UNIVERSIDAD DE SEVILLA

Modelo de Regresión PLS

Trabajo Fin de Grado - Grado en Estadística

Departamento de Estadística e Investigación Operativa.

Facultad de Matemáticas

Junio 2017

TRABAJO REALIZADO POR: CRISTINA MÁRQUEZ RUIZ

TUTOR: RAFAEL PINO MEJÍAS

ABSTRACT.

Partial least Square (PLS) methods relate the information present in two data tables that collect measurements on the same set of observations. PLS methods proceed by deriving latent variables which are (optimal) linear combinations of the variables of a data table. When the goal is to find the shared information between two tables, the approach is equivalent to a correlation problem and the technique is then called Partial Least Square correlation. In this case there are two sets of latent variables (one set per table), and these latent variables are required to have maximal covariance. When the goal is to predict one data table the other one, the technique is then called Partial Least Square regression. In this case there is one set of latent variables (derived from the predictor table) and these latent variables are required to give the best posible prediction.

INDICE

CAPITULO 1: BREVE CRONOLOGÍA HISTORICA

Los procedimientos de mínimos cuadrados parciales son desarrollados inicialmente por el estadístico sueco Herman Wold. Nació el 25 de diciembre de 1908 en la pequeña ciudad de Skien al sur de Oslo. Allí pasó sus primeros tres años de vida hasta 1912 cuando sus padres decidieron mudarse a Suecia. A temprana edad, Herman mostró un buen talento para las matemáticas, y después de la secundaria se matriculó en la Universidad de Estocolmo en 1927, donde estudió Física, Matemáticas y Economía. Allí se encontró con Harald Cramér, profesor sueco de Matemáticas y Estadística. Wold estaba interesado solo en la Estadística y decidió permanecer baja la tutela de Cramér y aprender acerca de los Elementos de Probabilidad, Estadística y Teoría del Riesgo. En 1930 Herman se graduó y encontró su primer trabajo en el sector de los seguros donde comenzó a hacer trabajo actuarial.

El interés de Herman en la Estadística fue mayor y más profundo que su intención de permanecer en el campo actuarial, y decidió volver a la universidad y obtener un título de doctorado. Nuevamente bajo la tutoría de Harald Cramér, realizó cursos sobre procesos estocásticos y series temporales. Por otra parte, empezó a interesarse por la Teoría de Probabilidades introducida poco antes por el famoso matemático Andrei Kolmogorov.

En 1938 Wold se doctoró con una tesis sobre *el análisis de series temporales estacionarias*. Sus primeras aportaciones fueron el estudio de la predicción en un solo paso de una serie temporal, y el *teorema de descomposición*, una de los más famosos resultados por Wold. Después de sus estudios de doctorado, Herman permaneció en la Universidad de Estocolmo como profesor en matemáticas actuariales y Estadística Matemática. En 1942 Wold obtuvo su Cátedra de Estadística en la prestigiosa Universidad de Uppsala, la universidad más antigua en Suecia.

En el año 1966 apareció el trabajo de Herman Wold en el que se presenta por primera vez lo que se conoce actualmente como Partial Least Squares (PLS) o regresión de mínimos cuadrados parciales.

A los primeros artículos le seguirían otros donde se elaboró más la técnica y, con posterioridad los trabajos fueron continuados por su hijo Svante Wold acompañado por un grupo de especialistas noruegos entre los que se pueden señalar de manera especial a H. Martens y T. Naes.

A partir de las propuestas de Herman Wold en la década de 1960, los métodos de mínimos cuadrados parciales han desarrollado un largo camino hasta llegar a la actualidad.

Históricamente, el primer tipo de algoritmo PLS es un método alternativo para el cálculo de componentes principales. Este procedimiento se extendió casi inmediatamente a una serie de procedimientos entre los que había una versión para el cálculo de correlaciones canónicas. Bajo el nombre de Niles, abreviatura de "no lineal por mínimos cuadrados iterativos" (Wold 1966) presenta un collage de ejemplos resueltos mediante procedimientos iterativos basados en modelos de regresiones de mínimos cuadrados

Curiosamente, estos trabajos iniciales contenían los elementos matemáticos fundamentales de todos los métodos PLS siguientes: cálculo de las componentes de datos como sumas ponderadas de variables, obtenidos operacionalmente a través de los pasos de regresiones de mínimos cuadrados. No mucho tiempo después de su presentación, se sustituyó el término "Niles" por "NIPALS" (no lineal Iterative Partial Least Squares), en consecuencia, cambiará de procedimientos Niles por procedimientos NIPALS. Debido a que estas primeras publicaciones fueron relacionadas con el cálculo de Análisis de Componentes Principales, hoy la mayoría de los autores se refieren a NIPALS como el algoritmo PLS para la PCA. Para evitar confusiones su correcta nomenclatura sería NIPALS-PCA.

Hacia finales de la década de 1970, el acrónimo "NIPALS" se acorta a "PLS" y se hizo hincapié en los modelos con variables latentes observadas indirectamente. A finales de esta década se presenta oficialmente el diseño básico para el modelado Path-PLS, lo que puede ser considerado como la versión estable. La primera publicación con todos los elementos del diseño básico es *"Causal-Predictive Analysis of Problems with High Complexity and Low Information: Recent Developments of Soft Modeling"* (Wold, 1979). Modelado suave es el nombre de la metodología para la estimación de modelos PLS con variables latentes observadas indirectamente por múltiples indicadores.

En la última de sus obras "*Theoretical Empiricism: A Rationale for Scientific Model-Building*"(Wold 1989) refleja y resume su punto de vista acerca de la construcción de modelos y modelado suave a través de enfoque PLS.

Herman Wold llevó a su equipo, en constante estudio detallado de sus procedimientos durante un largo período de tiempo; tomando forma y maduración, hasta que llegó una versión denominada "diseño básico", envuelto alrededor de la noción de Path Modeling. Wold da un enfoque de modelización para el análisis de sistemas de relaciones lineales con variables observadas y no observadas.

En 1971, Svante Wold, como joven profesor de la Universidad de Umea, Suecia, inventó la palabra *quimiometría:" Quimiometría, el arte de extraer la información relevante químicamente a partir de datos producidos en los experimentos químicos".*

A finales de 1970, como el marco de PLS Herman se hizo más maduro, Svante comenzó a mostrar cierto interés en la idea de variables latentes, y las oportunidades que parecía abrirse para el análisis de datos químicos de alta dimensión. El concepto variable latente era muy similar a los efectos que tenía en la química orgánica según Svante. Al principio no se tomó PLS muy en serio, pero gracias al entusiasmo que mostraba su padre se convenció finalmente de que la metodología PLS era un enfoque con gran potencial para trabajar.

Una vez comprendidos los conceptos básicos del marco PLS de Herman, Svante comenzó a trabajar con el modelo PLS más simple (dos bloques) en el comienzo de la década de 1980. Svante se reunió con el químico noruego Harald Martens en Oslo, en ese momento Harald estaba trabajando con modelos de predicción, utilizando regresión de Componentes Principales (PCR), lo cual no siempre proporcionaba buenos resultados. Juntos empezaron a aplicar dos bloques a los estudios de PLS de Herman; los primeros resultados no fueron lo que se esperaba, pero paso a paso descubrieron que tenían que hacer un par de ajustes en el algoritmo de Herman para el método de trabajo que resulto ser muy exitoso y prometedor.

Este es el comienzo del marco de la regresión PLS. El trabajo seminal *"El método de calibración multivariante en química por el método PLS*", corresponde a Svante Wold, Harald Martens y Herman Wold (Wold Martens Wold 1983).

En una industria en la que había una necesidad de herramientas de análisis capaces de hacer frente a la multicolinealidad, valores perdidos, y grandes conjuntos de datos, la regresión PLS cumple todas esas necesidades.

Al igual que Herman Wold en la econometría, su hijo Svante Wold se convirtió en una figura pionera y líder en su campo de especialización: quimiometría. A diferencia de su padre, la trayectoria de Svante sería tomar un camino diferente. Svante Wold y sus compañeros adaptaron y reformaron las ideas de Herman. Quitando cuestiones más teóricas, se centraron más en las cuestiones prácticas y aspectos computacionales. Si bien la evolución de Herman Wold se enmarca más dentro de una tradición de creación de modelos econométricos, el comportamiento de Svante y sus colaboradores fue más influenciado por la industria.

Herman falleció el 16 de febrero de 1992 en Estocolmo a los 83 años de edad. Había dejado un legado tremendo, y ya había pasado la antorcha a varios alumnos y colaboradores.

A comienzos de 1990, Svante Wold fue promotor de la regresión PLS dentro de las industrias químicas de todo el mundo.

Lo que hoy conocemos como mínimos cuadrados parciales, es el resultado de un largo período de evolución, con una amplia gama de métodos y técnicas propuestas desde finales de 1960. Vienen de diferentes disciplinas y campos de aplicación, motivados para resolver una serie de problemas relaciones multivariantes entre uno o más bloques de variables. Esto significa que los métodos PLS han crecido a lo largo de varias décadas, mutando progresivamente tanto en forma y contenido como la migración de un campo de estudio a otro.

A pesar de que los métodos de PLS se han construido sobre los cimientos de cálculo NIPALS de mediados de la década de 1960, históricamente e ideológicamente podemos distinguir dos ramas principales de mínimos cuadrados parciales: "*la ruta de modelado*" y la "*regresión*" ambos basados en obras presentadas originalmente por Herman Wold. El reconocimiento de estas dos grandes categorías tiene que ver con la forma en que se han desarrollado posteriormente. Al tomar diferentes direcciones, se han producido dos movimientos principales que, en su mayor parte, se han distanciado.

Aprovechando las ideas computacionales centrales de la metodología de su padre, Svante y sus colegas posteriormente pusieron en marcha una serie de algoritmos con énfasis en los problemas de regresión multivariante.

Según Gastón Sánchez (2015), los algoritmos de regresión PLS se desarrollaron a partir de la adaptación del algoritmo básico PLS Path Modeling (PLS-PM), la rama de regresión PLS tiene algunas diferencias contrastantes e incluso extremas. La notación cambió radicalmente cuando se introdujo la regresión PLS. Svante y sus colegas tomaron una decisión consciente de emplear una notación más de vector y matriz que mejora en gran medida la lectura de ecuaciones. Del mismo modo, la estructura general del algoritmo PLS-PM desaparece bajo la adaptación PLS regresión; simplifican en gran medida los pasos algorítmicos. Estos son quizás los rasgos que se destacan más e impide que el lector vea la regresión PLS como una versión ligeramente modificada del algoritmo PLS Path Modeling.

La forma y el estilo en el que H. Wold y S. Wold padre e hijo presentaron sus obran han dejado una huella profunda en sus desarrollos posteriores. El marco de Herman, surgió a partir de sistemas de ecuaciones econométricas con variables latentes, parece tener diferencias insalvables con los Modelos de Regresión de Svante surgido de la quimiometría. Ambas ramas, con sus distintas subdivisiones, muestran diferencias no solo en la zona de aplicación sino en leguaje, técnica, difusión; aunque sus elementos matemáticos y operativas son comunes.

Otra diferencia importante tiene que ver con las áreas de aplicación. Herman Wold aplicó sus métodos principalmente para aplicaciones económicas-sociológico. Svante Wold y sus colegas aplicaron sus métodos a los datos químicos e industrias relacionadas, que se ocupan de los problemas más pragmáticos y prácticos. Mientras que las aplicaciones Herman Wold eran de carácter más teórico, Svante era lo opuesto.

Si bien es cierto que tales diferencias no son despreciables, la mayoría de ellas están en el nivel de formato. A pesar de éstas, todavía hay fuertes lazos de similitudes. El denominador común más importante son los principios matemáticos y algorítmicos. Estos rasgos comunes pueden ser explotados para vincularlos de nuevo juntos. Afortunadamente, esta separación se ha reducido considerablemente en los últimos años, gracias a la organización de simposios PLS, y el trabajo activo de los investigadores que se han ocupado de llenar los vacíos existentes.

CAPITULO 2: INTRODUCCION

Partial Least Square (PLS) es un marco de modelado de datos multivariantes versátil para el análisis de múltiples relaciones entre uno o más conjuntos de variables medidas en algunos objetos.

La idea básica del PLS es la de reducción de la dimensión en regresión múltiple, con la garantía de que las primeras componentes ortogonales mejoran la predicción.

PLS no es un método, sino un conjunto de métodos con sus algoritmos asociados. Por mencionar algunos el algoritmo de modelado Path-PLS, el algoritmo PLS de Análisis de Componentes Principales (también conocido como NIPALS-PCA) o el algoritmo PLS para Análisis de correlación canónica (también conocido como NIPALS-CCA).

Si hay algo en común entre los métodos de mínimos cuadrados parciales es que todos ellos tienen un algoritmo asociado con un formato bastante uniforme. Lo métodos PLS proceden de una manera menos intuitiva en comparación con los procedimientos estadísticos clásicos, no se formulan términos para ser optimizados algebraicamente. Es decir, no se presenta ningún criterio de maximización o minimización. Por lo general expresamos un modelo de tal manera que se identifican componentes y las ecuaciones entre las componentes. En vez de obtener una solución analítica, desarrollamos hasta llegar a una solución a través de una serie de pasos secuenciales repetitivos para obtener una buena aproximación estable. En muchos casos los algoritmos PLS coinciden con las soluciones algebraicas.

