas verticales en las figuras, nos damos cuenta de la cantidad de información que estaríamos perdiendo si nos quedamos sólo con las 2 primeras componentes. De las figuras vemos que esta no es significativa si la consideramos en forma relativa a las otras dos componentes.

La última barra en ambos gráficos, que dice total, se refiere a la cantidad total de variación explicada por cada una de las componentes o vectores. Así que estamos hablando de que las variables de la matriz **Y**, serán explicadas sólo en 30% por las variables de la matriz **X**.

El programa de intervalos de confianza (INTERV-CONF) obtiene los valores puntuales ajustados, generados por el modelo PLS, como se muestra abajo. Observamos que los valores ajustados que presentan un error mayor son aquellos atípicos en el conjunto de datos para la variable Y considerada. El programa hace una pausa después de esta salida, para ingresar los valores de las variables **X** para una nueva observación. El programa obtiene su valor predicho. Vea a continuación toda la salida para el conjunto de datos de prueba LINNX y LINNY.

```
> print(YRYP)
               Y real Y predicho
 Observacion 1    162  143.29081
 Observacion 2    110  124.72690
 Observacion 3    101  111.08624
 Observacion 4    105  141.24771
 Observacion 5    155  158.66543
 Observacion 6    101  137.57778
 Observacion 7    101  123.90548
 Observacion 8    125  161.99460
 Observacion 9    200  222.65566
Observacion 10    251  169.36496
Observacion 11    120  162.05673
Observacion 12    210  177.53961
Observacion 13    215  153.09229
Observacion 14     50   10.16756
Observacion 15     70  144.18885
Observacion 16    210  135.57967
Observacion 17     60  115.54596
Observacion 18    230  188.37789
Observacion 19    225  170.54164
Observacion 20    110  159.39423
> print(LIYLS)
```

```
                    L.INF. REAL LIM.SUP.
 Observación 1   31.7658358  162 254.8158
 Observación 2   15.0009813  110 234.4528
 Observación 3   -0.3324171  101 222.5049
 Observación 4   27.7335239  105 254.7619
 Observación 5   41.4224046  155 275.9085
 Observación 6   28.9639878  101 246.1916
 Observación 7    8.8261140  101 238.9848
 Observación 8   52.4196006  125 271.5696
 Observación 9   82.4538780  200 362.8574
Observación 10   57.4884177  251 281.2415
Observación 11   49.7476004  120 274.3659
Observación 12   65.1536744  210 289.9255
Observación 13   37.7209607  215 268.4636
Observación 14 -126.4199321   50 146.7551
Observación 15   28.6975164   70 259.6802
Observación 16   19.3970786  210 251.7623
Observación 17    2.7851449   60 228.3068
Observación 18   73.9186751  230 302.8371
Observación 19   58.4135440  225 282.6697
Observación 20   34.5830650  110 284.2054
> print(PORCNC)
 PORCENTAJE DE INTERVALOS DE CONFIANZA QUE NO CONTIENEN A SU VALOR REAL:
                                                                        0

> print(LIYLS)
              L.INF. PREDICHO LIM.SUP.
Observación 1 41.4224 158.6654 275.9085
```

Los resultados de esta salida podrían aparecer como algo desalentador para utilizar PLS por la amplitud tan amplia de algunos de los intervalos, aún se puede argumentar a favor de la técnica que el conjunto de datos utilizado, no es el más favorecedor, primero porque las variables propias del mismo no teinen un poder predictivo muy bueno, y en segundo término porque el número de datos es muy pequeño comparado con el número de variables y es bien sabido que la proporción entre el número de datos y de variables debería ser grande. Aún así, vemos que las predicciones puntuales no son del todo erráticas y pueden ser útiles.

Como parte final, aparece la salida del programa gráficos de control (GRAFICOS). Este programa genera básicamente tres gráficas diferentes. La primera, que aparece a continuación, muestra la contribución de cada una de las variables al error de predicción del conjunto de prueba.

Antes de interpretar las próximas dos gráficas que tratan del error de predcción, es conveniente una aclaración. Generalmente al hablar del error de predicción, éste se entiende en su forma clásica de la diferencia entre las variables respuesta y sus predicciones, en las próximas gráficas no es así. El error de predicción se refiere a las predicciones de las variables **X** con sus valores generados por el modelo con dimensiones reducidas en diferencia con sus valores reales. Desde este punto de vista entonces, estamos hablando de una cantidad que podría funcionar como el mismo error clásico de predicción, en la medida en que la relación predictiva entre las variables latentes de **X** y **Y**, sea adecuada.

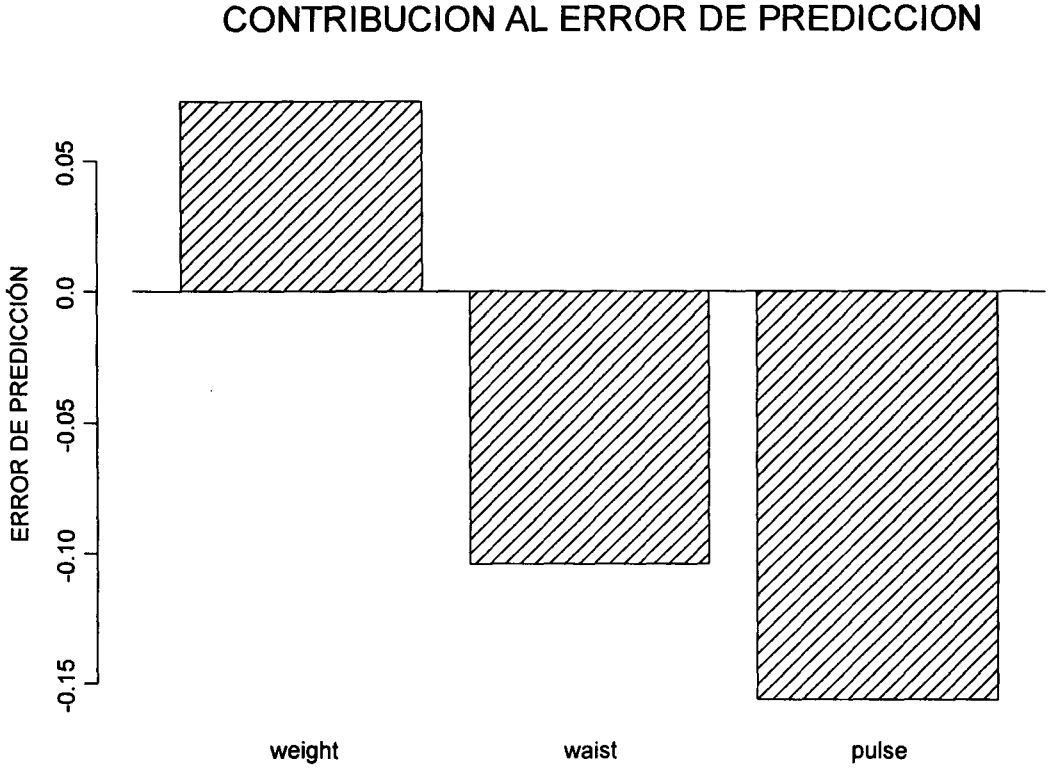


Figura 16

En la figura, podemos ver corroborado lo que ya habíamos interpretado con las gráficas de barras de la contribución al error de predicción de las diferentes variables de X y Y, específicamente que la variable *pulse* es la que es "explicada" con más error y por ende contribuye más al error de predicción, como se muestra por el tamaño de la barra. La dirección de las barras (hacia arriba o abajo) no tiene ningún efecto al trabajar con el error cuadrático, pero podría ser de utilidad en algunas situaciones específicas, determinar si el error viene de una subestimación o sobreestimación del valor real.

A continuación, de la misma rutina, aparece el cálculo del error de predicción cuadrático (SPE) generado con el conjunto de datos de prueba, pero que se utiliza como el error de predicción típico del modelo y puede utilizarse como una gráfica de control para nuevas observaciones; de hecho, puede verse en esta salida, el comportamiento de una nueva observación (vease el punto sólido). Vemos que está dentro de los límites del 95% y 99%, lo que nos indicaría que estamos con una observación similar a la del conjunto de prueba, o sea, en control. Nuevamente, debemos hacer énfasis en que estamos hablando del error de predicción de las variables **X**.

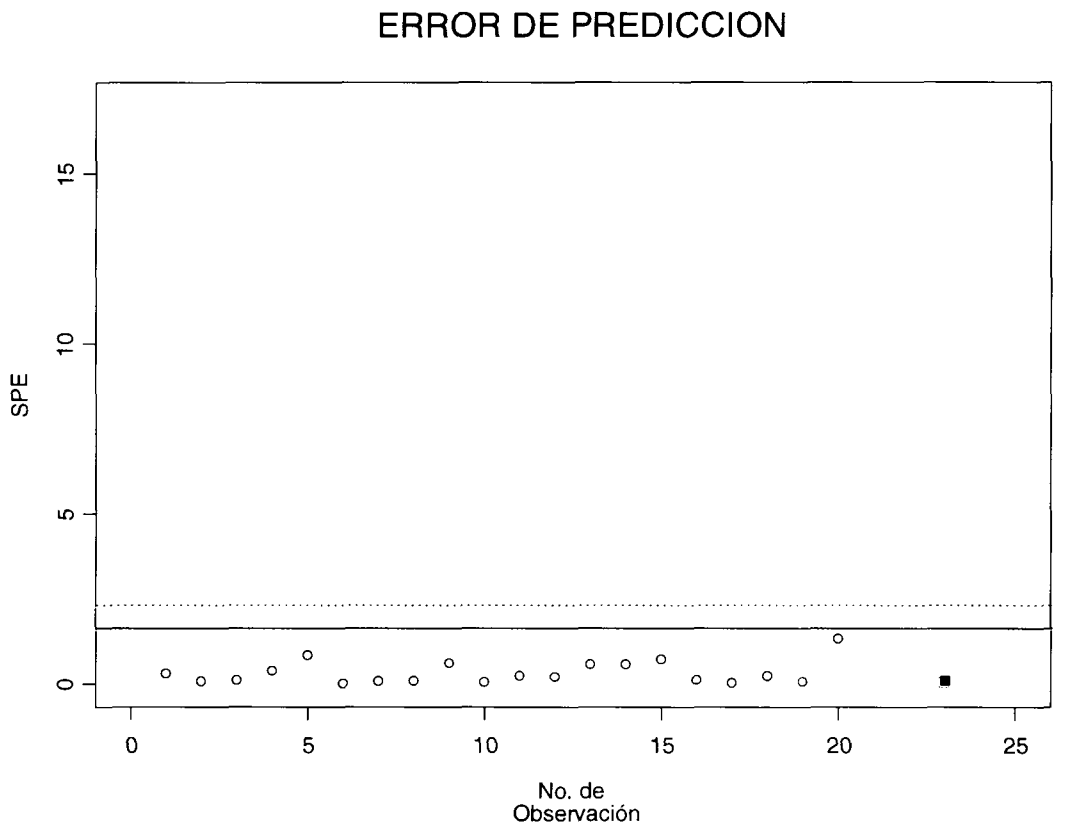
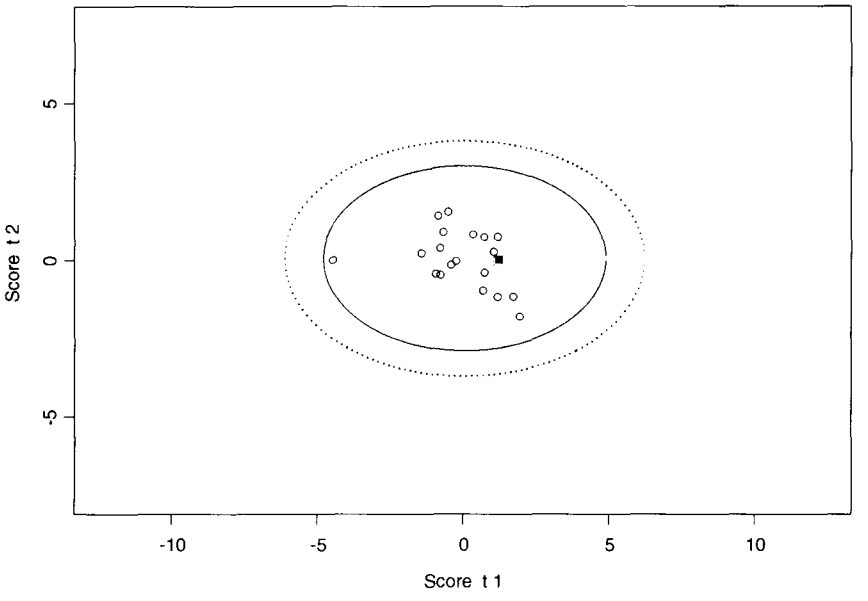


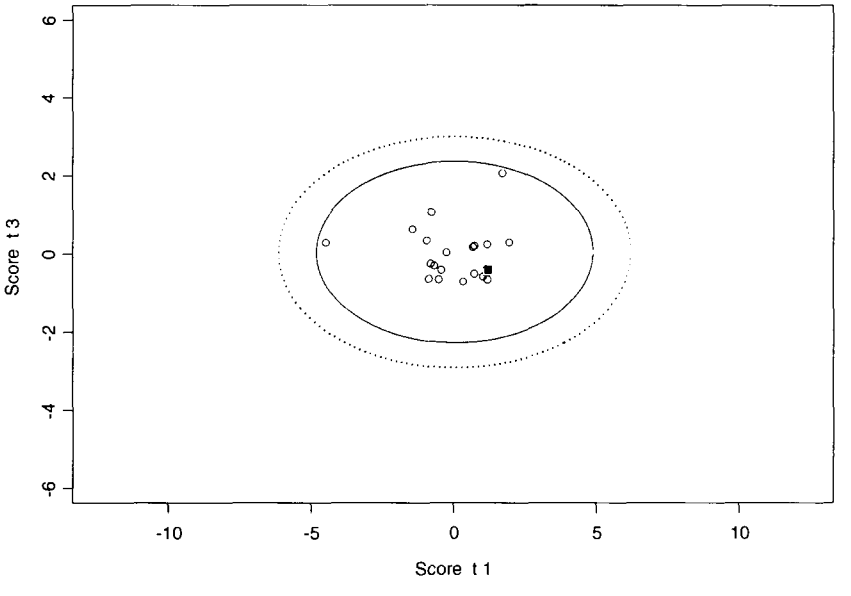
Figura 17

En la figura 18, vemos los 3 gráficos de control (elipses) de los scores por pares. Se muestran los límites de confianza del 95% y 99%, como línea punteada. Como pertenece a la misma salida que generó la figura 17, también refleja la contribución de scores de la nueva observación, que aparece en sólido. Nuevamente aquí vale la pena la reiteración que los escores sobre los que se generaron las elipses y los que por ende se utilizan para "ubicar" a las nuevas observaciones, son scores **t**, es decir, para que esto aporte información valiosa al sistema de monitoreo, se requiere poder medir una especie de "potencia" de la gráfica que indique si un dato fuera de límites (atípico) en estas elipses, será indicativo de forma inequívoca (o en algún grado) de un atípico para los valores de la variable respuesta (**Y**). Esto funcionaría en una relación directa en cuanto a outliers de scores de **X** implican outlires en scores de **Y**. Esto no siempre sucede de esta forma. Más bien estas relaciones son complicadas y merecen estudio aparte.

ELIPSE DE CONFIANZA DE t 1 vs t 2



ELIPSE DE CONFIANZA DE t 1 vs t 3



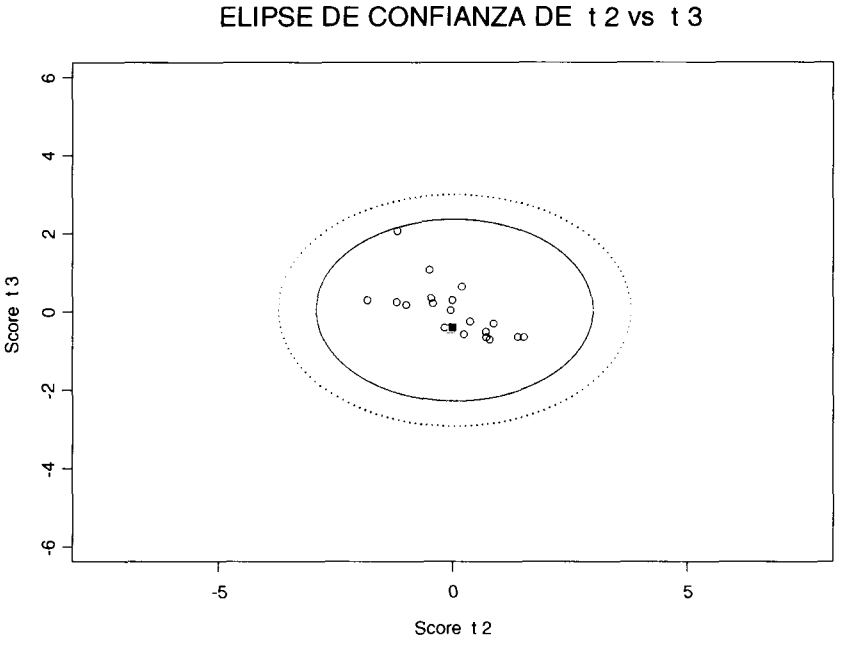
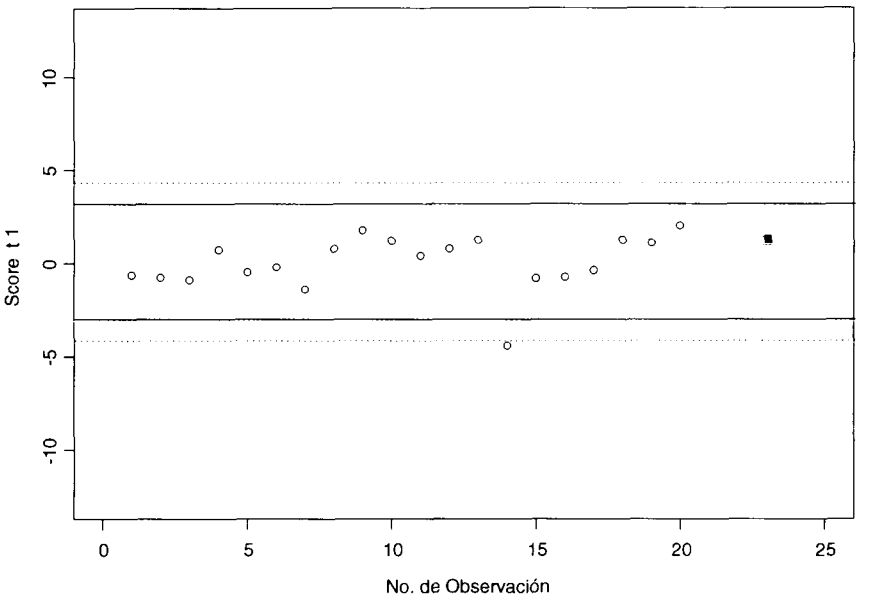


Figura 18

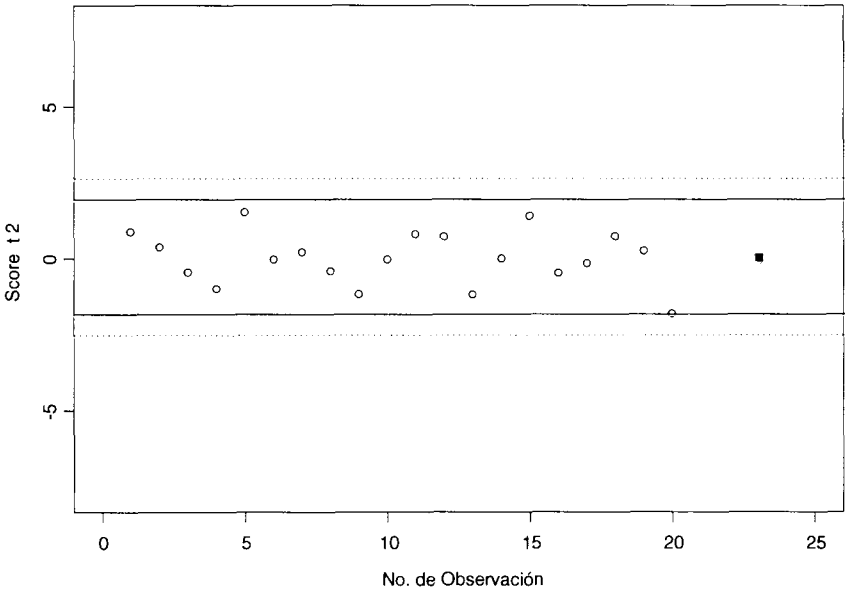
De la misma rutina, el tercer tipo de gráfico es el que ya se presentó antes en la sección de 2.1.6 sobre PCA. Tenemos aquí una gráfica de control para los scores tratados de forma individual. Vemos una de las observaciones del conjunto original cuyo score t_1 no se comporta como los demás (aparece sobre el límite de control inferior). Este es un buen ejemplo de un posible "outlier" que habría que revisar de cerca. Si se puede descartar, posiblemente genere unos límites de control más restrictivos o mejores identificadores de situaciones de fuera de control. Aparece también el "punto sólido" del score generado al proyectar la nueva observación sobre la dirección latente primera indicada por el proceso PLS.
Más abajo también aparece el score correspondiente a las otras dos direcciones latentes.

La forma en que se deberían utilizar estas gráficas es, primero vigilar de cerca el monitoreo de las elipses de control y en el momento en que éstas indíquen un posible "fuera de control", revisar estas gráficas de scores individuales para tratar de identificar cuál es el que está ocasionando el problema. Si además de esto, se van generando configuraciones de patrones de scores identificables de causas típicas de fallos, estos gráficos de control serán mucho mejor explotados.