El estudio de los métodos PLS tiene que ver con el hecho de que han evolucionado en la parte superior de las técnicas existentes. En general, primero se aprende acerca del problema y después en la Solución estándar, ya sea de regresión, la discriminación, los componentes principales, o correlaciones canónicas, solo por mencionar algunos de ellos. Es decir, hay que estudiar una técnica dada desde el ángulo PLS. En consecuencia, esto añade una capa adicional de conceptos, términos y jerga que hay que tratar bajo el enfoque estándar, y bajo el enfoque PLS.

Se ha demostrado que este método es una buena alternativa a los métodos más antiguos de regresión lineal porque es “robusto”, es decir, el modelo matemático no es muy alterado cuando se toman en cuenta nuevas muestras. Entre sus ventajas se encuentra resolver el problema de la multicolinealidad, esto es: alta correlaciones entre las variables predictoras. A este problema se ha unido en tiempos recientes el aumento extraordinario del número de variables con lo que es muy usual encontrarse con el requisito tradicional de que haya más individuos que variables no se cumple por lo tanto tampoco pueden emplearse los métodos tradicionales y para ellos nuevamente es una buena alternativa. Y por último tampoco funcionan bien los métodos tradicionales en el caso de existencia de datos ausentes y este algoritmo permite la realización de un análisis de datos ausentes sin necesidad de suprimirlos ni estimarlos.

Computacionalmente existe una gama amplia de programas de software que están disponibles comercialmente. El mayor grupo de herramientas es la docena de paquetes de R libremente disponibles en CRAN; también XLSTAT plug-ins (por Addinsoft) para MS Excel; SmartPLS (Ringle et al,2005), ADANCO, SIMCA (por Umetrics), los procedimientos de SAS y bibliotecas para MATLAB, y Python, entre otros.

CAAPITULO 3: ANALISIS DE COMPONENTES PRINCIPALES

Debido a que, las primeras publicaciones fueron relacionadas con el cálculo de Análisis de Componentes Principales y teniendo en cuenta que el estudio de los métodos PLS tiene que ver con las técnicas existentes de Análisis de Componentes Principales vamos hacer una breve mención a dicho modelo estadístico.

Esta técnica es debida a Hotelling (1933), aunque sus orígenes se encuentran en los ajustes ortogonales por mínimos cuadrados introducidos por Karl Pearson (1901).

El análisis de componentes principales tiene como objetivo describir con precisión los valores de p variables por un pequeño subconjunto m < p de ellas, se habrá reducido la dimensión del problema a costa de una pequeña pérdida de información. Es decir, dadas p variables correlacionadas (que miden información en común), se analiza si es posible representar adecuadamente esta información con un número menor de variables incorreladas entre sí (que no tenga redundancia en la información) y construidas como combinaciones lineales de las originales, denominando así a este subconjunto como componentes principales. Su utilidad es doble:

- Permite representar óptimamente en un espacio de dimensión pequeña observaciones de un espacio general p-dimensional. En este sentido, componentes principales es el primer paso para identificar las posibles variables latentes, o no observadas que genera los datos.
- Permite transformar las variables originales, en general, correladas, en nuevas variable incorreladas, facilitando la interpretación de los datos.

1.-DEFINICIÓN DE LAS COMPONENTES PRINCIPALES

Se considera una serie de variables $(x_1, x_2, \ldots, x_p)$ sobre un grupo de individuos y se trata de calcular un nuevo conjunto de variables $(y_1, y_2, \ldots, y_p)$ incorreladas entre sí, cuyas varianzas vayan decreciendo progresivamente.

Cada y_j (donde j=1,…..,p) es una combinación lineal de las $x_1, x_2, \ldots, x_p$ originales, es decir:

$$y_j = a_{j1}x_1 + a_{j2}\,x_2 + \ldots + a_{jp}\,x_p = a'_{jx}$$

Siendo a_j'= ($a_{1j}, a_{2j}, \ldots a_{pj}$) un vector de constantes, y x= $\begin{pmatrix} x1 \\ \ldots \\ x2 \end{pmatrix}$

Si lo que que se desea es maximizar la varianza, una forma simple podría ser aumentar los coeficientes a_{ij}. Por ello, para mantener la ortogonalidad de la transformación se impone que el módulo del vector a_j'= ($a_{1j}, a_{2j}, \ldots a_{pj}$) sea 1. Es decir,

$$a_j' a_j = \sum_{k=1}^{p} a_{kj}^2 = 1$$

El primer componente se calcula eligiendo a_1 de modo que y_1 tenga mayor varianza posible, sujeta a la restricción $a_1' a_1$ =1. El segundo componente se calcula obteniendo a_2 de modo que la variable obtenida, y_2 este incorrelada con y_1.

Del mismo modo se eligen $y_1, y_2, \ldots, y_p$, incorreladas entre sí, de manera que las variables aleatorias obtenidas vayan teniendo cada vez menor varianza.

2.-CÁLCULO DE LAS COMPONENTES

Se debe seleccionar a_1 que se maximice la varianza de y_1 sujeta a la condición de que $a_1' a_1$ =1.

$$\text{Var}\,(y_1) = \text{Var}\,(a_1' x) = a_1' \sum a_1$$

El método habitual para maximizar una función de varias variables sujeta a restricciones es el método de los *multiplicadores de Lagrange*.

El problema consiste en maximizar la función $a_1' \sum a_1$ sujeta a la restricción $a_1' a_1$ =1. Se puede observar que la incógnita es a_1 (el vector desconocido que nos da la combinación lineal óptima).

De esta forma se construye la función L:

$$L(a_1) = a_1' \sum a_1 - \lambda\,(a_1' a_1 - 1)$$

Y se obtiene el máximo, derivando e igualando a 0:

$$\frac{\partial L}{\partial a_1} = 2 \sum a_1 - 2\,\lambda I\, a_1 = 0 \rightarrow (\sum - \lambda I)\, a_1 = 0$$

Obteniendo en realidad un sistema lineal de ecuaciones que por el *teorema de Rochè-Frobenius* para que tenga una solución distinta de 0 la matriz $(\sum - \lambda I)$ tiene que ser singular, por lo que el determinante debe ser cero $|\sum - \lambda I)| = 0$; obteniendo de este modo

que λ es un autovalor de ∑. La matriz de covarianzas ∑ es de orden p y si además es definida positiva, tendrá p autovalores distintos, $\lambda_1, \lambda, \ldots, \lambda_p$ tales que $\lambda_1 > \lambda_2 > \ldots > \lambda_p$

Se tiene que, desarrollando la expresión anterior,

$$(\sum - \lambda I)\, a_1 = 0$$

$$\sum a_1 - \lambda\, I\, a_1 = 0$$

$$\sum a_1 = \lambda\, I\, a_1$$

Entonces,

$$\text{Var}\,(y_1) = \text{Var}\,(a_1' x) = a_1' \sum a_1 = a_1'\, \lambda\, I\, a_1 = \lambda\, a_1' a_1 = \lambda\, 1 = \lambda$$

Por lo tanto, para maximizar la varianza de y_1 se tiene que tomar el mayor autovalor, tomamos λ_1, y el correspondiente autovector a_1.

Realmente, a_1. es un vector que nos da la combinación de las variables originales que tiene mayor varianza, es decir, si $a_1' = (a_{11}, a_{12}, \ldots, a_{1p})$, entonces

$$y_1 = a_1' x = a_{11} x_1 + a_{12} x_2 + \ldots + a_{1p} x_p$$

El segundo componente principal, se tiene que $y_2 = a_2' x$, obteniéndose de forma análoga. Teniendo en cuenta además que de y_2 este incorrelada con de y_1, es decir,

$Cov(y_2, y_1) = 0$. Por lo tanto;

$$Cov(y_2, y_1) = Cov(a_2' x, a_1' x) = a_2'\, E[(x - \mu)(x - \mu)']\, a_1 = a_2' \sum a_1$$

Por lo tanto, se requiere que $a_2' \sum a_1 = 0$.

Anteriormente había obtenido que $\sum a_1 = \lambda\, a_1$, es equivalente a decir,

$$a_2' \sum a_1 = a_2'\, \lambda\, a_1 = \lambda\, a_2'\, a_1 = 0.$$

Esta expresión es equivalente a decir que $a_2'\, a_1 = 0$, que los vectores sean ortogonales.

De esta forma, hay que maximizar la varianza de y_2, es decir, $a_2 \sum a_2$, sujeta a las restricciones:

$$a_2'\, a_2 = 1$$

$$a_2'\, a_1 = 0$$

De la misma forma que antes, se toma la función:

$$L(a_2) = a_2' \sum a_2 - \lambda(a_2'\, a_2 - 1) - \delta\, a_2'\, a_1$$

Derivando con respecto a_2,

$$\frac{\partial L(a_2)}{\partial a_2} = 2\sum a_2 - 2\lambda\, a_2 - \delta a_1 = 0$$

Si se multiplica por a_1', se obtiene

$$2\, a_1' \sum a_2 - \delta = 0$$

Porque

$$a_1' a_2 = a_2'\, a_1 = 0$$

$$a_1' a_1 = 0$$

Por lo tanto,

$$\delta = 2\, a_1' \sum a_2 = 2\, a_2' \sum a_1 = 0$$

ya que la $Cov(y_2, y_1) = 0$.

De este modo,

$$\frac{\partial L(a_2)}{\partial a_2} = 2\sum a_2 - 2\lambda\, a_2 - \delta a_1 = 2\sum a_2 - 2\lambda\, a_2 = (\sum - \lambda I\ )a_2 = 0$$

Teniendo en cuenta los mismos razonamientos, se toma λ como el segundo mayor autovalor de la matriz ∑ con su autovector asociado a_2.

De igual forma, se puede extender estos razonamientos y de esta forma al j-ésimo componente de correspondería el j-ésimo autovalor.

Por lo que todos los componentes y (en total p) se expresan como el producto de una matriz formada por autovectores, multiplicada por el vector x que contiene las variables originales $x_1, x_2, \ldots, x_p$.

$$y = Ax$$

donde

$$y = \begin{pmatrix} y_1 \\ \cdots \\ y_p \end{pmatrix};\quad A = \begin{pmatrix} a_{11} & \cdots & a_{1p} \\ \cdot\cdot & \cdots & \cdots \\ a_{p1} & \cdots & a_{pp} \end{pmatrix};\quad x = \begin{pmatrix} x_1 \\ \cdots \\ x_p \end{pmatrix}$$

Como

$$Var(y_1) = \lambda_1$$

$$Var(y_2) = \lambda_2$$

….

$$Var(y_p) = \lambda_p$$

Y la matriz de covarianzas de **y** será

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix}$$

Como $(y_1, y_2, \dots, y_p)$ son variables incorreladas, se tiene que,

$$\Lambda = \mathrm{Var(Y)} = A' Var\,(X) A = A' \sum A$$

O bien,

$$\sum = A\,\Lambda\,\mathrm{A}'$$

Ya que A es una matriz ortogonal (porque $a_i' a_i = 1$ para todas sus columnas) por lo que $\mathrm{A}\,\mathrm{A}' = \mathrm{I}$

3.-PORCENTAJE DE VARIABILIDAD.

Se ha estudiado que cada autovalor correspondía a la varianza del componente y_i que se definía por medio del autovector a_i , esto es, $Var(\,y_i) = \lambda_i$

La varianza total de los componentes se obtiene sumando todos los autovalores ya que la matriz Λ es diagonal. Es decir:

$$\sum_{i=1}^{p} Var\,(\,y_i) = \sum_{i=1}^{p} \lambda_i \; = traza\,(\Lambda)$$

Por las propiedades del operador traza,

$$traza\,(\Lambda) = traza\,(\mathrm{A}'\sum\mathrm{A}) = traza\,(\sum\mathrm{A}'\mathrm{A}) = traza\,(\sum)$$

ya que $\mathrm{A}'\mathrm{A} = \mathrm{I}$ por ser A ortogonal, por lo tanto,

$$traza\,(\Lambda) = traza\,(\sum) = \sum_{i=1}^{p} Var\,(\,x_i)$$

Se puede concluir indicando que la suma de las varianzas de las variables originales y la suma de las varianzas de las componentes son iguales. Es decir, con ello podemos hablar del porcentaje de la varianza total que recoge un componente principal

$$\frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i} = \frac{\lambda_i}{\sum_{i=1}^{p} Var(\,x_i)}$$

Multiplicado por 100 obtendremos porcentajes.

De esta forma también se puede expresar el porcentaje de variabilidad recogido por los primeros m componentes donde m<p

$$\frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{p} Var(x_i)}$$

En la práctica, al tener p variables, nos quedaremos con un número mucho menos de componentes que recoja un porcentaje amplio de variabilidad total. En general, no se suele coger más de tres componentes principales, siempre que sea posible, para poder representarlos gráficamente.

4.-CALCULO DE LOS COMPONENTES PRINCIPALES A PARTIR DE LA MATRIZ DE CORRELACIONES.

Normalmente, se obtienen los componentes principales sobre variables originales estandarizadas, es decir, variables con media 0 y varianza 1. Esto equivales a tomar los componentes principales, no de la matriz de covarianzas sino de la matriz de correlaciones (en las variables estandarizadas coinciden las covarianzas y las correlaciones).

Los componentes son autovectores de la matriz de correlaciones y son distintos de los de la matriz de covarianzas. De esta forma, se da igual importancia a todas las variables originales.