GRAFICA DE CONTROL INDIVIDUAL PARA t 1



GRAFICA DE CONTROL INDIVIDUAL PARA t 2



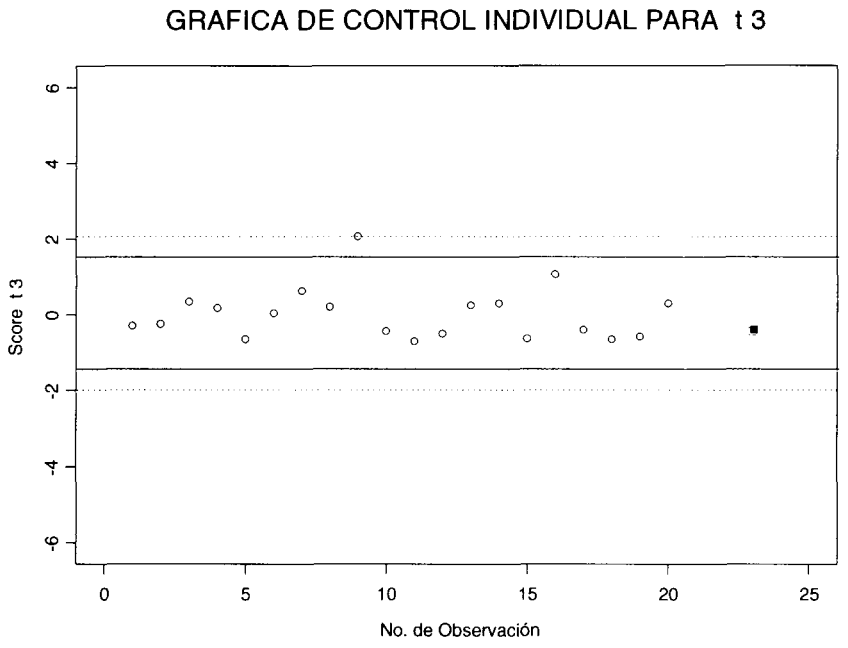


Figura 19

Resumiendo, la utilización de las gráficas que se programaron y cuyas salidas se han presentado, tenemos lo siguiente:

1) Se aplica la rutina que determina el número de dimensiones aecuadas para un situación dada, basándose en un conjunto de observaciones que represente el "funcionamiento adecuado" del proceso. En el caso que trabajamos de muestra, aquí. El número de dimensiones podría ser uno, como lo vimos en la gráfica 12, de dimensiones retenidas.

2) Se aplica la rutina de identificación de outliers, en la salida de ésta, "identificamos" con un clic derecho del "mouse" los puntos que estén aprtados del resto para conocer el número de dato correspondiente en nuestra base y revisarlo para posibe descarte del mismo por ser atípico. En nuestro ejemplo, aparecen un poco alejados del resto, los datos 9 y 16 en la figura 13. Procedería revisar esas observaciones para ver si es posible descartarlas como outliers.

3) Después del número dos, puede ser conveniente volver al paso uno para determinar si ha cambiado el número de componentes retenidas por la eliminación de los outliers.

4) Aplicar el programa de variabilidad explicada para un análisis descriptivo de la contribución que están teniendo cada una de las variables originales en las nuevas

variables o dimensiones latentes. En las figuras 14 y 15, pudimos determinar en nuestro problema, que la variable *cintura* está muy "alineada" al primer eje PLS, mientras que *el pulso* es la variable que menos contribuye a esta dimensión. También determinamos que la variable salto iba a ser difícil de pronosticar, pues aún con las tres componentes, solo se cubre una fracción muy pequeña de la misma.

5) De aquí en delante, el análisis deja de ser descriptivo para volverse inferencial o predictivo. Se aplican las rutinas de gráficas de control, se ubica en ellas los puntos o valores correspondientes a nuevas observaciones y se concluye que el proceso permanece estable mientras estos puntos permanezcan dentro de los gráficos de control. En nuestro ejemplo, así ocurrió con una sola observación nueva, marcada en cuadro sólido.

6) En las elipses de control, al igual que en las gráficas de control del error de predicción cuadrático identificamos desviaciones del proceso que pudieran incidir en producto de mala calidad (fuera de conformancia). Para los datos LINN, todas las gráficas observaron el nuevo valor dentro de límites, indicativo de que era muy similar a los 20 originales.

7) Cuando se presenta un dato fuera de los límites de control, se observan las gráficas de scores individuales para detectar si alguno de los scores de forma individual tiene un comprtamiento errático (diferente de los originales y por lo tanto fuera de control) con la esperanza de que con esto se pueda identificar más rápidamente la causa del problema. Por ejemplo, en el caso analizado aquí, sabemos que la variable pulso no contribuye, prácticamente, a la primer componente, de forma que si tuviéramos un fuera de control en la gráfica de score individual t1, no es factible que el "culpable" sea el pulso del nuevo dato.

3.1 Conclusiones y recomendaciones

Se debe tener bien claro que lo mínimo que requiere un proceso multivariado para poder utilizar estas técnicas de monitoreo y control, es la facilidad de tener mediciones de las variables del proceso en un período de tiempo corto (principalmente en MPLS).

Es importante, antes del análisis, el escalamiento de las diferentes variables a considerar. Generalmente, el centrado y escalado de estas variables para tener media cero y desviación estándar uno, es adecuado (esto está programado así en las diferentes rutinas, salvo en las primeras tres, que muestran sólo el algoritmo directo y las representaciones gráficas), pero en ocasiones donde sea más adecuado algún otro tipo de escalamiento, se debe hacer una pequeña modificación del programa.

Los conjuntos de mediciones para referencia (a partir del cual se construirá el modelo), deben ser escogidos de tal forma que reflejen el estado "normal" del proceso (esto es, cuando trabaja adecuadamente). Se debe cuidar que toda

causa que se quiera detectar como una alarma en el procedimiento de monitoreo, no esté presente en algunas de las mediciones de este conjunto.

El sistema de monitoreo será útil, mientras el desempeño normal del proceso esté reflejado en sus límites; cuando éste ha cambiado, habrá que construir otro esquema de monitoreo, adecuado a las nuevas especificaciones.

Sería deseable realizar un análisis de sensibilidad, para evaluar como se ve afectada en el caso general PLS por errores de medición (o de contaminación de los datos por otras situaciones). Si se va a implementar PLS en una situación específica, entonces bastaría con restringirse al tipo de datos y errores de medición comunes para las situaciones a monitorear con más frecuencia. Esto debe simplificar este tipo de análisis. Podría aprovecharse para tal fin la relación conocida para "contaminar" una variable,

Y contaminada = (1-r)**Y** + **U** para valores de r entre cero y uno.

Si se utiliza la misma relación para "contaminar" a las variables de **X** se puede entonces tener una idea del efecto de estos errores en el monitoreo a través de las gráficas y generar un valor límite de grado de contaminación que se podría tolerar para que el sistema de monitoreo pudiera seguir siendo efectivo.

Una forma de aprovechar al máximo las herramientas de control programadas sería "conectarlas" directamente a un sistema de control numérico instalado en línea que lleve de forma automática las mediciones de las variables del proceso, pero esto queda fuera del alcance de este trabajo.

4. Anexos.

Para las rutinas generadas hay que tomar en cuenta lo siguiente:

1) Por la forma de operar de s-plus, se requiere crear y especificar un archivo donde se almacenen los datos mientras opera internamente. Los programas ya tienen especificados estos archivos de trabajo (los carga automáticamente), de manera que lo único que resta al respecto es crearlos en la ruta especificada (primer línea del programa).

2) Las bases de datos con las que se trabajarán, deben crearse en Excel, guardarse con el formato .prn (esto es, la opción de caracteres delimitados por espacios) y colocarse en la ruta adecuada.

3) Habrá que especificar antes de la corrida de algún programa, los datos básicos referentes a la constitución de la matriz donde están las observaciones.

   n = número de observaciones en la muestra
   m = número de variables del proceso (**X**)
   p = número de variables de calidad (**Y**)
   A = número de dimensiones a considerar en el análisis PLS

Esto se hace en las primeras líneas de cada programa, bajo el encabezado de "definición de variables".

4.1 Rutinas de programación (S-PLUS)

Este apartado consta 6 rutinas programadas en S-PLUS que se utilizaron para este trabajo. A continuación aparece su nombre descriptivo y entre paréntesis su nombre de programa s-plus en disco:

1) Proyección gráfica PCA (PCA GRAFICO)
2) Comparación gráfica entre direcciones PCA Y PLS (PCA-PLS COMP GRAFICO)
3) Algoritmo NIPALS para PLS (NIPALS-PLS)
4) Criterios de detención (CRIT-DETEN)
5) Eliminación de outliers (ELIMIN-OUT)
6) Gráficos de control (GRAFICOS)
7) Intervalos de confianza (INTERV-CONF)
8) Variabilidad explicada (VARIAB-EXPLIC)

1) Proyección gráfica PCA

```
#Programa para representar gráficamente la descomposición de componentes
   principales#
#en 2 dimensiones usar datos UNOX DOSX TRESX O CUATROX#

attach("C:\\MI TESIS\\Espacios de trabajo\\grafico de proyecciones en PCA",pos=1)

  import.data(DataFrame="X",FileName="C:\\MI
   TESIS\\Datos\\CUATROX.XLS",FileType="EXCEL")
  X <- as.matrix(X)
  CENTROX <- colMeans(X)
  CORX <- t(X)%*%X
  EIG <- eigen(CORX)
  if(EIG[[2]][1,1]<0) EIG[[2]][,1] <- -1*EIG[[2]][,1]
  if(EIG[[2]][1,2]<0) EIG[[2]][,2] <- -1*EIG[[2]][,2]
  PCA1 <- EIG[[2]][1,1]
  PCA2 <- EIG[[2]][1,2]
  TETA1R <- acos(PCA1)
  TETA2R <- pi-acos(PCA2)
  TETA1G <-  TETA1R*180/pi
  TETA2G <- TETA2R*180/pi
  m1 <- tan(TETA1R)
  m2 <- tan(TETA2R)
  b1 <- CENTROX[2]-m1*CENTROX[1]
  b2 <- CENTROX[2]-m2*CENTROX[1]
  par(pty="s")
  plot(X,xlim=c(0,20),ylim=c(0,20),xlab="X1",ylab="X2",bty="l",main="Proyecciones
   sobre direcciones latentes en PCA")
  abline(b1,m1)
  abline(b2,m2)
  COORDSPC1 <-
   t(t(t(t(EIG[[2]][,1])%*%t(scale(X,scale=F)))%*%t(EIG[[2]][,1]))+CENTROX)
  COORDSPC2 <-
   t(t(t(t(EIG[[2]][,2])%*%t(scale(X,scale=F)))%*%t(EIG[[2]][,2]))+CENTROX)
  points(COORDSPC1,col=8,pch=18)
  points(COORDSPC2,col=7,pch=18)

print(TETA1G)
print(TETA2G)

detach("C:\\MI TESIS\\Espacios de trabajo\\grafico de proyecciones en PCA",pos=1)
```

2) Comparación gráfica entre direcciones PCA y PLS

```
#Programa para comparar gráficamente los ejes PCA con los PLS

attach("C:\\MI TESIS\\Espacios de trabajo\\comparacion de ejes PCA-PLS",pos=1)
  import.data(DataFrame="X",FileName="C:\\MI
   TESIS\\Datos\\CUATROX.XLS",FileType="EXCEL")
  import.data(DataFrame="Y",FileName="C:\\MI
   TESIS\\Datos\\CUATROY.XLS",FileType="EXCEL")
  X <- as.matrix(X); Y <- as.matrix(Y)

  CENTROIDEX <- colMeans(X) ; CENTROIDEY <- colMeans(Y)
  XTX <- t(X)%*%X ; XTYTXTY <- t(Y)%*%X%*%t(X)%*%Y ; XTYTXTY <-
   t(X)%*%Y%*%t(Y)%*%X
```

```
  EIGPCA <- eigen(XTX) ; EIGPLS <- eigen(XTYTXTY)
  if(EIGPCA[[2]][1,1]<0) EIGPCA[[2]][,1] <- -1*EIGPCA[[2]][,1] ;
   if(EIGPLS[[2]][1,1]<0) EIGPLS[[2]][,1] <- -1*EIGPLS[[2]][,1]
  if(EIGPCA[[2]][1,2]<0) EIGPCA[[2]][,2] <- -1*EIGPCA[[2]][,2] ;
   if(EIGPLS[[2]][1,2]<0) EIGPLS[[2]][,2] <- -1*EIGPLS[[2]][,2]
  LOADINGSPCA1 <- EIGPCA[[2]][1,1] ; LOADINGSPCA2 <- EIGPCA[[2]][1,2] ; WsPLS1 <-
   EIGPLS[[2]][1,1] ; WsPLS2 <- EIGPLS[[2]][1,2]
  TETA1PCAR <- acos(LOADINGSPCA1) ; TETA2PCAR <- pi-acos(LOADINGSPCA2) ;
   TETA1PLSR <- acos(WsPLS1) ; TETA2PLSR <- pi-acos(WsPLS2)
  TETA1PCAG <- TETA1PCAR*180/pi ; TETA2PCAG <- TETA2PCAR*180/pi ; TETA1PLSG <-
   TETA1PLSR*180/pi ; TETA2PLSG <- TETA2PLSR*180/pi
  m1PCA <- tan(TETA1PCAR) ; m2PCA <- tan(TETA2PCAR) ; m1PLS <- tan(TETA1PLSR) ;
   m2PLS <- tan(TETA2PLSR)
  b1PCA <- CENTROIDEX[2]-m1PCA*CENTROIDEX[1] ; b2PCA <- CENTROIDEX[2]-
   m2PCA*CENTROIDEX[1] ; b1PLS <- CENTROIDEX[2]-m1PLS*CENTROIDEX[1] ; b2PLS <-
   CENTROIDEX[2]-m2PLS*CENTROIDEX[1]
  par(pty="s")
  plot(X,xlim=c(0,20),ylim=c(0,20),xlab="X1",ylab="X2",bty="l",main="Proyecciones
   PCA, y ejes PLS juntos para comparación")
  abline(b1PCA,m1PCA) ; abline(b2PCA,m2PCA) ; abline(b1PLS,m1PLS,lty=8) ;
   abline(b2PLS,m2PLS,lty=8)

  SCORESPCA <- t(t(EIGPCA[[2]])%*%t(scale(X,scale=F)))
  COORDSX1PROYPCA1Y2 <-t(CENTROIDEX[1]+cos(atan(c(m1PCA,m2PCA)))*t(SCORESPCA))
  COORDSX2PROYPCA1Y2 <-t(CENTROIDEX[2]+sin(atan(c(m1PCA,m2PCA)))*t(SCORESPCA))
  COORDSX1X2PROYPCA1 <-
   matrix(c(COORDSX1PROYPCA1Y2[,1],COORDSX2PROYPCA1Y2[,1]),ncol=2)
  COORDSX1X2PROYPCA2 <-
   matrix(c(COORDSX1PROYPCA1Y2[,2],COORDSX2PROYPCA1Y2[,2]),ncol=2)
  points(COORDSX1X2PROYPCA1,col=8,pch=18)
  points(COORDSX1X2PROYPCA2,col=7,pch=18)

  #ESTE BLOQUE ES PARA PINTAR LAS PROYECCIONES DE LOS PUNTOS EN LOS EJES PLS, SI
   FUNCIONA, PERO NO LO CONSIDERO NECESARIO INCLUIR#
  #SCORESPLS <- t(t(EIGPLS[[2]])%*%t(scale(X,scale=F)))
  #COORDSX1PROYPLS1Y2 <-t(CENTROIDEX[1]+cos(atan(c(m1PLS,m2PLS)))*t(SCORESPLS))
  #COORDSX2PROYPLS1Y2 <-t(CENTROIDEX[2]+sin(atan(c(m1PLS,m2PLS)))*t(SCORESPLS))
  #COORDSX1X2PROYPLS1 <-
   matrix(c(COORDSX1PROYPLS1Y2[,1],COORDSX2PROYPLS1Y2[,1]),ncol=2)
  #COORDSX1X2PROYPLS2 <-
   matrix(c(COORDSX1PROYPLS1Y2[,2],COORDSX2PROYPLS1Y2[,2]),ncol=2)
  #points(COORDSX1X2PROYPLS1,col=10,pch=18)
  #points(COORDSX1X2PROYPLS2,col=12,pch=18)

  print(TETA1PCAG) ; print(TETA1PLSG) ; #print(SCORESPCA) ; print(EIGPLS) ;
   print(t) ; print(p) ; print (X-t%*%t(p)); print (X-t%*%t(EIGPLS[[2]]))
detach("C:\\MI TESIS\\Espacios de trabajo\\comparacion de ejes PCA-PLS",pos=1)
```

3) Algoritmo NIPALS para PLS

```
#ALGORITMO NIPALS (REALIZA PLS EN ETAPAS)#

attach("C:\\MI TESIS\\Espacios de trabajo\\NipalsPLS",pos=1)
  import.data(DataFrame="X",FileName="C:\\MI
   TESIS\\Datos\\CUATROX.XLS",FileType="EXCEL")
  import.data(DataFrame="Y",FileName="C:\\MI
   TESIS\\Datos\\CUATROY.XLS",FileType="EXCEL")
  Xo <- as.matrix (X); Yo <- as.matrix (Y)
  X <- as.matrix(X); Y <- as.matrix(Y)
```

```
  A <- ncol(X)
## CAMBIE ncol(X) POR EL NÚMERO DE DIMENSIONES QUE GUSTE, COMO MÁXIMO M ##
  SCORESX <- matrix(nrow=nrow(X),ncol=A);  SCORESY <- matrix(nrow=nrow(X),ncol=A)
  LOADINGSX <- matrix(nrow=ncol(X),ncol=A);LOADINGSY <-
   matrix(nrow=ncol(Y),ncol=A)
  ITERACIONES <- 1:A
  wS <- matrix(nrow=ncol(X),ncol=A)
  bS <- matrix(nrow=nrow(X),ncol=A)
  tact <- 1:nrow(X)

  for (j in 1:A)
  {u <- Y[,1]

     for (i in 1:1000)
     {t <- tact
     w <- t(X)%*%u/vecnorm(u)^2
     w <- scale(w,center=F,scale=vecnorm(w))
     tact <- X%*%w/vecnorm(w)^2
     q <- t(Y)%*%t/vecnorm(t)^2
     q <- scale(q,center=F,scale=vecnorm(q))
     u <- Y%*%q/vecnorm(q)^2
     ITERACIONES[j] <- i
     if (max(abs(tact-t))<0.0000000001) break}

  pold <- t(X)%*%t/vecnorm(t)^2
  normpold <- vecnorm(pold)
  p <- scale(pold,center=F,scale=normpold)
  t <- t*normpold
  w <- w*normpold
  b <- sum(t(u)%*%t/vecnorm(t)^2)
  X <- X-t%*%t(p)
  Y <- Y-b*(t%*%t(q))
  LOADINGSX[,j] <- p
  LOADINGSY[,j] <- q
  wS[,j] <- w
  bS[,j] <- rep(b,nrow(X))
  SCORESX[,j] <- t
  SCORESY[,j] <- u}

  print(SCORESX); print(LOADINGSX); print(ITERACIONES)
  cbind(Xo-SCORESX%*%t(LOADINGSX),Yo-(bS*SCORESX)%*%LOADINGSY)

#SCORESX;LOADINGSX;wS

detach("C:\\MI TESIS\\Espacios de trabajo\\NipalsPLS",pos=1)
```