En la matriz de correlaciones todos los elementos de la diagonal son iguales a 1. Si las variables originales están tipificadas, es decir que su matriz de covarianzas es igual a la de correlaciones, por lo que la variabilidad total (la traza) es igual al número total de variables que hay en la muestra. La suma total de todos los autovalores es p y la proporción de varianza recogida por el v autovector j-ésimo (componente) es

$$\frac{\lambda j}{p}$$

5.-CAMBIOS DE ESCALAS E IDENTIFICACIÓN DE COMPONENTES.

Si las variables originales $x_1, x_2, \ldots, x_p$ están incorreladas, entonces carece de sentido calcular unos componentes principales. Al calcularlos obtendríamos las mimas variables pero ordenadas de mayor a menor varianza. Para saber si $x_1, x_2, \ldots, x_p$ están correlacionadas, se puede calcular la matriz de correlaciones aplicándose posteriormente el *test de esfericidad de Barlett*.

El cálculo de los componentes principales de una serie de variables $x_1, x_2, \ldots, x_p$ depende normalmente de las unidades de medida empleadas. Si transformamos las unidades de medida, lo más probable es que cambien a su vez los componentes obtenidos.

Una solución frecuente es usar variables $x_1, x_2, \ldots, x_p$ tipificadas. Con ello, se eliminan las diferentes unidades de medida y se consideran todas las variables implícitamente equivalentes en cuanto a la información recogida.

Una de los objetivos del cálculo de componentes principales es la identificación de los mismos, es decir, averiguar qué información de la muestra resumen. Sin embargo, este es un problema difícil que a menudo resulta subjetivo. Habitualmente, se conservan sólo aquellos componentes que recogen la mayor parte de la variabilidad, hecho que permite representar los datos según dos o tres dimensiones si se conservan dos o tres ejes factoriales, pudiéndose identificar entonces grupos naturales entre las observaciones.

CAPITULO 4: ALGORITMO DE REGRESION PLS1

La idea general del PLS es intentar extraer estos factores latentes, recogiendo la mayor parte de la variación de los factores reales de forma que además sirvan para modelar las variables respuesta de la mejor manera posible.

Se denomina regresión PLS1 cuando se estudia una sola variable a explicar con p variables explicativas y regresión PLS2 cuando existen varias variables a explicar (q>1) y p variables explicativas. A continuación, se desarrolla el algoritmo.

1.-Normalización de los datos.

Existen dos tipos de normalización de datos que se encuentran con frecuencias en los distintos textos. El primero consiste en restar para cada una de las variables su media y dividir por la raíz cuadrada de la suma de cuadrados de las desviaciones de su media.

$$y_i = \frac{y_i - \bar{y}}{\sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \quad (i=1,2\ldots n)$$

$$x_{ij} = \frac{x_{ij} - \bar{x_j}}{\sqrt{\sum_{i=1}^{n} (x_{ij} - \bar{x_j})^2}} \quad (i=1,2\ldots n;\ j=1,2\ldots p)$$

El segundo, consiste en restar para cada una de las variables su media y dividir por la raíz cuadrada de la suma de cuadrados de las desviaciones a su media dividido por (n-1), es decir dividir por la raíz cuadrada de la cuasivarianza muestral.

$$y_i = \frac{y_i - \bar{y}}{\sqrt{\frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{n-1}}} \quad (i=1,2\ldots n)$$

$$x_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\frac{\sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2}{n-1}}} \quad (i=1,2\ldots n;\ j=1,2\ldots p)$$

Se puede demostrar que el tipo de normalización no influye en la ecuación de predicción lineal, es decir, los coeficientes de regresión que se obtienen son iguales para los dos métodos. En este caso vamos a usar el segundo método de normalización de datos por ser el más frecuente entre los distintos autores.

El primer paso para el algoritmo sería la normalización de los datos que como se ha especificado se realizaría con el segundo método.

2.-Construcción de la primera componente t_1.

La primera componente t_1 se define como con la siguiente fórmula, teniendo en cuenta los datos normalizados

$$y_1 = w_{11}x_1 + w_{12}\,x_2 + \ldots + w_{1p}\,x_p$$

es decir: $\sum_{j=1}^{p} w_{1j}x_j$ donde:

$$w_{1j} = \frac{Cov(x_j, y)}{\sqrt{\sum_{j=1}^{p} Cov^2(x_j, y)}} = \frac{\langle x_j, y \rangle}{\sqrt{\sum_{j=1}^{p} \langle x_j, y \rangle^2}} \quad (j = 1,2,\ldots,p)$$

denotando por $\langle x_j, y \rangle = Cov(x_j, y)$

2.1.-Detección de individuos atípicos.

Para mejorar la calidad se puede aplicar un procedimiento de detección de individuos atípicos. La regla general de decisión para dicha detección sobre un conjunto de A componentes está basada en la variable aleatoria:

$$t_i^A = \frac{n(n-A)}{A(n^2-1)} T_i^2$$

sigue una ley de Fisher-Snedecor con A grados de libertad para el numerador y n-A grados de libertad para el denominador, donde T_i^2 es la T^2 de Hotelling de la observación i, calculada utilizando A componentes siendo igual a:

$$T_i^2 = \frac{n}{n-1}\sum_{h=1}^{A}\cdot\frac{t_{i,h}^2}{Sh^2} \quad (i = 1, 2, \dots, n)$$

donde n es el número total de individuos, $\|t_h\|^2$ es la norma euclídea al cuadrado de la componente h S_h^2: es la cuasivarianza ($S_h^2 = \frac{\|t_h\|2}{n-1}$) de la componente h y $t_{i,h}$ es el valor de la componente h para la observación i.

En este caso pasamos a la detección de individuos atípicos para el modelo generado por la primera componente. La regla general de decisión para una sola componente, baja nuestra nomenclatura, tiene la siguiente forma (se sustituye en la fórmula A=1)

$$t_{i,1}^A = \frac{n(n-1)}{1(n^2-1)}\left(\frac{n\, t_{i,1}^2}{n-1\, \|t_1\|^2}\right) = \frac{n^2\, t_{i,1}^2}{n+1\, \|t_1\|^2}$$

Si $t_{i,1}^A \geq F_{F_{n-1}^1}^{-1}(1-\alpha)$ se acepta la hipótesis que el individuo i es atípico.

Si $t_{i,1}^A < F_{F_{n-1}^1}^{-1}(1-\alpha)$ se rechaza la hipótesis que el individuo i es atípico.

Esta regla de decisión equivale a:

Si $t_{i,1}^A \geq \left(F_{T_{n-1}}^{-1}(1-\frac{\alpha}{2})\right)^2$ se acepta la hipótesis que el individuo i es atípico.

Si $t_{i,1}^A < \left(F_{T_{n-1}}^{-1}(1-\frac{\alpha}{2})\right)^2$ se rechaza la hipótesis que el individuo i es atípico.

Siendo $F_{F_{n-1}^1}^{-1}(1-\alpha)$ la función inversa de la función de distribución (función cuantil) de la variable aleatoria de Fisher-Snedecor con 1 grado de libertad para el numerados y n-1 grado de libertad para el denominador para un área de 1-α y $F_{T_{n-1}}^{-1}(1-\frac{\alpha}{2})$ es la función inversa de la función de distribución de la variable aleatoria T de Student-Fisher con n-1 grado de libertad para un área de $(1-\frac{\alpha}{2})$.

$\|t_1\|^2$ es la norma euclídea al cuadrado de la componente t_1.

Si como resultado obtenemos una muestra homogénea continuamos con el algoritmo, si por el contrario resulta valor atípico, eliminaremos el individuo o individuos atípico y comenzamos de nuevo.

2.2.-Cálculo de la regresión lineal simple de y sobre t_1. y el test de significación de la regresión.

Primero se busca la ecuación lineal de predicción estimada de y, para posteriormente comprobar si dicha regresión lineal es significativa.

- La ecuación Lineal de Predicción Estimada se desarrolla de la siguiente forma:

$$y^* = \widehat{\beta}_1^* t_1$$

donde, la estimación del coeficiente de regresión ha sido calculada a partir de:

$$\widehat{\beta}_1 = \frac{< y,t_1 >}{||t_1||^2} = \frac{\sqrt{n-1}}{||t_1||} r_{y,t_1}$$

denotando por $< y, t_1 >= Cov\,(y, t_1)$

De esta fórmula se deduce que, el coeficiente de regresión es igual al coeficiente de correlación simple cuando se cumple $||t_1|| = \sqrt{n-1}$. Esta situación se verifica cuando las variables originales están normalizadas por el método de la cuasivarianza muestral. Ahora se pueden calcular los residuos asociado a la recta de regresión mediante una simple resta:

$$e_1 = y - y^*$$

- Test de Significación Global de la Regresión Lineal.

Usando el test de Fisher que permite determinar si la regresión lineal simple es significativa. Teniendo en cuenta la regla general de decisión del test de Fisher, para una componente explicativa, bajo nuestra nomenclatura se tiene la siguiente forma:

$$t_{i,1}^A \geq F_{F_{n-1}^1}^{-1}(1-\alpha)$$

Si $F_{n-2}^{1*} \geq F_{F_{n-2}^1}^{-1}(1-\alpha)$ la componente explicativa es significativa.

Si $F_{n-2}^{1*} < F_{F_{n-2}^1}^{-1}(1-\alpha)$ la componente explicativa no es significativa.

Donde, $F_{n-2}^1 = (n-2)\frac{r_{y,t_1}^2}{1-r_{y,t_1}^2} = (n-2)\frac{[< y,t_1 >]^2}{(n-1)||t_1||^2 - ||t_1||^2}$

Y $F_{F_{n-1}^1}^{-1}(1-\alpha)$ es la función cuantil de la función de distribución de la variable aleatoria F de Fisher-Snedecor con 1 grado de libertar para el numerador y n-2 grados de libertad para el denominar para un área de 1-α.

$< y, t_1 >$: es el producto escalar clásico de y con t_1.

$||t_1||^2$: es la norma al cuadrado de la componente t_1.

Suponiendo que la componente sea significativa procedemos a deshacer los cambios efectuados.

- Deshacer el cambio de t_1 a $x_1, x_2, \ldots, x_p$ siendo:

$$y = \hat{\beta}_1 t_1 = \hat{\beta}_1 (w_{11} x_1 + w_{12} x_2 + \ldots + w_{1p} x_p)$$

De esta forma se llega a la ecuación de predicción estimada en función de las variables explicadas originales:

$$y^* = \sum_{j=1}^{p} \hat{\beta}_1^* w_{1j} x_j$$

Los coeficientes de regresión de esta ecuación de predicción estimada, son más fáciles para su interpretación.

- Por último, deshacer el cambio de la normalización de las variables:

$$y = \hat{\beta}_1 (w_{11} (\frac{x_1 - \bar{x}_1}{s_{x_1}^2}) + \cdots + w_{1p} (\frac{x_p - \bar{x}_p}{s_{x_p}^2}))$$

3.-Construcción de la segunda componente t_2.

Si el resultado obtenido en el test de significación global de la regresión lineal realizado anteriormente es demasiado débil, se busca construir una segunda componente t_2., combinación lineal de las x_j, además no correlacionada con t_1.y explicando bien el residuo. Esta componente $t_2$es combinación lineal de los residuos e_{1j} de las regresiones de las variables $x_j$sobre la componente t_1.. Teniendo en cuenta todo ello se obtiene t_2 con la fórmula:

$$t_2 = w_{21} e_{11} + w_{22} e_{12} + \ldots + w_{2p} e_{1p}$$

donde:

$$w_{2j} = \frac{Cov\,(e_{1j}, e_1)}{\sqrt{\sum_{j=1}^{p} Cov^2\,(e_{1j}, e_1)}} = \frac{<e_1, e_{1j}>}{\sqrt{\sum_{j=1}^{p} <(e_{1j}, e_1)>^2}};\ (\mathrm{j} = 1, 2, \dots, \mathrm{p})$$

Para el cálculo de los residuos: $\mathrm{e}_{ij}\ (\mathrm{j} = 1, 2, \dots, \mathrm{p})$ efectuamos las regresiones simples de $\mathrm{x_j}$ sobre $\mathrm{t}_1\ (\mathrm{j} = 1, 2, \dots, \mathrm{p})$ y obtenemos las rectas de predicción estimadas:

$$x_j^* = \hat{\alpha}_j^* \mathrm{t}_1 \quad (\mathrm{j} = 1, 2, \dots, \mathrm{p})$$

donde las estimaciones de los coeficientes de regresión han sido calculadas de la siguiente forma:

$$\hat{\alpha}_j = \frac{<x_j, t_1>}{||t_1||^2} = \frac{\sqrt{n-1}}{||t_1||}\, \mathrm{r}_{\mathrm{x}_j, \mathrm{t}_1} \quad (\mathrm{j} = 1, 2, \dots, \mathrm{p})$$

Ahora se puede calcular los residuos asociados a las rectas de regresión:

$$\mathrm{e}_{ij} = \mathrm{x}_j - x_j^* \quad (\mathrm{j} = 1, 2, \dots, \mathrm{p})$$

Como ya se conoce e_1 y $\mathrm{e}_{ij}\ (\mathrm{j} = 1, 2, \dots, \mathrm{p})$ sólo se tiene que calcular los productos escalares clásicos $<\mathrm{e}_1, \mathrm{e}_{1j}>\ (\mathrm{j} = 1, 2, \dots, \mathrm{p})$ para calcular la componente t_2.

3.1.-Detección de individuos atípicos para la segunda componente.

El procedimiento es similar que en el caso de la primera componente y de igual forma si el resultado es una muestra con valores atípico se eliminan y recomenzar y si por el contrario se obtiene una muestra homogénea se continua con el algoritmo.