4) Criterios de detención

```
attach("C:\\MI TESIS\\Espacios de trabajo\\criterios de detencion",pos=1)

# PROGRAMA DE LOS CRITERIOS PARA OBTENER LAS DIMENSIONES SIGNIFICATIVAS

#INICIALIZACION DE VARIABLES
m <- 3 # DE VARS. INDEP.(x's)
p <- 3 # DE VARS. DEP.(y's)
n <- 20 # DE MUESTRAS EN EL CONJUNTO DE CALIBRACION
A <- 3 # DE DIMENSIONES A PROBAR
g <- 4 # No. de subgrupos en que se dividen los datos
```

```
#CREADO DE RECIPIENTES
neng <- n/g # No.observaciones dejadas fuera c/vez
Mxpt <- matrix(rep(0,A*m),ncol=m)
Mxqt <- matrix(rep(0,A*p),ncol=p)
Mxwt <- matrix(rep(0,A*m),ncol=m)
Mxt <- matrix(rep(0,n*A),ncol=A)
Mxu <- matrix(rep(0,n*A),ncol=A)
Mxb <- matrix(rep(0,A*A),ncol=A)
X <- matrix(rep(0,n*m),ncol=m)
Y <- matrix(rep(0,n*p),ncol=p)
NORMAFa <- matrix(rep(0,A+1),ncol=1)
VeccentX <- matrix(rep(0,m),ncol=m)
VecescX <- matrix(rep(0,m),ncol=m)
VeccentY <- matrix(rep(0,p),ncol=p)
VecescY <- matrix(rep(0,p),ncol=p)

# LECTURA DE DATOS
X <- read.table("C:\\MI TESIS\\Datos\\LINNXf.PRN",header=T)
Y <- read.table("C:\\MI TESIS\\Datos\\LINNYf.PRN",header=T)
#X <- import.data(DataFrame="X",FileName="C:\\MI
    TESIS\\Datos\\LINNX.XLS",FileType="EXCEL")
#Y <- import.data(DataFrame="Y",FileName="C:\\MI
    TESIS\\Datos\\LINNY.XLS",FileType="EXCEL")
X <- data.matrix(X)
Y <- data.matrix(Y)
PROC <- X
CALID <- Y

# ESCALAMIENTO DE VARIABLES
E <- scale(X)
FF <- scale(Y)

#GUARDANDO LOS VALORES DEL ESCALAMIENTO
for (i in 1:m)
 { VeccentX[1,i] <- mean(X[,i])
   VecescX[1,i] <- sqrt(var(X[,i])) }
for (i in 1:p)
 { VeccentY[1,i] <- mean(Y[,i])
   VecescY[1,i] <- sqrt(var(Y[,i])) }

# PLS
if( p > 1 ) {
for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

  for (j in 1:100)
    {

     # en el block X
     wt <- t(u)%*%E/vecnorm(u)^2
     wt <- wt/vecnorm(wt)
     tnew <- E%*%t(wt)/vecnorm(wt)^2

     # en el block Y
     qt <- t(tnew)%*%FF/vecnorm(tnew)^2
     qt <- qt/vecnorm(qt)
     u <- FF%*%t(qt)/vecnorm(qt)^2

     #checar convergencia
```

```
    dif <- tnew-t
    c <- vecnorm(dif)
    if (c < 0.0000005) break else t <- tnew
    }

  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew%*%qt
  NORMAFa[i+1,1] <- sqrt(sum(diag(t(FF)%*%FF)))
  }

NORMAFa[1,1] <- sqrt(sum(diag(t(scale(Y))%*%scale(Y))))
a <- t(t(0:A))
tabla <- data.frame(a,NORMAFa)
plot(a,NORMAFa,main=" No. DIMENSIONES vs RESIDUALES",xlab="No. de
dimensiones incluidas",ylab="|Fa|",pch=16)

for(i in 1:A){
SS <- matrix(rep(0,3),ncol=1)
SS[1,1] <- Mxb[i,i]^2*vecnorm(Mxt[,i,drop=F])^2
dh <- Mxu[,i,drop=F]-Mxb[i,i]*Mxt[,i,drop=F]
SS[2,1] <- vecnorm(dh)^2
SS[3,1] <- vecnorm(Mxu[,i,drop=F])^2
df <- matrix(c(1,n-2,n-1),ncol=1)
MS <- SS/df
estF <- MS[1,1]/MS[2,1]
pval <- 1-pf(estF,1,n-2)
Source <- matrix(c("Explained","Residual","Total"),ncol=1)
AnaVar <- data.frame(Source,SS,df,MS,estF,pval)
print(cat("DIMENSION",i,"\n"))
print(AnaVar,"\n")}
print(cat("VARIABILIDAD EXPLICADA POR DIMENSION","\n"))
print(tabla,"\n")
}

else {
#PARA CUANDO HAY UNA SOLA VARIABLE Y

for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]
```

```
      # en el block X
    wt <- t(u)%*%E/vecnorm(u)^2
    wt <- wt/vecnorm(wt)
    tnew <- E%*%t(wt)/vecnorm(wt)^2

    # en el block Y
    qt <- 1


  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew*qt
  NORMAFa[i+1,1] <- sqrt(sum(diag(t(FF)%*%FF)))
  }
NORMAFa[1,1] <- sqrt(sum(diag(t(scale(Y))%*%scale(Y))))
a <- t(t(0:A))
tabla <- data.frame(a,NORMAFa)
plot(a,NORMAFa,main=" No. DIMENSIONES vs RESIDUALES",xlab="No. de
dimensiones incluidas",ylab="|Fa|",pch=16)

for(i in 1:A){
SS <- matrix(rep(0,3),ncol=1)
SS[1,1] <- Mxb[i,i]^2*vecnorm(Mxt[,i,drop=F])^2
da <- Mxu[,i,drop=F]-Mxb[i,i]*Mxt[,i,drop=F]
SS[2,1] <- vecnorm(da)^2
SS[3,1] <- vecnorm(Mxu[,i,drop=F])^2
df <- matrix(c(1,n-2,n-1),ncol=1)
MS <- SS/df
estF <- MS[1,1]/MS[2,1]
invisible(estF[2,1])
pval <- 1-pf(estF,1,n-2)
Source <- matrix(c("Explained","Residual","Total"),ncol=1)
AnaVar <- data.frame(Source,SS,df,MS,estF,pval)
print(cat("DIMENSION",i,"\n"))
print(AnaVar,"\n")}
print(cat("VARIABILIDAD EXPLICADA POR DIMENSION","\n"))
print(tabla,"\n")
}

#ESTE PROGRAMA OBTIENE EL NUMERO DE COMPONENTES PRINCIPALES
#QUE SE DEBEN INCLUIR EN EL PLS(CROSSVALIDATION)

# INICIALIZACION DE VARIABLES
library(Matrix) #cargando una library
```

```
# INCIALIZACION DE RECIPIENTES

#CREADO DE RECIPIENTES
Mxpt <- matrix(rep(0,m*A),nrow=A)
Mxqt <- matrix(rep(0,p*A),nrow=A)
Mxwt <- matrix(rep(0,m*A),nrow=A)
Mxt <- matrix(rep(0,(n-neng)*A),ncol=A)
Mxu <- matrix(rep(0,(n-neng)*A),ncol=A)
Mxb <- matrix(rep(0,A*A),ncol=A)
McentX <- matrix(matrix(0,neng*m),ncol=m)
MescX <- matrix(matrix(0,neng*m),ncol=m)
McentY <- matrix(matrix(0,neng*p),ncol=p)
MescY <- matrix(matrix(0,neng*p),ncol=p)
suma <- rep(0,A)
NORFo <- rep(0,A)
XVAL <- t(t(rep(0,A)))

#ESCOGIENDO SUBGRUPOS ALEATORIAM
orden <- sample(n,replace=F)

#PARA CUANDO HAY VARIAS VARIABLES DE CALIDAD
if( p > 1 ) {

#VA ELIMINANADO UN CONJUNTO PEQUEÑO DEL TOTAL(SOBRE EL QUE SE HARA
#PREDICCION)
for(sub in 1:g)
 { Xjuega <- PROC[-orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]
  Xelim <- PROC[orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]
  Yjuega <- CALID[-orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]
  Yelim <- CALID[orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]

  X <- Xjuega; Xnew <- Xelim; Y <- Yjuega; Ynew <- Yelim

# ESCALAMIENTO DE X y Y
E  <- scale(X)
FF  <- scale(Y)

#GUARDANDO LAS MATRICES DE CENTRADO Y ESCALADO PARA USO POSTERIOR
for (i in 1:m)
  {McentX[ ,i] <- rep(mean(X[ ,i]),neng)
   MescX[,i] <- rep(sqrt(var(X[ ,i])),neng)}
for (i in 1:p)
  {McentY[,i] <- rep(mean(Y[ ,i]),neng)
   MescY[,i] <- rep(sqrt(var(Y[ ,i])),neng)}

# RUTINA PLS
for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

  for (j in 1:100)
    {

     # en el block X
     wt <- t(u)%*%E/vecnorm(u)^2
     wt <- wt/vecnorm(wt)
     tnew <- E%*%t(wt)/vecnorm(wt)^2

     # en el block Y
```

```
    qt <- t(tnew)%*%FF/vecnorm(tnew)^2
    qt <- qt/vecnorm(qt)
    u <- FF%*%t(qt)/vecnorm(qt)^2

    #checar convergencia
    dif <- tnew-t
    c <- vecnorm(dif)
    if (c < 0.0000005) break else t <- tnew
    }

  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew%*%qt
  }

#CALCULO DE LAS PREDICCIONES PARA LAS Y ELIMINADAS
Xnewnorm <-(Xnew-McentX)/MescX
Ynewnorm <- (Ynew-McentY)/MescY
Eo <- Xnewnorm
Fo <- Ynewnorm

for(z in 1:A)
{ tpred <- Eo%*%t(Mxwt[z, ,drop=F])
  Eo <- Eo-tpred%*%Mxpt[z, ,drop=F]
  Fo <- Fo-Mxb[z,z]*tpred%*%Mxqt[z, ,drop=F]
  NORFo[z] <- norm.Matrix(Fo,type ="F")
  suma[z] <- suma[z]+sum(Fo^2) } }

PRESS <- suma/(n*p)
for(i in 2:A)
{XVAL[i,1] <- sqrt(PRESS[i])/NORFo[i-1]}
 XVAL[1,1] <- sqrt(PRESS[1])/norm.Matrix(Ynewnorm,type ="F")
a <- t(t(1:A))
XVALF <- data.frame(a,XVAL)
XVALF <- list(ValorXVAL=XVALF)
print(XVALF)

}
else {
#PARA CUANDO HAY UNA SOLA VARIABLE Y

#VA ELIMINANADO UN CONJUNTO PEQUEÑO DEL TOTAL(SOBRE EL QUE SE HARA
#PREDICCION)
for(sub in 1:g)
```

```
 { Xjuega <- PROC[-orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]
  Xelim <- PROC[orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]
  Yjuega <- CALID[-orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]
  Yelim <- CALID[orden[(neng*sub-(neng-1)):(neng*sub)], ,drop=F]

  X <- Xjuega; Xnew <- Xelim; Y <- Yjuega; Ynew <- Yelim

# ESCALAMIENTO DE X y Y
E  <- scale(X)
FF  <- scale(Y)

#GUARDANDO LAS MATRICES DE CENTRADO Y ESCALADO PARA USO POSTERIOR
for (i in 1:m)
  {McentX[ ,i] <- rep(mean(X[ ,i]),neng)
   MescX[,i] <- rep(sqrt(var(X[ ,i])),neng)}
for (i in 1:p)
  {McentY[,i] <- rep(mean(Y[ ,i]),neng)
   MescY[,i] <- rep(sqrt(var(Y[ ,i])),neng)}

#RUTINA PLS
for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

  # en el block X
  wt <- t(u)%*%E/vecnorm(u)^2
  wt <- wt/vecnorm(wt)
  tnew <- E%*%t(wt)/vecnorm(wt)^2

  # en el block Y
  qt <- 1

  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew*qt
 }

#CALCULO DE LAS PREDICCIONES PARA LAS Y ELIMINADAS
Xnewnorm <- (Xnew-McentX)/MescX
Ynewnorm <- (Ynew-McentY)/MescY
Eo <- Xnewnorm
Fo <- Ynewnorm
```

```
for(z in 1:A)
{ tpred <- Eo%*%t(Mxwt[z, ,drop=F])
  Eo <- Eo-tpred%*%Mxpt[z, ,drop=F]
  Fo <- Fo-Mxb[z,z]*tpred%*%Mxqt[z, ,drop=F]
  NORFo[z] <- norm.Matrix(Fo,type ="F")
  suma[z] <- suma[z]+sum(Fo^2) } }

PRESS <- suma/(n*p)
for(i in 2:A)
{XVAL[i,1] <- sqrt(PRESS[i])/NORFo[i-1]}
 XVAL[1,1] <- sqrt(PRESS[1])/norm.Matrix(Ynewnorm,type ="F")
a <- t(t(1:A))
XVALF <- data.frame(a,XVAL)
XVALF <- list(ValorXVAL=XVALF)
print(XVALF)
}

detach("C:\\MI TESIS\\Espacios de trabajo\\criterios de detencion",pos=1)
```

5) Eliminación de outliers

```
attach("C:\\MI TESIS\\Espacios de trabajo\\eliminacion de outliers",pos=1)

#ALGORITMO DEL TUTORIAL

#INICIALIZACION DE VARIABLES
m <- 3 # DE VARS. INDEP.(x's)
p <- 3 # DE VARS. DEP.(y's)
n <- 20 # DE MUESTRAS EN EL CONJUNTO DE CALIBRACION
A <- 3 # DE DIMENSIONES PLS

# LECTURA DE DATOS
X <- read.table("C:\\MI TESIS\\Datos\\LINNXf.PRN",header=T)
Y <- read.table("C:\\MI TESIS\\Datos\\LINNYf.PRN",header=T)
#X <- import.data(DataFrame="X",FileName="C:\\MI
   TESIS\\Datos\\LINNX.XLS",FileType="EXCEL")
#Y <- import.data(DataFrame="Y",FileName="C:\\MI
   TESIS\\Datos\\LINNY.XLS",FileType="EXCEL")
X <- data.matrix(X)
Y <- data.matrix(Y)

#CREADO DE RECIPIENTES
Mxpt <- matrix(rep(0,A*m),ncol=m)
Mxqt <- matrix(rep(0,A*p),ncol=p)
Mxwt <- matrix(rep(0,A*m),ncol=m)
Mxt <- matrix(rep(0,n*A),ncol=A)
Mxu <- matrix(rep(0,n*A),ncol=A)
Mxb <- matrix(rep(0,A*A),ncol=A)

# LECTURA DE DATOS
X <- matrix(scan("C:\\MI TESIS\\Datos\\LINNX.PRN"),ncol=m,byrow=T)
Y <- matrix(scan("C:\\MI TESIS\\Datos\\LINNY.PRN"),ncol=p,byrow=T)

# ESCALAMIENTO DE VARIABLES
E <- scale(X)
FF <- scale(Y)

# PLS
for (i in 1:A)
```

```
{
 #inicialización de t y u
 t <- E[,1,drop=F]
 u <- FF[,1,drop=F]

 for (j in 1:100)
   {

    # en el block X
    wt <- t(u)%*%E/vecnorm(u)^2
    wt <- wt/vecnorm(wt)
    tnew <- E%*%t(wt)/vecnorm(wt)^2

    # en el block Y
    qt <- t(tnew)%*%FF/vecnorm(tnew)^2
    qt <- qt/vecnorm(qt)
    u <- FF%*%t(qt)/vecnorm(qt)^2

    #checar convergencia
    dif <- tnew-t
    c <- vecnorm(dif)
    if (c < 0.0000005) break else t <- tnew
    }

 # cálculo de las cargas de E y reescalamiento de los scores y pesos
 pt <- t(tnew)%*%E/vecnorm(tnew)^2
 normapt <- vecnorm(pt)
 pt <- pt/normapt
 tnew <- normapt*tnew
 wt <- normapt*wt

 # salvando los vectores de interés
 Mxpt[i,] <- pt
 Mxqt[i,] <- qt
 Mxwt[i,] <- wt
 Mxt[,i] <- tnew
 Mxu[,i] <- u

 # encontrando los coeficientes de regresión b
 b <- crossprod(u,tnew)/vecnorm(tnew)^2
 Mxb[i,i] <- b[1]

 # cálculo de residuales
 E <- E-tnew%*%pt
 FF <- FF-b[1]*tnew%*%qt
 }

#SELECCION DE OBSERVACIONES PARA REFERENCIA
for(k in 1:(A-1))
{ss <- k+1
for(kk in ss:A)
{vm <- paste("IDENTIFICANDO OUTLIERS EN t",k, "Y t",kk)
vx <- paste("Score  t",k)
vy <- paste("Score  t",kk)
plot(Mxt[,k],Mxt[,kk],main=vm,xlab=vx,ylab=vy)
identify(Mxt[,k],Mxt[,kk]) #permite identificar outliers
locator() #me da las coordenadas del punto que señale en el plot
}}

detach("C:\\MI TESIS\\Espacios de trabajo\\eliminacion de outliers",pos=1)
```

6) Gráficos de control

```
attach("C:\\MI TESIS\\Espacios de trabajo\\graficos de control",pos=1)

#ALGORITMO DEL TUTORIAL

#INICIALIZACION DE VARIABLES
m <- 3 # DE VARS. INDEP.(x's)
p <- 3 # DE VARS. DEP.(y's)
n <- 20 # DE MUESTRAS EN EL CONJUNTO DE CALIBRACION
A <- 3 # DE DIMENSIONES PLS

# LECTURA DE DATOS
X <- read.table("C:\\MI TESIS\\Datos\\LINNXf.PRN",header=T)
Y <- read.table("C:\\MI TESIS\\Datos\\LINNYf.PRN",header=T)
X <- data.matrix(X)
Y <- data.matrix(Y)

#CREADO DE RECIPIENTES
Mxpt <- matrix(rep(0,A*m),ncol=m)
Mxqt <- matrix(rep(0,A*p),ncol=p)
Mxwt <- matrix(rep(0,A*m),ncol=m)
Mxt <- matrix(rep(0,n*A),ncol=A)
Mxu <- matrix(rep(0,n*A),ncol=A)
Mxb <- matrix(rep(0,A*A),ncol=A)

# ESCALAMIENTO DE VARIABLES
E <- scale(X)
FF <- scale(Y)

#GUARDANDO LOS VALORES DEL ESCALAMIENTO
VeccentX <- matrix(rep(0,m),ncol=m)
VecescX <- matrix(rep(0,m),ncol=m)
VeccentY <- matrix(rep(0,p),ncol=p)
VecescY <- matrix(rep(0,p),ncol=p)

for (i in 1:m)
 {VeccentX[1,i] <- mean(X[,i])
  VecescX[1,i] <- sqrt(var(X[,i])) }
for (i in 1:p)
 {VeccentY[1,i] <- mean(Y[,i])
  VecescY[1,i] <- sqrt(var(Y[,i])) }


# PLS
if( p > 1 )
for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

  for (j in 1:100)
    {

     # en el block X
     wt <- t(u)%*%E/vecnorm(u)^2
     wt <- wt/vecnorm(wt)
     tnew <- E%*%t(wt)/vecnorm(wt)^2
```

```
    # en el block Y
    qt <- t(tnew)%*%FF/vecnorm(tnew)^2
    qt <- qt/vecnorm(qt)
    u <- FF%*%t(qt)/vecnorm(qt)^2

    #checar convergencia
    dif <- tnew-t
    c <- vecnorm(dif)
    if (c < 0.0000005) break else t <- tnew
    }

  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew%*%qt
  }

#PARA CUANDO HAY UNA SOLA VARIABLE Y
else

for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

      # en el block X
    wt <- t(u)%*%E/vecnorm(u)^2
    wt <- wt/vecnorm(wt)
    tnew <- E%*%t(wt)/vecnorm(wt)^2

    # en el block Y
    qt <- 1


  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
```

```
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew*qt
 }

#LIMITE PARA SPE(X)
MxtCALC <- scale(X)%*%t(Mxwt)
predX <- MxtCALC%*%Mxpt
MxDif <- (scale(X)-predX)^2
ErrPredX <- MxDif%*%matrix(rep(1,m),ncol=1)
mErrPredX <- mean(ErrPredX)
vErrPredX <- var(ErrPredX)
limSPE95 <- vErrPredX/(2*mErrPredX)*qchisq(.95,(2*mErrPredX)^2/vErrPredX)
limSPE99 <- vErrPredX/(2*mErrPredX)*qchisq(.99,(2*mErrPredX)^2/vErrPredX)
plot(ErrPredX,main="ERROR DE PREDICCION",xlab="No. de
Observación",ylab="SPE",xlim=c(0,25),ylim=c(0,17),col=1)
abline(limSPE95,0,lty=1,col=8)
abline(limSPE99,0,lty=1,col=8,lty=2)


#CONSTRUCCION DE ELIPSES DE CONFIANZA
c95 <- (A*(n^2-1)/(n*(n-A)))*qf(0.95,A,(n-A))
c99 <- (A*(n^2-1)/(n*(n-A)))*qf(0.99,A,(n-A))
SEJES95 <- matrix(rep(0,A),ncol=A)
SEJES99 <- matrix(rep(0,A),ncol=A)
for(i in 1:A)
 { SEJES95[i] <- sqrt(c95*var(MxtCALC[,i]))
   SEJES99[i] <- sqrt(c99*var(MxtCALC[,i])) }

for(k in 1:(A-1))
{ss <- k+1
for(kk in ss:A)
{vm <- paste("ELIPSE DE CONFIANZA DE  t",k, "vs  t",kk)
vx <- paste("Score  t",k)
vy <- paste("Score  t",kk)
sequen95 <- seq(-SEJES95[1,k],SEJES95[1,k],.005)
formulaelip95 <- sqrt(SEJES95[1,kk]^2*(1-sequen95^2/SEJES95[1,k]^2))
sequen99 <- seq(-SEJES99[1,k],SEJES99[1,k],.005)
formulaelip99 <- sqrt(SEJES99[1,kk]^2*(1-sequen99^2/SEJES99[1,k]^2))
plot(MxtCALC[ ,k],MxtCALC[ ,kk],main=vm,xlab=vx,ylab=vy,
xlim=c(-2*SEJES99[1,k],2*SEJES99[1,k]),ylim=c(-
   2*SEJES99[1,kk],2*SEJES99[1,kk]),col=1)
points(sequen95,1*formulaelip95,type="l",col=8)
points(sequen95,-1*formulaelip95,type="l",col=8)
points(sequen99,1*formulaelip99,type="l",col=8,lty=2)
points(sequen99,-1*formulaelip99,type="l",col=8,lty=2)
}}

#LIMITES DE CONTROL PARA SCORES INDIVIDUALES
LimTS95 <- matrix(rep(0,A),ncol=A)
LimTS99 <- matrix(rep(0,A),ncol=A)
for(i in 1:A)
 { LimTS95[,i] <- qt(0.975,n-1)*sqrt(var(MxtCALC[,i]))*sqrt(1+1/n)
```

```
   LimTS99[,i] <- qt(0.995,n-1)*sqrt(var(MxtCALC[,i]))*sqrt(1+1/n) }
LimTI95 <- -1*LimTS95
LimTI99 <- -1*LimTS99
for(i in 1:A){
vm <- paste("GRAFICA DE CONTROL INDIVIDUAL PARA  t",i)
vy <- paste("Score  t",i)
yinf <- 3*LimTI99[1,i]
ysup <- 3*LimTS99[1,i]
plot(MxtCALC[,i],main=vm,xlab="No. de
   Observación",ylab=vy,xlim=c(0,(n+5)),ylim=c(yinf,ysup),col=1)
abline(LimTS95[1,i],0,lty=1,col=8)
abline(LimTI95[1,i],0,lty=1,col=8)
abline(LimTS99[1,i],0,col=8,lty=2)
abline(LimTI99[1,i],0,col=8,lty=2) }
```

```
#MONITOREO DE UNA NUEVA CORRIDA(OBSERVACION)

Xnew<-matrix(c(154,33,56),ncol=3)#nueva observacion a monitorear

Xnewesc <- (Xnew-VeccentX)/VecescX
MxtCALCn <- Xnewesc%*%t(Mxwt)
predXne <- MxtCALCn%*%Mxpt
MxDifXne <- Xnewesc-predXne
MxDifXne2 <- (Xnewesc-predXne)^2
ErrPredXne <- sum(MxDifXne2)

# MONITOREO GRAFICA SPE
plot(ErrPredX,main="ERROR DE PREDICCION",xlab="No. de
Observación",ylab="SPE",xlim=c(0,25),ylim=c(0,17))
abline(limSPE95,0,lty=1,col=8)
abline(limSPE99,0,lty=1,col=8,lty=2)
points(23,ErrPredXne,pch=9,cex=1,col=17)

#MONITOREO SCORES(ELIPSES)

for(k in 1:(A-1))
{ss <- k+1
for(kk in ss:A)
{vm <- paste("ELIPSE DE CONFIANZA DE  t",k, "vs  t",kk)
vx <- paste("Score  t",k)
vy <- paste("Score  t",kk)
sequen95 <- seq(-SEJES95[1,k],SEJES95[1,k],.005)
formulaelip95 <- sqrt(SEJES95[1,kk]^2*(1-sequen95^2/SEJES95[1,k]^2))
sequen99 <- seq(-SEJES99[1,k],SEJES99[1,k],.005)
formulaelip99 <- sqrt(SEJES99[1,kk]^2*(1-sequen99^2/SEJES99[1,k]^2))
plot(MxtCALC[ ,k],MxtCALC[ ,kk],main=vm,xlab=vx,ylab=vy,
xlim=c(-2*SEJES99[1,k],2*SEJES99[1,k]),ylim=c(-
   2*SEJES99[1,kk],2*SEJES99[1,kk]),col=1)
points(sequen95,1*formulaelip95,type="l",col=8)
points(sequen95,-1*formulaelip95,type="l",col=8)
points(sequen99,1*formulaelip99,type="l",col=8,lty=2)
points(sequen99,-1*formulaelip99,type="l",col=8,lty=2)
points(MxtCALCn[1,k],MxtCALCn[1,kk],pch=9,cex=1,col=17)
}}

#MONITOREO SCORES(INDIVIDUALES)
for(i in 1:A){
vm <- paste("GRAFICA DE CONTROL INDIVIDUAL PARA  t",i)
vy <- paste("Score  t",i)
yinf <- 3*LimTI99[1,i]
ysup <- 3*LimTS99[1,i]
```

```
plot(MxtCALC[,i],main=vm,xlab="No. de
   Observación",ylab=vy,xlim=c(0,(n+5)),ylim=c(yinf,ysup),col=1)
abline(LimTS95[1,i],0,lty=1,col=8)
abline(LimTI95[1,i],0,lty=1,col=8)
abline(LimTS99[1,i],0,col=8,lty=2)
abline(LimTI99[1,i],0,col=8,lty=2)
points(23,MxtCALCn[1,i],pch=9,cex=1,col=17)
}

#GRAFICA DE DIAGNOSTICO
barplot(MxDifXne,names=dimnames(X)[[2]],main="CONTRIBUCION AL ERROR DE
   PREDICCION",
style="old",dbangle=45,density=10,ylab="ERROR DE PREDICCIÓN")

detach("C:\\MI TESIS\\Espacios de trabajo\\graficos de control",pos=1)
```