3.2.-Cálculo de la regresión lineal simple de y_1 sobre t_2 y el test de significación de la regresión.

Análogamente a lo realizado, buscar la ecuación lineal de predicción estimada y posteriormente comprobar si la regresión lineal simple es globalmente significativa.

- La ecuación Lineal de Predicción Estimada se desarrolla de la siguiente forma:

$$y^* = \widehat{\beta}_2^* \mathrm{t}_2$$

donde, la estimación del coeficiente de regresión ha sido calculada a partir de:

$$\widehat{\beta}_2 = \frac{< y, t_2 >}{||t_2||^2} = \frac{\sqrt{n-1}}{||t_2||} r_{y,t_2}$$

De esta fórmula se deduce que, el coeficiente de regresión es igual al coeficiente de correlación simple cuando se cumple $||t_2|| = \sqrt{n-1}$. Esta situación se verifica cuando las variables originales están normalizadas por el método de la cuasivarianza muestral. Ahora se puede calcular el residuo asociado a la recta de regresión mediante una simple resta:

$$e_2 = y - y^*$$

- Test de Significación Global de la Regresión Lineal.

En este caso el test se realiza de la misma forma que para el caso de la primera componente.

4.-Detección de individuos atípicos en el plano (t_1-t_2).

La regla general de decisión para la detección de individuos atípicos cuando se trata de dos componentes, adopta la siguiente forma:

Si $t^A_{i(1-2)} \geq 1$ se acepta la hipótesis que el individuo i es atípico.

Si $t^A_{i(1-2)} < 1$ se rechaza la hipótesis que el individuo i es atípico.

Siendo $t^A_{i(1-2)} = \frac{t^2_{i,1}}{\frac{2(n^2-1)}{n^2(n-2)}||t_1||^2 F^{-1}_{F^2_{n-2}}(1-\alpha)} + \frac{t^2_{i,2}}{\frac{2(n^2-1)}{n^2(n-2)}||t_2||^2 F^{-1}_{F^2_{n-2}}(1-\alpha)}$ y $F^{-1}_{F^2_{n-2}}(1-\alpha)$ es la función cuantil de la función de distribución de la variable aleatoria F de Fisher-Snedecor con dos grados de libertad para el numerador y n-2 grados de libertad para el denominador para un área de $(1-\alpha)$.

$||t_1||^2$ es la norma al cuadrado de la componente t_1.

$||t_2||^2$ es la norma al cuadrado de la componente t_2.

En el caso de que la muestra haya sido homogénea, se procede a realizar el siguiente apartado. En caso contrario, se elimina el individuo o individuos y comenzar de nuevo

5.- Regresión lineal múltiple sobre las dos primeras componentes y test de significatividad global de la regresión.

Primero buscar la ecuación lineal de predicción estimada y-- a continuación, comprobar la significatividad global de la regresión lineal múltiple.

5.1.-Ecuación lineal de predicción estimada.

La ecuación lineal de predicción estimada toma la siguiente forma:

$$y^* = \widehat{\beta}^*_1 t_1 + \widehat{\beta}^*_2 t_2$$

donde las estimaciones de los coeficientes de regresión han sido calculadas a partir de las siguientes fórmulas:

$$\widehat{\beta}_1 = \frac{\sqrt{n-1}}{||t_1||}\left(\frac{r_{y,t_1} - r_{y,t_2}\, r_{t_1,t_2}}{1-r^2_{t_1,t_2}}\right)$$

$$\widehat{\beta}_2 = \frac{\sqrt{n-1}}{||t_2||}\left(\frac{r_{y,t_2} - r_{y,t_1}\, r_{t_1,t_2}}{1-r^2_{t_1,t_2}}\right)$$

y dado que las componentes t_1 y t_2 son ortogonales: $r_{t_1,t_2} = 0$ y por tanto, los dos estimadores se reducen a:

$$\widehat{\beta}_1 = \frac{\sqrt{n-1}}{||t_1||} r_{y,t_1} \qquad \widehat{\beta}_2 = \frac{\sqrt{n-1}}{||t_2||} r_{y,t_2}$$

En estos momentos estamos en condiciones de calcular el residuo asociado a la línea de regresión mediante una sustracción:

$$e_2 = y - y^* = e_1 - y^* \qquad \text{recordando que } (e_1 = y)$$

5.2.-Test de significación global de la regresión.

El test de Fisher permite determinar si, la regresión lineal múltiple es globalmente significativa.

La regla general de decisión del test de Fisher para dos componentes explicativas ortogonales, bajo nuestra nomenclatura, adopta la siguiente forma:

Si $F_{n-3}^{2*} \geq F_{F_{n-3}^{2}}^{-1}(1-\alpha)$, las componentes explicativas t_1 y t_2 son significativas.

Si $F_{n-3}^{2*} < F_{F_{n-3}^{2}}^{-1}(1-\alpha)$, las componentes explicativas t_1 y t_2 son significativas.

donde, $F_{n-3}^{2*} = \frac{n-3}{2}\left(\frac{\sum_{j=1}^{2} r_{y,t_j}^2}{1-\sum_{j=1}^{2} r_{y,t_j}^2}\right)$ o bien:

$$F_{n-3}^{2*} = \frac{n-3}{2}\left(\frac{||t_2||^2[<y,t_1>]^2+||t_1||^2[<y,t_2>]^2}{(n-1)||t_1||^2||t_2||^2-(||t_2||^2[<y,t_1>]^2+||t_1||^2[<y,t_2>]^2)}\right)$$

- $F_{F_{n-3}^{2}}^{-1}(1-\alpha)$: es la función cuantil de la función de distribución de la variable aleatoria F de Fisher-Snedecor con 2 grados de libertad para el numerador y n-3 grados de libertad para el denominador para un área $1-\alpha$.
- r_{y,t_j}^2: coeficientes de correlación lineal de Bravais-Pearson al cuadrado entre y y t_j $j=(1,2,\dots,p)$.
- $< y, t_j >$: es el producto escalar clásico de y con t_j $j=(1,2,\dots,p)$.
- $||t_j||^2$ es la norma al cuadrado de la componente t_j $j=(1,2,\dots,p)$.

En el caso hipotético de que las componentes t_1 y t_2 sean significativas proceder a deshacer los siguientes cambios.

En primer lugar, de t_1 y t_2 a $x_1, x_2, \dots, x_p$ y por último deshacer el cambio de la normalización de las variables.

Con ello se llega a la ecuación de predicción estimada en función de las variables explicativas originales.

6.- Construcción de las sucesivas componentes.

Si el poder explicativo de esta regresión es todavía débil, se busca construir una tercera componente. Esta tercera componente, es combinación lineal de los residuos e_{2j} obtenidos como consecuencia de la regresión de los residuos e_{1j} sobre t_2 . De esta forma obtener t_3 con la siguiente fórmula:

$$t_3 = w_{31}e_{21} + w_{32}\,e_{22} + \dots + w_{3p}\,e_{2p}$$

donde:

$$w_{3j} = \frac{Cov\,(e_{2j}, e_2)}{\sqrt{\sum_{j=1}^{p} Cov^2\,(e_{2j}, e_2)}} = \frac{<e_2, e_{2j}>}{\sqrt{\sum_{j=1}^{p} <(e_{2j}, e_2)>^2}};\ (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$$

Para el cálculo de los residuos: $\mathrm{e}_{2j}\ (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$ efectuar las regresiones simples de e_{1j} sobre $\mathrm{t}_2\ (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$ y obtener las rectas de predicción estimadas:

$$e_{1j}^{*} = \hat{\alpha}_{1j}^{*}\,\mathrm{t}_2 \quad (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$$

donde las estimaciones de los coeficientes de regresión han sido calculadas de la siguiente forma:

$$\hat{\alpha}_{1j} = \frac{<e_{1j}, t_2>}{||t_2||^2} \quad (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$$

Ahora se podría calcular los residuos asociados a las rectas de regresión:

$$\mathrm{e}_{2j} = \mathrm{e}_{1j} - e_{1j}^{*} \quad (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$$

Como ya se conoce e_2 y $\mathrm{e}_{2j}\ (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$ sólo se tiene que calcular los productos escalares clásicos $<\mathrm{e}_2, \mathrm{e}_{2j}>\ (\mathrm{j} = 1, 2, \ldots, \mathrm{p})$ para calcular la componente t_3.

A partir de aquí, seguir los mismos pasos que los realizados anteriormente para las dos componentes anteriores.

Este procedimiento iterativo continúa hasta que el número de componentes a retener sea significativo.

Usar el método de validación cruzada para obtener de forma más precisa de la que se ha expuesto con anterioridad el número de componentes $t_1, t_2, \ldots, t_H$ a retener. En el tema en el que se desarrolla la regresión PLS2 se hará una mención más detallada del método de validación cruzada, todo lo que allí se mencione puede ser extendido a la regresión PLS1 como caso particular en el que el número de variables a explicar es sólo una.

CAPITULO 5: ALGORITMO DE REGRESION PLS2.

Teniendo un conjunto de variables a explicar $Y = (y_1, y, \ldots, y_q)$ que tratamos de relacionar con otro conjunto de variables explicativas o predictoras $X = (x_1, x, \ldots, x_p)$ el algoritmo de regresión PLS2 efectúa una reducción de la dimensionalidad de un conjunto de variables X, bajo la condición de que estas componentes principales sean también los más explicativas posibles respecto del conjunto de variables Y. En este caso es posible predecir las variables y_k a partir de las x_j separando mejor lo que es común a los datos de aquello que es más específico.

La notación que se va a utilizar será:

n= números de individuos.

p= número de variables explicativas.

q= número de variables objetivo.

A= número de componentes a retenidas.

X = matriz de datos para las variables explicativas Nxp.

Y = matriz de datos Nxq para las variables a explicar.

E_0 = matriz de las variables x_j explicativas centradas y reducidas (tipificadas).

F_0 = matriz de las variables a explicar centradas y reducidas (tipificadas).

E_h = matriz de residuos de la descomposición de E_0 utilizando h componentes.

E_{hj} = j-ésima columna de E_h.

F_h = matriz de residuos de la descomposición de F_0 utilizando h componentes.

F_{hk} = k-ésima columna de F_h.

1.-Visión general del algoritmo de la regresión PLS2.

Etapas y fundamentos más relevantes:

Etapa 0: se comienza con la construcción de las matrices E_0 y F_0 (matrices formadas por

las variables centradas y tipificadas de las variables predictoras y respuesta respectivamente).

Etapa 1: se construye una combinación lineal u_1 de las columnas de F_0 y una

combinación lineal t_1 de las columnas de E_0 que maximice:

$$cov(u_1, t_1) = \sqrt{var(t_1) * var(u_1)} \; corr(u_1, t_1)$$

Se obtienen dos nuevas variables u_1 y t_1 lo más correlacionadas posible y que resumen lo mejor que se pueda la información contenida en las matrices E_0 y F_0.

Posteriormente se construye la regresión lineal simple tanto del conjunto de variables explicativas como del conjunto de variables a explicar sobre la componente t_1 .

$$E_0 = t_1 \; p_1^t + E_1$$

$$F_0 = t_1 \; r_1^t + F_1$$

donde p_1 y r_1 son los vectores de los coeficientes de regresión.

Etapa 2: se repite la etapa 1, reemplazando E_0 y F_0 por las nuevas matrices (residuales)

E_1 y F_1. De esta forma se obtiene dos nuevas componentes; t_2(combinación lineal de las columnas de E_1) y u_2 (combinación lineal de las columnas de F_1) que maximicen la covarianza de (u_2, t_2). A partir de estas componentes se obtiene por regresión lineal simple:

$$E_1 = t_2 \; p_2^t + E_2$$

$$F_1 = t_2 \; r_2^t + F_2$$

Por lo que se deduce:

$$E_0 = t_1 \; p_1^t + t_2 \; p_2^t + E_2$$

$$F_0 = t_1 \; r_1^t + t_2 \; r_2^t + F_2$$

Las etapas se repiten hasta que las componentes $t_1, t_2, \ldots, t_A$ expliquen suficientemente F_0. En el siguiente apartado se desmuestra que las componentes t_h son combinaciones lineales de las columnas de E_0 y no se encuentran correlacionadas entre ellas.

De la descomposición:

$$F_0 = t_1\ r_1^t\ + t_1\ \ r_2^t + \cdots + t_h\ r_h^t + F_h$$

Se deducen las ecuaciones de regresión PLS:

$$y_k^* = \widehat{\beta}_{k,0}^*\ + \widehat{\beta}_{k,1}^*\, x_1 + \widehat{\beta}_{k,2}^*\, x_2 + \cdots + \widehat{\beta}_{k,p}^*\, x_p \qquad k = 1,2,\ldots,q$$

2.-Determinación y propiedades de las primeras componentes.

2.1.-Determinación de las primeras componentes.

Buscar una componente combinación lineal de E_0 denominada t_1 y otra componente u_1 combinación lineal de F_0.

$$t_1\ = E_0 w_1 \qquad \text{y} \qquad u_1\ = F_0 c_1$$

de tal forma que w_1 y c_1 tengan norma 1. Es decir, $\left|\left|w_1\right|\right| = \left|\left|c_1\right|\right| = 1$

Estas combinaciones lineales deben obtenerse teniendo en cuenta que se maximice la covarianza entre ellas,

$$\text{cov}(u_1, t_1) = \sqrt{\text{var}(t_1) * \text{var}(u_1)}\ \ \text{corr}(u_1, t_1)$$

por lo que se maximiza simultáneamente la varianza explicada por t_1, la varianza explicada por u_1, y la correlación entre estas dos componentes.