7) Intervalos de confianza

```
attach("C:\\MI TESIS\\Espacios de trabajo\\intervalos de confianza",pos=1)

#ALGORITMO DEL PAPER 5

#INICIALIZACION DE VARIABLES
m <- 3 # DE VARS. INDEP.(x's)
p <- 1 # DE VARS. DEP.(y's)#debe ser 1 sin importar el No. de vars. ya que solo
   se pueden obtener IC para 1 var. a la vez
n <- 20 # DE MUESTRAS EN EL CONJUNTO DE CALIBRACION
A <- 3 # DE DIMENSIONES PLS

# LECTURA DE DATOS
X <- read.table("C:\\MI TESIS\\Datos\\LINNXf.PRN",header=T)
Y <- read.table("C:\\MI TESIS\\Datos\\LINNYf.PRN",header=T)
X <- data.matrix(X)
Y <- data.matrix(Y)

#DEFINIR AQUI LA VARIABLE A CONSIDERAR
Y <- Y[ ,2,drop=F] #Aqui debe especificarse el lugar de la variable sobre la que
   se hara prediccion

#CREADO DE RECIPIENTES
Mxpt <- matrix(rep(0,A*m),ncol=m)
Mxqt <- matrix(rep(0,A*p),ncol=p)
Mxwt <- matrix(rep(0,A*m),ncol=m)
Mxt <- matrix(rep(0,n*A),ncol=A)
Mxu <- matrix(rep(0,n*A),ncol=A)
Mxb <- matrix(rep(0,A*A),ncol=A)
VeccentX <- matrix(rep(0,m),ncol=m)
VecescX <- matrix(rep(0,m),ncol=m)
VeccentY <- matrix(rep(0,p),ncol=p)
VecescY <- matrix(rep(0,p),ncol=p)

# ESCALAMIENTO DE VARIABLES
E <- scale(X)
FF <- scale(Y)

#GUARDANDO LOS VALORES DEL ESCALAMIENTO
for (i in 1:m)
 {VeccentX[1,i] <- mean(X[,i])
  VecescX[1,i] <- sqrt(var(X[,i])) }
```

```
for (i in 1:p)
 {VeccentY[1,i] <- mean(Y[,i])
  VecescY[1,i] <- sqrt(var(Y[,i])) }

# PLS

for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

  for (j in 1:100)
    {

     # en el block X
     wt <- t(u)%*%E/vecnorm(u)^2
     wt <- wt/vecnorm(wt)
     tnew <- E%*%t(wt)/vecnorm(wt)^2

     # en el block Y
     qt <- t(tnew)%*%FF/vecnorm(tnew)^2
     #####qt <- qt/vecnorm(qt)
     u <- FF%*%t(qt)/vecnorm(qt)^2

     #checar convergencia
     dif <- tnew-t
     c <- vecnorm(dif)
     if (c < 0.0000005) break else t <- tnew
     }

  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  ####normapt <- vecnorm(pt)
  ####pt <- pt/normapt
  ####tnew <- normapt*tnew
  ####wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew%*%qt
  }

#PREDICCION E INTERVALOS DE CONFIANZA

# OBTENCION DE X y Y PREDICHOS
LICnn <- matrix(rep(0,n),ncol=1)
LSCnn <- matrix(rep(0,n),ncol=1)
TERM <- matrix(rep(0,n),ncol=1)
CONT <- 0
Yprednn <- matrix(nrow=n,ncol=1)
```

```
FACTOR <- t(Mxwt)%*%ginverse(Mxpt%*%t(Mxwt))
Xpred <- Mxt%*%Mxpt
Ypred <- Mxt%*%Mxqt

#INTERVALOS DE CONFIANZA PARA LAS PREDICCIONES EN Y
ErrorY <- scale(Y)-(scale(X)%*%FACTOR)%*%Mxqt
SSE <- vecnorm(ErrorY)^2
MSE <- SSE/(n-A-1)
tstud <- qt(0.975,n-A-1)
MxtINV <- ginverse(t(Mxt)%*%Mxt)
for(i in 1:n)
{TERM[i, ] <- sqrt(1+Mxt[i, ,drop=F]%*%MxtINV%*%t(Mxt[i, ,drop=F]))}
LIC <- Ypred - tstud*sqrt(MSE)*TERM
LSC <- Ypred + tstud*sqrt(MSE)*TERM

#regresando a las unidades originales
for(i in 1:n)
  { LICnn[i, ] <- LIC[i, ,drop=F]*VecescY+VeccentY
    LSCnn[i, ] <- LSC[i, ,drop=F]*VecescY+VeccentY
    Yprednn[i,] <- Ypred[i, ,drop=F]*VecescY+VeccentY }

YRYP <- cbind(Y,Yprednn)
dimnames(YRYP) <- list(paste("Observacion",1:n),c("Y real","Y predicho"))
LIYLS <- cbind(LICnn,Y,LSCnn)
dimnames(LIYLS) <- list(paste("Observación",1:n),c("L.INF.","REAL","LIM.SUP."))

#conteo del No. de intervalos que no contienen al valor real
for(i in 1:n)
  { if (Y[i,1]<LICnn[i,1] || Y[i,1]>LSCnn[i,1]) CONT <- CONT+1 }
PORCNC <- (CONT/n)*100
names(PORCNC) <- "PORCENTAJE DE INTERVALOS DE CONFIANZA QUE NO CONTIENEN A SU
   VALOR REAL:"
print(YRYP)
print(LIYLS)
print(PORCNC)
-------------------------------------------------------------------------------
#PREDICCION E INTERVALOS DE CONFIANZA PARA UNA NUEVA OBSERVACION

# OBTENCION DE X y Y PREDICHOS

Xnew <- matrix(c(189,35,46),ncol=m) #ESPECIFICAR LA NUEVA OBSERVACION(puede ser
   mas de una)

No.f <- dim(Xnew)[1]
Xnewesc <- matrix(rep(0,m*No.f),ncol=m)
Yprednn <- matrix(rep(0,No.f),ncol=1)
LICnn <- matrix(rep(0,No.f),ncol=1)
LSCnn <- matrix(rep(0,No.f),ncol=1)
TERM <- matrix(rep(0,No.f),ncol=1)
CONT <- 0

for(i in 1:No.f)
{Xnewesc[i, ] <- (Xnew[i, ,drop=F]-VeccentX)/VecescX}
Tnew <- Xnewesc%*%FACTOR
Xpred <- Tnew%*%Mxpt
Ypred <- Tnew%*%Mxqt
for(i in 1:No.f)
{Yprednn[i,] <- Ypred[i, ,drop=F]*VecescY+VeccentY}

#INTERVALOS DE CONFIANZA PARA LAS PREDICCIONES EN Y
for(i in 1:No.f)
```

```
{TERM[i, ] <- sqrt(1+Tnew[i, ,drop=F]%*%MxtINV%*%t(Tnew[i, ,drop=F]))}
LIC <- Ypred - tstud*sqrt(MSE)*TERM
LSC <- Ypred + tstud*sqrt(MSE)*TERM

#regresando a las unidades originales
for(i in 1:No.f)
{LICnn[i, ] <- LIC[i, ,drop=F]*VecescY+VeccentY
LSCnn[i, ] <- LSC[i, ,drop=F]*VecescY+VeccentY}

LIYLS <- cbind(LICnn,Yprednn,LSCnn)
dimnames(LIYLS) <-
   list(paste("Observación",1:No.f),c("L.INF.","PREDICHO","LIM.SUP."))

print(LIYLS)


detach("C:\\MI TESIS\\Espacios de trabajo\\intervalos de confianza",pos=1)
```

8) Variabilidad explicada

```
attach("C:\\MI TESIS\\Espacios de trabajo\\variabilidad explicada",pos=1)

#ALGORITMO DEL TUTORIAL

#INICIALIZACION DE VARIABLES
m <- 3 # DE VARS. INDEP.(x's)
p <- 3 # DE VARS. DEP.(y's)
n <- 20 # DE MUESTRAS EN EL CONJUNTO DE CALIBRACION
A <- 3 # DE DIMENSIONES PLS

# LECTURA DE DATOS
X <- read.table("C:\\MI TESIS\\Datos\\LINNXf.PRN",header=T)
Y <- read.table("C:\\MI TESIS\\Datos\\LINNYf.PRN",header=T)
X <- data.matrix(X)
Y <- data.matrix(Y)

#CREADO DE RECIPIENTES
Mxpt <- matrix(rep(0,A*m),ncol=m)
Mxqt <- matrix(rep(0,A*p),ncol=p)
Mxwt <- matrix(rep(0,A*m),ncol=m)
Mxt <- matrix(rep(0,n*A),ncol=A)
Mxu <- matrix(rep(0,n*A),ncol=A)
Mxb <- matrix(rep(0,A*A),ncol=A)
VeccentX <- matrix(rep(0,m),ncol=m)
VecescX <- matrix(rep(0,m),ncol=m)
VeccentY <- matrix(rep(0,p),ncol=p)
VecescY <- matrix(rep(0,p),ncol=p)
POREXPXT <- matrix(rep(0,A),ncol=A)
POREXPYT <- matrix(rep(0,A),ncol=A)
POREXPXI <- matrix(rep(0,A*m),ncol=A)
POREXPYI <- matrix(rep(0,A*p),ncol=A)

# ESCALAMIENTO DE VARIABLES
E <- scale(X)
FF <- scale(Y)

#GUARDANDO LOS VALORES DEL ESCALAMIENTO
for (i in 1:m)
 {VeccentX[1,i] <- mean(X[,i])
```

```
  VecescX[1,i] <- sqrt(var(X[,i])) }
for (i in 1:p)
 {VeccentY[1,i] <- mean(Y[,i])
  VecescY[1,i] <- sqrt(var(Y[,i])) }

# PLS
if( p > 1 ) {
for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

  for (j in 1:100)
    {

     # en el block X
     wt <- t(u)%*%E/vecnorm(u)^2
     wt <- wt/vecnorm(wt)
     tnew <- E%*%t(wt)/vecnorm(wt)^2

     # en el block Y
     qt <- t(tnew)%*%FF/vecnorm(tnew)^2
     qt <- qt/vecnorm(qt)
     u <- FF%*%t(qt)/vecnorm(qt)^2

     #checar convergencia
     dif <- tnew-t
     c <- vecnorm(dif)
     if (c < 0.0000005) break else t <- tnew
     }

  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u

  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew%*%qt
  POREXPXT[i] <- 100-(100/(m*(n-1)))*sum(E^2)
  POREXPYT[i] <- 100-(100/(p*(n-1)))*sum(FF^2)
  for(j in 1:m)
    { POREXPXI[j,i] <- 100-(100/(n-1))*sum(E[ ,j]^2) }
  for(j in 1:p)
    { POREXPYI[j,i] <- 100-(100/(n-1))*sum(FF[ ,j]^2) }
  }

POREXPX <- rbind(POREXPXI,POREXPXT)
```

```
POREXPY <- rbind(POREXPYI,POREXPYT)
dimnames(POREXPX) <-
   list(c(dimnames(X)[[2]],"TotalX"),paste(c("Cum%ExpDim"),1:A))
dimnames(POREXPY) <-
   list(c(dimnames(Y)[[2]],"TotalY"),paste(c("Cum%ExpDim"),1:A))
POREXPXNC <- matrix(nrow=(m+1),ncol=A)

for (j in 1:(A-1))
  { for(i in 1:(m+1))
     { POREXPXNC[i,(A+1-j)] <- POREXPX[i,(A+1-j)]-POREXPX[i,(A-j)] } }

POREXPXNC[,1] <- POREXPX[,1]
POREXPYNC <- matrix(nrow=(p+1),ncol=A)

for (j in 1:(A-1))
  { for(i in 1:(p+1))
    { POREXPYNC[i,(A+1-j)] <- POREXPY[i,(A+1-j)]-POREXPY[i,(A-j)] } }

POREXPYNC[,1] <- POREXPY[,1]

barplot(t(POREXPXNC),main="CONTRIBUCION DE LAS VARIABLES X A CADA
DIMENSION",xlab="VARIABLES",names=dimnames(POREXPX)[1],ylab="
%VAR.EXPLICADO")

barplot(t(POREXPYNC),main="CONTRIBUCION DE LAS VARIABLES Y A CADA
DIMENSION",xlab="VARIABLES",names=dimnames(POREXPY)[1],ylab=
"%VAR.EXPLICADO")

RESUL <- list(VariabilidadX=POREXPX,VariabilidadY=POREXPY)
print(RESUL)
}

else {
#PARA CUANDO HAY UNA SOLA VARIABLE Y
for (i in 1:A)
 {
  #inicialización de t y u
  t <- E[,1,drop=F]
  u <- FF[,1,drop=F]

       # en el block X
     wt <- t(u)%*%E/vecnorm(u)^2
     wt <- wt/vecnorm(wt)
     tnew <- E%*%t(wt)/vecnorm(wt)^2

     # en el block Y
     qt <- 1


  # cálculo de las cargas de E y reescalamiento de los scores y pesos
  pt <- t(tnew)%*%E/vecnorm(tnew)^2
  normapt <- vecnorm(pt)
  pt <- pt/normapt
  tnew <- normapt*tnew
  wt <- normapt*wt

  # salvando los vectores de interés
  Mxpt[i,] <- pt
  Mxqt[i,] <- qt
  Mxwt[i,] <- wt
  Mxt[,i] <- tnew
  Mxu[,i] <- u
```

```
  # encontrando los coeficientes de regresión b
  b <- crossprod(u,tnew)/vecnorm(tnew)^2
  Mxb[i,i] <- b[1]

  # cálculo de residuales
  E <- E-tnew%*%pt
  FF <- FF-b[1]*tnew*qt
POREXPXT[i] <- 100-(100/(m*(n-1)))*sum(E^2)
POREXPYT[i] <- 100-(100/(p*(n-1)))*sum(FF^2)

for(j in 1:m)
  { POREXPXI[j,i] <- 100-(100/(n-1))*sum(E[ ,j]^2) }

for(j in 1:p)
  { POREXPYI[j,i] <- 100-(100/(n-1))*sum(FF[ ,j]^2) }
                        }

POREXPX <- rbind(POREXPXI,POREXPXT)
POREXPY <- rbind(POREXPYI,POREXPYT)
dimnames(POREXPX) <-
   list(c(dimnames(X)[[2]],"TotalX"),paste(c("Cum%ExpDim"),1:A))
dimnames(POREXPY) <-
   list(c(dimnames(Y)[[2]],"TotalY"),paste(c("Cum%ExpDim"),1:A))
POREXPXNC <- matrix(nrow=(m+1),ncol=A)

for (j in 1:(A-1))
  { for(i in 1:(m+1))
     { POREXPXNC[i,(A+1-j)] <- POREXPX[i,(A+1-j)]-POREXPX[i,(A-j)] } }

POREXPXNC[,1] <- POREXPX[,1]
POREXPYNC <- matrix(nrow=(p+1),ncol=A)

for (j in 1:(A-1))
  { for(i in 1:(p+1))
    { POREXPYNC[i,(A+1-j)] <- POREXPY[i,(A+1-j)]-POREXPY[i,(A-j)] } }

POREXPYNC[,1] <- POREXPY[,1]
b
barplot(t(POREXPXNC),main="CONTRIBUCION DE LAS VARIABLES X A CADA
DIMENSION",xlab="VARIABLES",names=dimnames(POREXPX)[1],ylab="
%VAR.EXPLICADO")

barplot(t(POREXPYNC),main="CONTRIBUCION DE LAS VARIABLES Y A CADA
DIMENSION",xlab="VARIABLES",names=dimnames(POREXPY)[1],ylab=
"%VAR.EXPLICADO")

RESUL <- list(VariabilidadX=POREXPX,VariabilidadY=POREXPY)
print(RESUL)
}

detach("C:\\MI TESIS\\Espacios de trabajo\\variabilidad explicada",pos=1)
```

4.2 Datos utilizados

Se presentan en esta parte, los datos utilizados para probar las rutinas s-plus y para generar los resultados y gráficas discutidos. Los conjuntos de datos

numerados como UNO, DOS, etc., son simulados para evidenciar las características relevantes de las técnicas (pero las rutinas programadas funcionan con cualesquiera otros). Los datos LINNX y LINNY. Son datos reales que se presentan en [1].

UNOX

10	10.7
10.4	9.8
9.7	10
9.7	10.1
11.7	11.5
11	10.8
8.7	8.8
9.5	9.3
10.1	9.4
9.6	9.6
10.5	10.4
9.2	9
11.3	11.6
10.1	9.8
8.5	9.2

DOSX

12	8
15	8
13	7
11	8
9	7
11	8
15	9
16	8
8	6
12	7
10	7
13	9
15	7
11	7
15	8
16	9
7	4
8	4
13	7
11	8

DOSY

87	150
85	151
47	172
68	147
77	127
59	162
76	143
58	163
77	157
55	136
49	141
75	167
90	132
95	142
69	137
67	137
95	173
77	157
69	133
62	168

LINNX

191	36	50
189	37	52
193	38	58
162	35	62
189	35	46
182	36	56
211	38	56
167	34	60
176	31	74
154	33	56
169	34	50
166	33	52
154	34	64
247	46	50
193	36	46
202	37	62
176	37	54
157	32	52
156	33	54
138	33	68

LINNY

5	162	60
2	110	60
12	101	101
12	105	37
13	155	58
4	101	42
8	101	38
6	125	40
15	200	40
17	251	250
17	120	38
13	210	115
14	215	105
1	50	50
6	70	31
12	210	120
4	60	25
11	230	80
15	225	73
2	110	43

TRESX

5	3
5	4
6	5
6	4
7	5
7	6
8	7
8	6
9	8
9	7
10	9
10	8
11	10
11	9
12	11
12	10
13	12
13	11
14	13
14	12
15	13
15	14

CUATROX	
11	16
13	17
8	18
4	14
7	2
12	6
16	2
4	7
8	2
9	6
13	17
18	5
15	7
7	17
3	8

CUATROY	
125	136
132	100
100	141
167	144
143	132
165	131
120	102
110	145
152	131
111	134
162	121
158	145
176	153
118	141
132	107

5. Bibliografía.

[1] Jackson, Edward (1991). *A user's guide to principal components*. John Wiley & Sons. E.U.A.

[2] Johnson, Richard; Wichern, Dean. (1982). *Applied Multivariate Statistical Analysis*. Prentice Hall, New Yersey

[3] Rawlings, John. (1988). *Applied regression analysis: a research tool.* Wadsworth & Brooks, California.

[4] Venables, W; Ripley, B. (1996). *Modern Applied Statistics with S-PLUS.* Springer Verlag, New York.

[5] Mason, Robert; Young, Jihn (2002). *Multivariate Statistical Process Control with Industrial Applications*. ASA-SIAM, Pennsylvania.

[6] W. Glen; W. Dunn III. (1989). *Principal Component Analysis and Partial Least Squares Regression.* Tetrahedrom Computer Methodolgy, Vol. 2, No. 6 pp 349 a 376, Great Britain.

[7] Geladi, Paul; Kowalski, Bruce. (1986*). Partial Least Squares Regression: a tutorial.* Analytica Chimica Acta, 185 pp 1-17, Netherlands.

[8] Geladi, Paul; Kowalski, Bruce (1986*). An example of 2-block predictive partial least-sqares regressión with simulated data.* Analytica Chimica Acta, 185 pp 19-32, Netherlands.

[9] Helland, Inge. (1988). *On the structure of Partial Least Squares regressio*n. Communications on Statistics-Simulation; Vol 17, No. 2, pp 581-607, Norway.

[10] Kresta, James; MacGregor, John. (1991). *Multivariate Satistical Montoring of Process Operating Performance.* The Canadian Journal of Chemical Engineering, Vol 69 pp 35-47.

[11] MacGregor, John; Jaeckle, Christiane. (1994). *Process Monitoring and Diagnosis by Multiblock Methods*. American Industrial Chemical Engineering, Vol. 40, No. 5 pp 826-838.

[12] Nomikos, Paul; MacGregor, John. (1994). *Monitoring batch process using multiway principal components analysis*. American Industrial Chemical Engineering, Vol. 40, No. 8 pp 1361-1375.

[13] Nomikos, Paul; MacGregor, John. (1995). *Multivariate SPC charts for monitoring batch process*. Technometrics, Vol. 37, No. 1 pp 41-59.

[14] Nomikos, Paul; MacGregor, John. (1995). *Multi-way partial least squares in monitoring batch processes*. Chemometrics and intelligent laboratory systems. Vol. 30, pp 97-108.

[15] Woodhall, W; M., Ncube. (1985). *Multivariate CUSUM Quality Control Procedures*. Technometrics, Vol. 27, pp 285-292.

[16] Jackson, J. (1980). *Principal Components and Factor Analysis*: Part 1. Journal of Quality Technology. Vol. 12, pp 201-213.

[17] Hoerl, A. E.; Kennard, R. W. (1970). *Ridge Regression: Biased Estimation for Non-orthogonal Problems*. Technometrics, Vol. 12, pp 55-67.