Se buscan por lo tanto los vectores de norma 1, w_1 y c_1 que maximicen:

$$< t_1, u_1\ >= ||t_1||||u_1||corr(t_1, u_1)$$

donde $< t_1, u_1\ >$, representa el producto escalar clásico.

Utilizando el método de los multiplicadores de Langrange, se trata de maximizar la función:

$$s = \ w_1^t\, E_0^t F_0 c_1\ - \lambda_1\left(\, w_1^t w_1\ - 1\right) - \lambda_2\left(\, c_1^t c_1\ - 1\right)$$

Igualando a 0 las derivadas parciales:

$$\frac{\partial s}{\partial \lambda_1} = \ -\left(\, w_1^t w_1\ - 1\right) = 0$$

$$\frac{\partial s}{\partial \lambda_2} = -\left(c_1^t c_1 - 1 \right) = 0$$

$$\frac{\partial s}{\partial w_1} = E_0^t F_0 c_1 - 2\lambda_1 w_1 = 0 \Rightarrow w_1^t E_0^t F_0 c_1 = 2\lambda_1 \Rightarrow < t_1, u_1 >= 2\lambda_1 = \theta_1$$

$$\frac{\partial s}{\partial c_1} = F_0^t E_0 w_1 - 2\lambda_2 c_1 = 0 \Rightarrow c_1^t F_0^t E_0 w_1 = 2\lambda_2 \Rightarrow < t_1, u_1 >= 2\lambda_1 = 2\lambda_2 = \theta_1$$

De donde se obtienen las siguientes relaciones:

$$E_0^t F_0 c_1 = 2\lambda_1 w_1 \Rightarrow E_0^t F_0 = \theta_1 w_1 c_1^t$$

$$F_0^t E_0 w_1 = 2\lambda_2 c_1 \Rightarrow F_0^t E_0 w_1 = \theta_1 c_1$$

$$\Rightarrow E_0^t F_0 F_0^t E_0 w_1 = \theta_1^2 w_1$$

por lo que w_1 es el autovector de la matriz $E_0^t F_0 F_0^t E_0$ correspondiente al mayor autovalor θ_1^2 de la matriz citada (recordar que el producto de una matriz cualquiera de números reales por su traspuesta y el producto de su traspuesta por ella misma tienen los mismos autovalores no nulos).

Posteriormente, se llevan a cabo las dos regresiones siguientes: E_0 sobre t_1 y F_0 sobre t_1:

$$E_0 = t_1 p_1^t + E_1$$

$$F_0 = t_1 r_1^t + F_1$$

donde $p_1 = \frac{E_0^t t_1}{t_1^t t_1}$ es el vector de los coeficientes de regresión sobre t_1 para cada variable original independiente x_j (en la notación general E_{0j}) y $r_1 = \frac{F_0^t t_1}{t_1^t t_1}$ es el vector de los coeficientes de regresión de t_1 para cada variable original dependiente y_k (en la notación general F_{0k}). Se considera que la norma euclídea al cuadrado de un vector cualquiera (y las componentes lo son) es la suma de cuadrados de sus elementos, es decir:

$$t_1^t t_1 = ||t_1||^2$$

La representación en el plano (t_1, u_1) permite visualizar la relación entre las x_j y las y_k detectada por estas primeras componentes PLS. El grado de ajuste si los datos se encuentran muy próximos a una recta), la existencia de curvatura e incluso puntos atípicos, se pueden detectar a través de los patrones que se vislumbren en estas gráficas.

2.2.-Propiedades de las primeras componentes.

1) $p_1^t w_1 = 1$,

ya que $p_1^t w_1 = \frac{t_1^t E_0}{t_1^t t_1} w_1 = \frac{t_1^t}{t_1^t t_1} E_0 w_1 = \frac{t_1^t}{t_1^t t_1} t_1 = 1$

2) $r_1 = b_1 c_1$, es decir, r_1 y c_1 son colineales (donde b_1 es el coeficiente de regresión de u_1 sobre t_1.)

3) $t_1^t E_1 = 0$, las componentes y los residuos de la regresión para cada variable X son ortogonales.

$$t_1^t E_1 = t_1^t (E_0 - t_1 p_1^t) = t_1^t E_0 - t_1^t t_1 \left(\frac{t_1^t E_0}{t_1^t t_1} \right) = 0$$

De la misma forma $t_1^t F_1 = 0$

3.-Determinación y propiedades de las segundas componentes.

3.1.-Determinación de las primeras componentes.

De forma similar al del paso anterior, con la salvedad de que ahora se reemplazan los conjuntos de datos iniciales normalizados $E_0(X)$ y $F_0(X)$ por las matrices de los residuales E_1 y F_1

Así se determina:

$t_2 = E_1 w_2$ con w_2 vector propio de $E_1^t F_1 F_1^t E_1$

$u_2 = F_1 c_2$ con c_2 vector propio de $F_1^t E_1 E_1^t F_1$

$$\square_1 = u_2^t t_2$$

$$p_2 = \frac{E_1^t t_2}{t_2^t t_2}$$

$$r_2 = \frac{F_1^t t_2}{t_2^t t_2} = \frac{F_1^t E_1 w_2}{t_2^t t_2} = \frac{\square_2 c_2}{t_2^t t_2} = b_2 c_2$$

donde nuevamente b_2 es el coeficiente de regresión de u_2 sobre t_2 $b_2 = \frac{u_1^t t_2}{t_2^t t_2}$

$$E_1 = t_2\ p_2^t + E_2 => E_0 = t_1\ p_1^t + t_2\ p_2^t + E_2$$

$$F_1 = t_2\ r_2^t + F_2 => F_0 = t_1\ r_1^t + t_2\ r_2^t + F_2$$

En las siguientes etapas se desarrolla de forma similar.

3.2.-Gráficas de interpretación.

Al representar (t_1, t_2) se permite visualizar a los individuos en un plano que resume a las variables x_j $(j = 1,2, \dots, p)$, orientándolas hacia la mejor explicación posible de las variables y_k $(k = 1,2, \dots, q)$. También se pueden detectar irregularidades en los datos, patrones no aleatorios, agrupación de observaciones a partir de las cuales puedan establecerse conglomerados; si se observan alguna curvatura podría ser un indicio para añadir algún término cuadrático, dos o más agrupaciones de observaciones indican que deben analizarse los grupos de forma separada, etc.

4.-Resultados generales para cualquier etapa.

Se desarrollan diversos resultados generales para la compresión del método.

En una etapa cualquiera h, se obtienen:

$$E_{h-1}^t F_{h-1} c_h = \theta_h w_h$$

$$F_{h-1}^t E_{h-1} w_h = \theta_h c_h$$

$$E_{h-1}^t F_h F_{h-1}^t E_{h-1} w_h = \theta_h^2 w_h$$

$$F_{h-1}^t E_h E_{h-1}^t F_{h-1} c_h = \theta_h^2 c_h$$

Estas ecuaciones permiten calcular w_h y c_h como autovectores correspondientes a los autovalores mayores de las matrices $E_{h-1}^t F_h F_{h-1}^t E_{h-1}$ y $F_{h-1}^t E_h E_{h-1}^t F_{h-1}$ respectivamente, y las componentes PLS u_h y t_h como,

$$u_h = F_{h-1} c_h$$

$$t_h = E_{h-1} w_h$$

A continuación, se llevan a cabo las regresiones de E_{h-1} sobre t_h y de F_{h-1} sobre t_h determinándose las siguientes ecuaciones:

$E_{h-1} = t_h\ p_h^t + E_h$ donde $p_h = \frac{E_{h-1}^t t_h}{t_h^t t_h}$

$F_{h-1} = t_h\ r^t_h + F_h$ donde $r_h = \frac{F^t_{h-1} t_h}{t^t_h t_h} = b_h c_h$

siendo b_h el coeficiente de regresión de u_h sobre t_h, ya que:

$$r_h = \frac{F^t_{h-1} t_h}{t^t_h t_h} = \frac{F^t_{h-1} E_{h-1}\ w_h}{t^t_h t_h} = \frac{\theta_h}{t^t_h t_h} c_h = \frac{w^t_h\ E^t_{h-1} F_{h-1}\ c_h}{t^t_h t_h} c_h = \frac{t^t_h u_h}{t^t_h t_h} c_h$$

Además, siempre se verifican las igualdades:

$$w^t_h p_h = 1$$

$$t^t_h E_h = 0$$

La primera igualdad se desarrolla porque:

$$w^t_h p_h = \frac{t^t_h E_{h-1}}{t^t_h t_h} w_h = \frac{t^t_h}{t^t_h t_h} E_{h-1} w_h = \frac{t^t_h}{t^t_h t_h} t_h = 1$$

La segunda también se desarrolla:

$$t^t_h E_h = t^t_h (E_{h-1} - t_h\ p^t_h) = t^t_h E_{h-1} - t^t_h t_h \left(\frac{t^t_h E_{h-1}}{t^t_h t_h}\right) = 0$$

Partiendo del conocimiento de w_h, que no hay que olvidar que se determina como el autovector correspondiente al autovalor propio de θ^2_h de la matriz $(E^t_{h-1} F_h F^t_{h-1} E_{h-1})$, se llega a obtener las relaciones cíclicas siguientes:

$$t_h = E_{h-1} w_h$$

$$c_h = \frac{1}{\theta_h} F^t_{h-1} E_{h-1} w_h = \frac{1}{\theta_h} F^t_{h-1} t_h$$

$$u_h = F_{h-1} c_h$$

$$w_h = \frac{1}{\theta_h} E^t_{h-1} F_{h-1} c_h = \frac{1}{\theta_h} E^t_{h-1} u_h$$

Usualmente la correlación entre las componentes t_h y u_h van decreciendo a medida que extraemos nuevas componentes. Las representaciones entre las componentes de las variables predictoras y las componentes de las variables respuesta pueden confirmar esta situación generalizada.

5.-Relaciones de ortogonalidad.

El cumplimiento de las siguientes reglas de ortogonalidad es lo que realmente permite aplicar con rigor el procedimiento PLS, puesto que cuando se añade una nueva componente a las h obtenidas con anterioridad se sabe que se está optimizando los objetivos sobre h+1 dimensiones.

Se detallan las reglas de ortogonalidad:

1) Las componentes PLS, obtenidas a partir de las variables explicativas y sus valores residuales, son ortogonales entre sí, es decir:

$$t_h^t t_l = 0 \qquad \forall l > h$$

Se demuestra por recurrencia:

- $t_1^t t_2 = t_1^t E_1 w_2 = 0$ ya que se demostró antes que $t_h^t E_h = 0 \qquad \forall h$
- Suponer que $(t_1, t_2, \dots, t_h)$ son ortogonales; ver que $(t_1, t_2, \dots, t_{h+1})$ también lo son. Para ello basta ver que t_{h+1} es ortogonal a los vectores $(t_1, t_2, \dots, t_h)$.
  - $t_h^t t_{h+1} = t_h^t E_h w_{h+1} = 0$ ya que se demostró antes que $t_h^t E_h = 0 \quad \forall h$
  - $t_{h-1}^t t_{h+1} = t_{h-1}^t E_h w_{h+1} = t_{h-1}^t (E_{h-1} - t_h p_h^t) w_{h+1} = (t_{h-1}^t E_{h-1} - t_{h-1}^t t_h p_h^t) w_{h+1} = 0$

    ya que $t_{h-1}^t E_{h-1} = 0$ y por la hipótesis de partida $t_h^t t_l = 0$

  - $t_{h-2}^t t_{h+1} = t_{h-2}^t E_h w_{h+1} = t_{h-2}^t (E_{h-1} - t_h p_h^t) w_{h+1} = t_{h-2}^t (E_{h-2} - t_{h-1} p_{h-1}^t - t_h p_h^t) w_{h+1} = (t_{h-2}^t E_{h-2} - t_{h-2}^t t_{h-1} p_{h-1}^t - t_{h-2}^t t_h p_h^t) w_{h+1} = 0$

    ya que:

$$t_{h-2}^t E_{h-2} = 0$$

$$t_{h-2}^t t_{h-1} = 0$$

$$t_{h-2}^t t_h = 0$$

    y así sucesivamente obteniéndose el resultado perseguido.

2) $w_h^t E_l^t = 0 \ \ para \ \forall l \geq h$. Los valores residuales obtenidos por el método PLS para una etapa h son ortogonales con respecto a los coeficientes de cualquiera de las componentes anteriores.

$$w_h^t E_l^t = w_h^t (E_{h-1} - t_h p_h^t)^t = w_h^t E_{h-1}^t - w_h^t p_h t_h^t = t_h^t - t_h^t = 0$$

ya que $w_h^t p_h = 1$

A continuación, se demuestra que si se verifica

$$w_h^t E_l^t = 0 \quad \forall l \geq h \ \Rightarrow \ w_h^t E_{l+1}^t = 0$$

$$w_h^t E_{l+1}^t = w_h^t (E_l - p_{l+1} t_{l+1}^t) = w_h^t E_l^t - w_h^t p_{l+1} t_{l+1}^t$$

$$= w_h^t E_l^t - w_h^t \frac{E_l^t t_{l+1}}{t_{l+1}^t t_{l+1}} t_{l+1}^t = 0$$

3) Los coeficientes obtenidos al realizar la regresión entre los residuales y los coeficientes de las componentes de una etapa son ortogonales a los coeficientes de las componentes de cualquier etapa anterior. Es decir, $w_h^t p_l = 0 \ \ para \ l > h$

$w_h^t p_l = w_h^t \frac{E_{l-1}^t t_l}{t_l^t t_l} = 0$ ya que si $l > h$ entonces $l \geq h - 1$, y $w_h^t E_{l-1}^t = 0$ por la propiedad 2.

4) Los coeficientes de las componentes de una etapa son ortogonales a los de cualquier etapa anterior. $w_h^t p_l = 0$ para $l > h$.

$w_h^t p_l = w_h^t \frac{1}{\theta_h} E_{l-1}^t F_l c_l = 0$ ya que si $l > h$ entonces $l \geq h - 1$, y $w_h^t E_{l-1}^t = 0$ por la propiedad 2.

5) Las componentes obtenidas en una etapa cualquiera son ortogonales a los residuos de cualquiera de las variables explicativas de la misma etapa o de etapas superiores: $t_h^t E_l = 0 \ \ \forall l \geq h$.

$t_h^t E_l = t_h^t (E_{l-1} - t_l p_l^t) = t_h^t E_{l-1} - t_h^t t_l p_l^t = t_h^t E_{l-1}$, ya que por la propiedad 1 $t_h^t t_l = 0$.

En general: $t_h^t E_l = t_h^t E_{l-1} = \ldots = t_h^t E_h = 0$ (propiedad vista anteriormente)

6.-Fórmulas de descomposición.

Sin olvidar que el objetivo del análisis PSL es obtener una ecuación que sirva para predecir los valores de las variables a explicar (F_0) según los valores que tomen las variables independientes (E_0).

Las matrices E_0 y F_0 se pueden descomponer por regresión sobre las componentes $t_1, t_2, \dots, t_A$ donde A es el rango de E_0 como:

$$E_0 = t_1\ p^t_1 + t_2\ p^t_2 + \cdots + t_A\ p^t_A$$

$$F_0 = r^t_1 + t_2\ r^t_2 + \cdots + t_A\ r^t_A + F_A$$

Los vectores $t_1, t_2, \dots, t_A$ son ortogonales entre ellos y tendrán por tanto los mismos coeficientes en esta regresión múltiple que en la obtenida mediante el método iterativo anteriormente descrito.

Esta descomposición entraña resultados importantes a la hora de las interpretaciones. Si definimos $\left|\left|E_j\right|\right|^2$ como la suma de los cuadrados de todos los elementos de la matriz E_j y si denotamos E_{0j} a la columna j-ésima de la matriz E_0 (lo que equivale a x_j), se verifica $\left|\left|E_j\right|\right|^2 = < E_{0j}, E_{0j} >$ y:

$$||E_0||^2 = \sum_{j=1}^{p} \left|\left|E_{0j}\right|\right|^2 = \sum_{j=1}^{p} traza\,(\,E^t_{0j} E_{0j}\, t_1\,) = \sum_{j=1}^{p} (\,E^t_{0j} E_{0j}\,)$$

ya que $E^t_{0j} E_{0j}$ es un escalar y por tanto su traza será igual a el producto mismo. Por tanto, sustituyendo E_{0j} por su estimación a partir de la regresión con respecto a las componentes t_i:

$$||E_0||^2 = \sum_{j=1}^{p} (\,(t_1 p_{1j} + t_2 p_{2j} + \cdots + t_A p_{Aj})^t\ (t_1 p_{1j} + t_2 p_{2j} + \cdots + t_A p_{Aj}))$$

$$= \sum_{j=1}^{p} \sum_{h=1}^{A} t^t_h t_h\ p^2_{hj} = \sum_{h=1}^{A} t^t_h t_h \sum_{j=1}^{p} p^2_{hj} = \sum_{h=1}^{A} t^t_h t_h\ ||p_h||^2$$

$$= \sum_{h=1}^{A} ||t_h||^2 ||p_h||^2$$

Siempre se conoce a priori el valor de $\left||\mathrm{E}_0|\right|^2$ porque las variables están centradas y reducidas en E_0. La suma de los cuadrados de cada columna es n-1 (suponer que la tipificación se ha realizado suponiendo que la varianza de una *variable columna* será la suma de dichos cuadrados divida por n-1) y, por lo tanto $\left||\mathrm{E}_0|\right|^2 = (n-1)p$

De esta forma se puede obtener una medida del poder explicativo de cada componente $\mathrm{t_h}$ a partir de:

$$R^2X_h = \frac{\left||t_h|\right|^2\left||p_h|\right|^2}{\left||E_0|\right|^2} = \frac{\left||t_h|\right|^2\left||p_h|\right|^2}{\sum_{u=1}^{A}\left||t_u|\right|^2\left||p_u|\right|^2}$$

O bien la importancia relativa de las primeras z-componentes como:

$$R^2X(acum)_z = \frac{\sum_{h=1}^{z}\left||t_h|\right|^2\left||p_h|\right|^2}{\sum_{z=1}^{A}\left||t_u|\right|^2\left||p_u|\right|^2}$$

También se puede obtener la descomposición de $\left||F_0|\right|^2$(que por las mimas razones explicadas con anterioridad para $\left||\mathrm{E}_0|\right|^2$se sabe que vale (n-1)q):

$$\left||F_0|\right|^2 = \left||t_1|\right|^2\left||r_1|\right|^2 + \left||t_2|\right|^2\left||p_2|\right|^2 + \cdots + \left||t_A|\right|^2\left||p_A|\right|^2 + \left||F_A|\right|^2$$

Pudiendo por tanto medirse el efecto generado por cada componente o bien por las primeras i-componentes para explicar F_0:

$R^2Y_z = \frac{\left||t_z|\right|^2\left||r_z|\right|^2}{\sum_{h=1}^{A}\left||t_h|\right|^2\left||r_h|\right|^2+\left||F_A|\right|^2}$, para cualquier componente $z = 1,2,\dots,A$ y

$R^2y(acum)_z = \frac{\sum_{h=1}^{z}\left||t_h|\right|^2\left||r_h|\right|^2}{\left||F_0|\right|^2} = \frac{\sum_{h=1}^{z}\left||t_h|\right|^2\left||t_h|\right|^2}{\sum_{h=1}^{A}\left||t_h|\right|^2\left||r_h|\right|^2+\left||F_A|\right|^2}$, para las z primeras componentes.

Aunque las reglas de decisión más utilizadas se encuentran relacionados con la validación cruzada que se analizaran posteriormente, existen reglas relativas en cuanto al número de componente a retener en función de estos cálculos, como por ejemplo la siguiente:

Retener las componentes que expliquen más de una determinada proporción de la variabilidad tanto de las variables predictoras (α') como de las dependientes (α), como ocurre en el método de las componentes principales. Es decir, retener el mínimo z que verifique:

$R^2y(acum)_z > \alpha$ y $R^2X(acum)_z > \alpha'$

7.-Interpretación de las componentes PLS.

La matriz de residuos E_h en la etapa h puede expresarse en función de E_0:

$$\begin{aligned} E_h &= E_{h-1} - t_h\, p_h^t = E_{h-1} - E_{h-1} w_h\, p_h^t = E_{h-1}\left(I - w_h\, p_h'\right) \\ &= \left(E_{h-2} - t_{h-1}\, p_{h-1}^t\right)\left(I - w_h\, p_h^t\right) \\ &= \left(E_{h-2} - E_{h-2} w_{h-1}\, p_{h-1}^t\right)\left(I - w_h\, p_h^t\right) \\ &= E_{h-2}\left(I - w_{h-1}\, p_{h-1}^t\right)\left(I - w_h\, p_h^t\right) = \cdots \\ &= E_0\left(I - w_1\, p_1^t\right)\left(I - w_2\, p_2^t\right)\left(I - w_h\, p_h^t\right). \end{aligned}$$

Por tanto, cualquier componente t_h es combinación lineal de las columnas de E_0, ya que:

$$t_h = E_{h-1} w_h = E_0 \prod_{i=1}^{h-1}\left(I - w_i\, p_i'\right) w_h = E_0 \widetilde{w_h}$$

donde el vector $\widetilde{w_h} = \prod_{i=1}^{h-1}\left(I - w_i\, p_i'\right)$ no tiene por qué tener norma 1.

Es decir, las componentes PLS $t_1, t_2, \ldots, t_h$ son combinaciones lineales de las columnas de E_0, no correlacionadas entre ellas, que resumen lo mejor posible la variabilidad de E_0 tratando de explicar lo mejor que les sea posible a F_0.

Para interpretar las componentes PLS t_h en función de x_j e y_k, se pueden calcular la correlación entre una componente cualquiera y las variables tipificadas (utilizando nuevamente como divisor N-1), obteniéndose:

$$Corr\left(t_h, x_j\right) = \frac{\frac{t_h^t E_{0j}}{N-1}}{\sqrt{\frac{t_h^t t_h}{N-1}}} = \frac{\frac{t_h^t\left(t_1 p_{1j} + t_2 p_{2j} + \cdots + t_A p_{Aj}\right)}{N-1}}{\sqrt{\frac{t_h^t t_h}{N-1}}} =$$

(por ser t_i ortogonal con t_j, $\forall i \neq j$ se verifica $t_i^t t_j = 0$)

$$= \frac{\frac{t_h^t t_h\, p_{hj}}{N-1}}{\sqrt{\frac{t_h^t t_h}{N-1}}} = \sqrt{var\left(t_h\right)}\; p_{hj} = \sqrt{\lambda_h}\, p_{hj}$$

donde se ha denotado por λ_h a la var (t_h).

$$Corr(t_h, y_k) = \frac{\frac{t_h^t F_{0k}}{N-1}}{\sqrt{\frac{t_h^t t_h}{N-1}}} = \frac{\frac{t_h^t (t_1 r_{1k} + t_2 r_{2k} + \cdots + t_A r_{Ak} + F_{Ak})}{N-1}}{\sqrt{\frac{t_h^t t_h}{N-1}}}$$

$$= \frac{\frac{t_h^t t_h r_{hk} + t_h^t F_{Ak}}{N-1}}{\sqrt{\frac{t_h^t t_h}{N-1}}} = \sqrt{var\,(t_h)}\, r_{hk} + \frac{t_h^t F_{Ak}}{\sqrt{(N-1)\, t_h^t t_h}}$$

$$= \sqrt{var\,(t_h)}\, r_{hk} = \sqrt{\lambda_h}\, r_{hk}$$

ya que $t_h^t F_{lk} = 0 \;\; \forall l \geq h$

Para interpretar t_1 y t_2, las dos primeras componentes, se construyen los círculos de correlaciones que se obtienen al representar los puntos en un plano:

$$A_j = \left(\sqrt{\lambda_1}\, p_{1j},\ \sqrt{\lambda_2}\, p_{2j} \right), B_k = \left(\sqrt{\lambda_1}\, r_{1k},\ \sqrt{\lambda_2}\, r_{2k} \right)$$

Los productos escalares $< A_j, A_j >$, $< B_k, B_k >$ y $< A_j, B_k >$ representan las aproximaciones de orden dos de las correlaciones (x_j, x_j) , (y_k, y_k) y (x_j, y_k) respectivamente.

La norma de A_j representa la correlación múltiple entre x_j y (t_1, t_2) de la misma forma que la norma B_k representa la correlación múltiple entre y_k y (t_1, t_2).

De esta forma, los círculos de correlaciones indican las variables bien correlacionadas con las dos primeras componentes PLS. Para las variables bien explicadas por t_1 y t_2, el círculo de correlaciones explica tanto las correlaciones internas en cada grupo de variables como las correlaciones entre los grupos.

8.-Ecuaciones de regresión PLS.

En la descomposición de F_0 sobre $t_1, t_2, \ldots, t_A$ se deduce la regresión PLS de cada variable y_k sobre $x_1, x_2, \ldots, x_p$. Por tanto:

$$F_{0k} = \frac{y_k - \overline{y_k}}{s_{y_k}} = \sum_{h=1}^{A} r_{hk} t_h + F_{Ak} = \sum_{h=1}^{A} r_{hk} E_0 \widetilde{w_h} + F_{Ak}$$

$$= \sum_{h=1}^{A} r_{hk} \sum_{j=1}^{p} \widetilde{w_h} \left(\frac{x_j - \overline{x_j}}{s_{x_j}} \right) + F_{Ak} = \sum_{h=1}^{A} \sum_{j=1}^{p} r_{hk} \widetilde{w_h} \left(\frac{x_j - \overline{x_j}}{s_{x_j}} \right) + F_{Ak}$$

Se denota por s_{y_k} y s_{x_j} la desviación típica muestral de la variable y_{k} y x_{j} respectivamente.

Se denota por $\beta_{\mathrm{j}} = \sum_{h=1}^{A} r_{\mathrm{hk}} \widetilde{w_{\mathrm{hj}}}$, entonces se puede en los programas de regresión PLS tres tipos de resultados:

a) Las variables y_{k} y x_{j} son centradas y reducidas.

$$\mathrm{F_{0k}} \approx \sum_{j=1}^{p} \beta_{\mathrm{j}}\, \mathrm{E_{0j}}$$

Para explorar sobre las variables predictoras a eliminar del análisis, se observan precisamente estos datos tipificados. Si los coeficientes de regresión tienen un valor absoluto pequeño producen una contribución pequeña al modelo. Otro estadístico que resume la contribución de una variable al modelo es la importancia de la variable en la proyección (VIP), que se comentará en el próximo apartado. Mientras que el coeficiente de regresión representa la importancia que cada variable explicativa tiene en la predicción de las variables respuesta, la VIP representa el valor de cada variable explicativa en el ajuste del modelo tanto para el conjunto de variables respuesta o dependientes como para las variables predictoras.

b) Si se utilizan las variables reducidas (no centradas) $y'_k = \frac{y_k}{s_{y_k}}$ y $x'_j = \frac{x_j}{s_{x_j}}$

$$y'_k = \frac{\overline{y_k}}{s_{y_k}} + \sum_{j=1}^{p} \beta_{\mathrm{j}}\, x'_j - \sum_{j=1}^{p} \frac{\beta_{\mathrm{j}} - \overline{x_j}}{s_{x_j}} = \beta_0 + \sum_{j=1}^{p} \beta_{\mathrm{j}}\, x'_j$$

$$con\ \ \beta_0 = \frac{\overline{y_k}}{s_{y_k}} - \sum_{j=1}^{p} \frac{\beta_{\mathrm{j}} - \overline{x_j}}{s_{x_j}}$$

c) Utilizando sólo las variables originales:

$$y_k = \overline{y_k} + \sum_{j=1}^{p} \beta_{\mathrm{j}} \frac{s_{y_k}}{s_{x_j}} x_j - \sum_{j=1}^{p} \frac{\beta_{\mathrm{j}} - \overline{x_j}}{s_{x_j}} s_{y_k} = \beta'_0 + \sum_{j=1}^{p} \beta'_j\, x_j$$

donde,

$$\beta'_0 = \overline{y_k} - \sum_{j=1}^{p} \frac{\beta_{\mathrm{j}} - \overline{x_j}}{s_{x_j}} s_{y_k} \qquad \text{y} \qquad \beta'_j = \left(\sum_{h=1}^{A} r_{\mathrm{hk}} \widetilde{w_{\mathrm{hj}}} \right) \frac{s_{y_k}}{s_{x_j}} = \beta_{\mathrm{j}}\, \frac{s_{y_k}}{s_{x_j}}$$

9.-Calidad de reconstrucción de datos activos por el modelo.

Al poder explicativo de una variable x_j sobre el modelo con A componentes se denota por VIP_{A_j}(importancia de la variable x_j en la proyección). Viene definida por:

$$\text{VIP}_{A_j} = \sqrt{\frac{\text{p}}{Rd(Y;\ t_1, t_2, \ldots . t_{\text{A}})\ \sum_{h=1}^{A} Rd(Y;\ t_{\text{h}})}\ w_{hj}^2}$$

donde $\sum_{j=1}^{p} VIP_{Aj}^2 = p\ \forall h$. A representa el número de componentes que se retienen en el modelo.

La contribución de una variable x_j a la construcción de la componente t_{h} se mide por el coeficiente w_{hj}^2. Para cada h la suma de esos valores al cuadrado sobre el conjunto de las p variables explicativas x_j asciende a 1. ($\sum_{j=1}^{p} w_{hj}^2 = 1$)

Para medir la contribución de la variable x_ja la construcción Y a partir de las componentes t_{h} es necesario tener en cuenta el poder explicativo de dicha componente, el cual se mide a través de la redundancia $Rd(Y;\ t_h)$.

Se entiende por redundancia:

$$Rd(Y;\ t_h) = \frac{1}{q}\sum_{k=1}^{q} corr^2(y_{\text{k}}, t_{\text{h}}) = \frac{1}{q}\sum_{k=1}^{q} \lambda_{\text{h}}\ r_{hk}^2$$

No es otra cosa que una medida de las correlaciones al cuadrado entre la componente y las variables que forman el conjunto Y (variable a explicar).

Mientras que,

$$Rd(Y;\ t_1, t_2, \ldots . t_{\text{A}}) = \frac{1}{q}\sum_{h=1}^{A}\sum_{k=1}^{q} corr^2(y_{\text{k}}, t_{\text{h}}) = \frac{1}{q}\sum_{h=1}^{A}\sum_{k=1}^{q} \lambda_{\text{j}} r_{hk}^2$$

Los valores (VIP) permiten clasificar las variables x_j en función del poder explicativo que tiene sobre Y y sobre el espacio de proyección de la nube de puntos. Las variables con un valor grande VIP (>1) son las más importantes en la construcción de Y.

Si una variable explicativa posee un coeficiente de regresión relativamente pequeño en valor absoluto y el VIP es también pequeño (en este caso Wold considera como pequeño cualquier valor VIP inferior a 0.8), entonces se convierte en una candidata a ser eliminada del análisis, aunque en ningún caso éste sea un objetivo perseguido por la regresión PLS.

Antes de eliminar una variable, se pueden representar gráficamente las variables en el plano (w_i, w_k) para i, k componentes retenidas (sobre todo las principales 1, 2), si las variables candidatas a eliminar se encuentran cerca del origen del citado plano entonces tendríamos otro argumento adicional en que basarnos para eliminarlas.

Por tanto, estos tres argumentos:

- $VIP_{A_j} < 0.8$
- β_j pequeño y
- vector *w* próximo al origen.

podrían servir para poder eliminar variables y comenzar nuevamente el algoritmo con las variables restantes. No obstante, como el objetivo principal que persigue la regresión PLS es la predicción de las variables y no la descripción de las relaciones, muchos autores aconsejan no eliminar variables a no ser que se den todas las circunstancias anteriormente descritas y que los valores previstos no varíen excesivamente tras su eliminación.

10.-Estudio de los residuales. Distancia al modelo.

Una vez obtenido el modelo es muy conveniente comprobar si éste predice bien a las variables a explicar. La herramienta estadística con la que se puede maniobrar, como en el análisis de regresión ordinario, es el estudio de los valores residuales (tanto de la muestra con la que hayamos obtenido el modelo como con otra muestra que sirva de comprobación y examen, por supuesto esta última mucho más recomendable siempre que sea posible).

Es conveniente recordar que los residuales de cada observación se obtienen como diferencia entre el valor realmente observado y el valor estimado a partir del modelo.

La regresión PLS tiene como característica peculiar la obtención de dos tipos de residuales: los relativos a las variables explicativas, a los que denominaremos e_{ij} (residual de la observación i-ésima para x_j) y los relativos a las variables respuesta que denotaremos por f_{ik} (residual de la observación i-ésima para y_k), puesto que se obtienen dos regresiones sobre las componentes $t_1, t_2, \ldots . t_A$ (una para las X, cuyos coeficientes eran los p_j y otra para las Y, cuyos coeficientes los denotamos anteriormente como r_k):

$$e_{ij} = x_{ij} - x_{ij}^{*} \qquad \forall i = 1,2,\dots,n \qquad \forall j = 1,2,\dots,p$$

$$f_{ik} = y_{ik} - y_{ik}^{*} \qquad \forall i = 1,2,\dots,n \qquad \forall k = 1,2,\dots,q$$

Como en el método de regresión ordinario es conveniente representar gráficamente los residuos para detectar patrones, estudiar su distribución de probabilidad y compararlos visualmente frente a los valores observados.

A partir de las anteriores medidas individuales (por individuo y variable) se pueden obtener los residuales estandarizados exclusivamente para cada individuo o para cada variable, denominados en ambos casos RSD.

El RSD de una observación en el espacio X o en el espacio Y es proporcional a la distancia de la observación al hiperplano del modelo PLS en el espacio correspondiente. Así para cada individuo se denota por:

$$\mathrm{DModX}_i = \sqrt{\frac{\sum_{j=1}^{p} e_{ij}^2}{p - A}} \,\mathrm{x}\, \sqrt{\frac{n}{n - A - 1}}$$

y mide la distancia al modelo del individuo i en el espacio generado por las variables explicativas.

De la misma forma se obtiene:

$$\mathrm{DModY}_i = \sqrt{\frac{\sum_{k=1}^{p} f_{ik}^2}{q - A}}$$

que representa la distancia al modelo del individuo i en el espacio generado por las variables a explicar.

Se pueden calcular las desviaciones típicas de los residuos e_{ij} y f_{ik} como:

$$\mathrm{s}_x = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{p} e_{ij}^2}{(p - A)(\mathrm{n} - \mathrm{A} - 1)}} \qquad \mathrm{s}_y = \sqrt{\frac{\sum_{i=1}^{n}\sum_{k=1}^{q} f_{ik}^2}{(q - A)(\mathrm{n} - \mathrm{A} - 1)}}$$

y con ello se construye las distancias normalizadas:

$$\mathrm{DModX}_i\,, N = \frac{\mathrm{DModX}_i}{\mathrm{s}_x} \qquad \mathrm{DModY}_i, N = \frac{\mathrm{DModY}_i}{\mathrm{s}_y}$$

La distancia normalizada $\mathrm{DModX}_i\,, N$ elevada al cuadrado sigue una aproximadamente una ley $\mathrm{F}_{k1,k2}$ si los residuos e_{ij} se distribuyen según una ley normal, por lo que puede deducir la probabilidad de que la observación pertenezca al conjunto de datos modelizado.

De esta forma si la probabilidad es muy baja (menor que un valor de referencia α), se puede considerar a las observaciones que verifiquen esa condición como observaciones atípicas.

Si la distancia normalizada DModY_i, N es mayor que 2 también se considera que la observación se encuentra mal reconstituida al nivel del conjunto de variables a explicar.

Así mismo se puede obtener una estimación de la desviación típica de los residuales para una variable que se denota por: $RSD(x_j)$ para una variable predictora x_j o $RSD(y_k)$ para una variable respuesta y_k. Siendo por tanto otra medida de su relevancia en el modelo para la variable predictora y para la variable a explicar.

$$\mathrm{RSD}(\mathrm{x}_j) = \sqrt{\frac{\sum_{i=1}^{n} e_{ij}^2}{n-2}} \qquad \mathrm{RSD}(\mathrm{y}_k) = \sqrt{\frac{\sum_{i=1}^{n} f_{ik}^2}{n-2}}$$

Con estos valores se puede obtener un estimador de la varianza de $\hat{\mathrm{y}}_{ik}$

$$V(\hat{\mathrm{y}}_{ik}) = \frac{\sum_{i=1}^{n} f_{ik}^2}{n-A-1}\left(\sum_{h=1}^{A} \frac{t_{hi}^2}{\mathrm{t}_h^t \mathrm{t}_\mathrm{h}}\right)$$

Por tanto, se puede determinar un intervalo de predicción al nivel (1-α) para el valor de una observación cualquiera para la que se conozca el valor de las variables a explicar ($x_j \;\; j = 1,2,\ldots,p$) y se pretenda estimar el de cualquier variable y_k.Este intervalo será:

$$\left(\hat{\mathrm{y}}_{ik} \pm F_{T_{n-A-1}}^{-1}\left(1-\frac{\alpha}{2}\right) \mathrm{x}\ \mathrm{RSD}(\mathrm{y}_k)\ \mathrm{x}\ \sqrt{1+\frac{1}{\mathrm{n}}+\sum_{h=1}^{A}\frac{t_{hi}^2}{\mathrm{t}_h^t \mathrm{t}_\mathrm{h}}}\right.$$

donde,

- $F_{T_{n-A-1}}^{-1}\left(1-\frac{\alpha}{2}\right)$ es la función cuantil de la función de distribución de la t-Student con n-A-1 grados de libertad para la probabilidad $1-\frac{\alpha}{2}$.
- n es el número de observaciones de la muestra que determinó las ecuaciones de regresión PLS (y por tanto las ecuaciones de previsión)
- t_{hi} es el valor de la componente t_h para las observaciones que se va a predecir (se conoce por ser combinación lineal de las variables X)
- $\mathrm{t}_h^t \mathrm{t}_\mathrm{h}$ es el módulo de la componente t_h para las n observaciones que determinaron el análisis.

11.-Número de componentes a retener por validación cruzada.

Un método bastante extendido para determinar el número de componentes a retener se basa en la validación cruzada. El conjunto de datos disponibles se divide en g agrupaciones de observaciones aproximadamente iguales (suele ser común una observación por agrupación). Se separan los datos en un primero conjunto formado por g-1 agrupaciones (n-1 observaciones generalmente) sobre las que se ajusta el modelo, y un segundo conjunto formado por las agrupaciones restantes (la otra observación no incluida en el primer conjunto) que servirá para comprobar la bondad del análisis, es decir para estimar cual sería el valor de las variables a explicar a partir de los valores de las variables explicativas, utilizando el modelo generado por las g-1 agrupaciones.

Este proceso se repite g veces de forma que cada una de las g agrupaciones sirva una vez y sólo una vez de comprobación. Así se tienen los valores estimados para las variables a explicar para todas las observaciones, sin que ninguna de ellas haya participado en el análisis del modelo con el que se estiman, con la consiguiente reducción de sesgo.

Como se conocen los valores observados para todas las variables y_k, y por validación cruzada se han obtenido estimaciones menos sesgadas de las mismas a través del modelo PLS (en el que no han participado de forma activa en el modelo generado para su estimación), también se conoce por tanto la desviación entre el valor observado y el estimado. Se denomina PRESS_{y_k} a la suma de cuadrados de los errores de la predicción para la variable y_k(desviación al cuadrado entre el valor observado y el estimado), es decir:

$$\text{PRESS}_{y_k} = \sum_{i=1}^{n} (y_{ik} - y^*_{ik(-i)})^2$$

donde $y^*_{ik(-i)}$ es el valor previsto de la variable y_k para la observación i sin que ella haya participado en el modelo.

Al sumar para todas las variables y_k nos produce el estadístico:

$$\text{PRESS}_{(h)} = \sum_{k=1}^{q} \sum_{i=1}^{n} (y_{ik} - y^*_{ik(-i)})^2$$

(reteniendo h componentes en el modelo). Por lo que un buen modelo será aquel que minimice esta cantidad. No obstante, por el *principio de parsimonia* (es principio concluye que si existen dos modelo que solucionan el problema planteado de forma no significativamente diferente es preferible elegir aquel que sea más sencillo por ser más fácilmente interpretable y reducir los riesgos de sobreajuste a la muestra) se debería

buscar aquel modelo que contenga el menos número posible de componentes y que no difiera demasiado del valor mínimo PRESS; en terminología estadística, aquel que no fuera significativamente diferente al mínimo. En base a los valores del estadístico PRESS obtenidos existen varios métodos que nos ayudan a decidir el número de componentes a retener. Cabe destacar:

a) Índices Q^2 y Q^2 acumulado de Stone Geiser.

Para cada nueva componente h y para cada variable y_k se calcula el índice:

$$Q^2_{y_k h} = 1 - \frac{\text{PRESS}_{y_k(h)}}{\text{RSS}_{y_k(h-1)}}$$

donde $\text{RSS}_{y_k(h-1)} = \sum_{i=1}^{n} (y_{ik} - y^*_{ik(h-1)})^2$ representa la suma de cuadrados de los errores (residuales) calculados con el modelo con las h-1 primeras componentes.

Este índice viene a representar la fracción de la variación total de y_k que puede predecirse mediante la componente t_h, es decir, el aporte marginal de cada componente PLS t_h al poder predictivo del modelo para la variable y_k.

Globalmente, sobre el conjunto de variables Y, se calcula:

$$Q^2_h = 1 - \frac{\sum_{k=1}^{q} \text{PRESS}_{y_k(h)}}{\sum_{k=1}^{q} \text{RSS}_{y_k(h-1)}}$$

Este índice mide el aporte marginal de cada componente PLS t_h al poder predictivo del modelo (del conjunto de variables $y_1, y_2, \dots, y_q$).

Se puede medir el aporte global del conjunto de las h primeras componentes PLS, con los siguientes índices:

$$(Q^2_{cum})_{y_k h} = 1 - \prod_{z=1}^{h} \frac{\text{PRESS}_{y_k z}}{\text{RSS}_{y_k(z-1)}} \quad (\textit{índice para la variable } y_k \; k = 1,2, \dots, q).$$

$$(Q^2_{cum})_h = 1 - \prod_{z=1}^{h} \frac{\sum_{k=1}^{q} \text{PRESS}_{y_k(z)}}{\sum_{k=1}^{q} \text{RSS}_{y_k(z-1)}}$$

Se utilizan las siguientes reglas de decisión que nos permiten decidir si el aporte de una componente t_h es significativo.

1) El aporte de la componente t_h es significativo si

$$Q_h^2 \geq (1 - 0.95^2) = 0.0975$$

2) El aporte de la componente t_h es significativo si al menos para una variable y_j se verifica:

$$Q_{hj}^2 \geq (1 - 0.95^2) = 0.0975$$

Además, una variable y_k se considera bien modelizada por h componentes $t_1, t_2, \dots, t_h$ si: $(Q_{cum}^2)_{jh} \geq 0.5$

Estos límites son arbitrarios, pero corresponden a la experiencia del creador de la teoría de la regresión PLS Svante Wold.

b) Criterio de Van der Voet.

Van der Voet ha propuesto un índice para comparar la suma de los residuales previstos al cuadrado ($\text{PRESS}_{(h)}$) para diferentes modelos (según el número de componentes retenidas). Supongamos que C es el número de componentes que verifica:

$$\text{PRESS}_{(C)} = Min_h \text{PRESS}_{(h)}$$

Es decir, el modelo que retiene sólo C componentes es el que minimiza la suma de cuadrados de los errores basándose en el método de validación cruzada.

El valor crítico del test de Van der Voet se basa en las diferencias entre cada modelo de h componentes y el modelo con 'C' componentes que minimiza la suma de cuadrados de los residuales previstos. A esa diferencia la denominamos:

$$C_{(h)} = \text{PRESS}_{(h)} - \text{PRESS}_{(C)}$$

Virtualmente, el nivel de significación para el test de Van der Voet se obtiene comparando $C_{(h)}$ con la distribución de los valores que resultan de intercambiar aleatoriamente los residuales $f_{(h)ik}^2$ y $f_{(C)ik}^2$. En la práctica se simula una muestra y el nivel de significación es aproximadamente la proporción de valores simulados que son mayores que $C_{(h)}$. Si el nivel de significación es mayor que un nivel α se considera que no hay diferencias por lo que se deduce que con este método elegiríamos el menor valor de h cuyo nivel de significación fuera mayor que α. Es decir, cuya diferencia con respecto al modelo que minimiza la suma de cuadrados de los errores prevista no es significativa.

12.-Otros algoritmos sobre PLS. Algoritmo NIPALS.

Existen otros algoritmos difundidos sobre PLS, algunos vienen determinados por técnicas diferentes que en algunas circunstancias coinciden con el algoritmo PLS.

Mención especial por su desarrollo y por ser utilizados en los programas estadísticos; como en que se detallará con posterioridad, tiene el algoritmo NIPALS. Los principios de este algoritmo sirvieron de base de la regresión PLS que se ha desarrollado anteriormente (PLS1 y PLS2). Como se menciono al principio este algoritmo lo creo Wold para obtener componentes principales con el nombre de NILES. Como se indica no necesita suprimir ni estimar los datos faltantes de una observación para que ésta se utilice en el análisis. Su comprensión es muy recomendable por que tiene a demostrar cómo se extraen componentes principales (o se factoriza una matriz) a partir de una serie de regresiones simples por mínimos cuadrados. Por esta razón se procede a detallarlo.

Algoritmo NIPALS.

Se dispone de una matriz $X = \{x_{ij}\}$ de individuos por variables de rango a. Las columnas de X se denotarán por $x_1, x_2, \dots, x_p$; vamos a suponer que las citadas columnas se encuentran centradas. La fórmula de descomposición del análisis de componentes principales se escribe:

$$X = \sum_{h=1}^{a} t_h p_h'$$

donde los vectores $t_h = (t_{1h}, t_{2h}, \dots, t_{nh})'$ y $p_h = (p_{h1}, p_{h2}, \dots, p_{hp})'$ son respectivamente las componentes principales y los vectores directores de los ejes principales. Las variables x_j se expresan en función de las componentes $t_1, t_2, \dots, t_a$:

$$x_j = \sum_{h=1}^{a} p_{hj} t_h \qquad j = 1,2, \dots, p$$

la i-ésima línea (observación) de X se denota por $x_i^t = (x_{i1}, x_{i2}, \dots, x_{ip})$, los individuos x_i se pueden expresar por tanto también en función de los vectores $p_1, p_2, \dots, p_a$:

$$x_i = \sum_{h=1}^{a} t_{ih} p_h \qquad i = 1,2, \dots, n$$

La doble ortogonalidad de las componentes principales y de los vectores directores es característica del análisis de componentes principales. De ahí se deduce que:

- p_{ij} es el coeficiente de regresión de t_1 en la regresión de x_j sobre t_1.
- t_{1i} es el coeficiente de regresión de p_1 en la regresión de x_i sobre p_1.
- Para $h > 1$ p_{hj} es el coeficiente de regresión de t_h en la regresión de $x_j - \sum_{l=1}^{h-1} p_{lj} t_1$ sobre t_h.
- Para $h > 1$ t_{hi} es el coeficiente de regresión de p_h en la regresión de $x_i - \sum_{l=1}^{h-1} t_{li}\ p_1$ sobre p_h.

Se puede considerar también de la descomposición $X = \sum_{h=1}^{a} t_h\, p_h'$ como un modelo y los parámetros p_{hj} y t_{hi} como cantidades a estimar. Wold propuso una búsqueda iterativa de estos parámetros.

Para $h = 1$ se obtiene una solución (p_1, t_1) tal que p_{1j} es la pendiente de la recta de mínimos cuadrados de la nube de puntos (t_1, x_j) en R'' y t_{1i} es la pendiente de la recta de mínimos cuadrados pasando por el origen de la nube de puntos (p_1, x_i) en R^p.

Para $h > 1$ se obtiene una solución (p_h, t_h) tal que p_{hj} es la pendiente de la recta de mínimos cuadrados de la nube de puntos $(t_h, x_j - \sum_{l=1}^{h-1} p_{lj} t_l)$ y t_{hi} es la pendiente de la recta de mínimos cuadrados pasando por el origen de la nube de puntos $(p_h, x_i - \sum_{l=1}^{h} t_{li} p_h)$.

Cuando no existen datos ausentes, NIPALS conduce el análisis de componentes principales tradicional. Cuando hay datos ausentes se obtienen estimaciones de las componentes t_h y de los vectores p_h que permiten describir la matriz X y estimar los datos ausentes.

De esta forma el algoritmo NIPALS permite estimar los parámetros de un modelo no lineal (bilineal) con la ayuda de una serie de regresiones simples entre los datos y una parte de los parámetros. De ahí la significación completa del término NIPALS (Nonlinear estimation by Iterative Partial Least Square)

Descripción del algoritmo NIPALS.

Etapa 1: $X_0 = X$

Etapa 2: Para $h = 1,2, \dots, a$:

2.1: t_h es la primera columna de X_{h-1}.

2.2: se repite hasta que converja p_h:

2.2.1:

$$p_h = \frac{X_{h-1}^t t_h}{t_h^t t_h}$$

2.2.2: Normaliza p_h a norma 1.

2.2.3:

$$t_h = \frac{X_{h-1} p_h}{p_h^t p_h}$$

2.3: $X_h = X_{h-1} - t_h p_h^t$.

Las relaciones cíclicas de la etapa 2.2 muestran que en el límite los vectores p_h y t_h verifican las ecuaciones siguientes:

$$\frac{1}{n-1} X_{h-1}^t X_{h-1} p_h = \lambda_h p_h$$

$$\frac{1}{n-1} X_{h-1} X_{h-1}^t t_h = \lambda_h t_h$$

donde el valor λ_h es el mayor autovalor común a estas matrices. Como p_h esta normado se da la igualdad $\lambda_h = \frac{1}{n-1} t_h^t t_h$ (varianza máxima, se puede también dividir por n en lugar de n-1).

Por tanto, la etapa 2.2 corresponde a una aplicación del *método de potencia iterada* para el cálculo del vector propio de una matriz asociado a su mayor autovalor (Hotelling 1936).

La etapa 2.3 corresponde al cálculo del residuo X_h de la regresión X_{h-1} sobre t_h.

Para $h = 1$ se obtiene el primer eje factorial p_1 y la primera componente principal t_1. La matriz $X_1 = X - t_1 p_1^t$ representan el residuo de la regresión de X sobre la primera componente principal t_1. La matriz $\frac{1}{n-1} X_1^t X_1$ se puede escribir como:

$$\frac{1}{n-1} X_1^t X_1 = \frac{1}{n-1}(X - t_1 p_1^t)(X - t_1 p_1^t) = \frac{1}{n-1} X^t X - \lambda_1 p_1 p_1^t$$

Por tanto, para $h = 2$ el vector propio p_2 de la matriz $\frac{1}{n-1} X_1^t X_1$ asociado a su mayor valor propio se corresponde con el vector propio de $\frac{1}{n-1} X^t X$ asociado al segundo mayor autovalor de dicha matriz.

De manera general la matriz $\frac{1}{\mathrm{n}-1}\ \mathrm{X}^{t}_{h-1}\mathrm{X}_{\mathrm{h}-1}$ se puede escribir como:

$$\frac{1}{\mathrm{n}-1}\ \mathrm{X}^{t}_{h-1}\mathrm{X}_{\mathrm{h}-1} = \frac{1}{\mathrm{n}-1}X^{t}X - \lambda_1 \mathrm{p}_1\ \mathrm{p}^{t}_{1} - \lambda_2 \mathrm{p}_2\ \mathrm{p}^{t}_{2} - \cdots - \lambda_{h-1}\mathrm{p}_{\mathrm{h}-1}\ \mathrm{p}^{t}_{h-1}$$

y el autovector p_h de la matriz $\frac{1}{\mathrm{n}-1}\ \mathrm{X}^{t}_{h-1}\mathrm{X}_{\mathrm{h}-1}$ asociado a su mayor autovalor se corresponde con el autovector de $\frac{1}{\mathrm{n}-1}X^{t}X$ asociado al h-ésimo mayor autovalor λ_h.

De esta forma el problema del análisis en componentes principales se puede resolver por una serie de regresiones simples locales.

No obstante, el interés fundamental del algoritmo NIPALS se manifiesta más claramente en presencia de datos ausentes, en cuyo caso resultan las siguientes etapas:

Etapa 1: $\mathrm{X}_0 = X$

Etapa 2: Para $h = 1,2,\dots,a$:

2.1: t_h es la primera columna de $\mathrm{X}_{\mathrm{h}-1}$.

2.2: se repite hasta que converja p_h:

2.2.1: Para $j = 1,2,\dots,p$: $\mathrm{p}_{\mathrm{hj}} = \dfrac{\sum_{\{i:\mathrm{x}_{ji}y\mathrm{t}_{hi}existen\}} \mathrm{x}_{(h-1)ji}\mathrm{t}_{\mathrm{hi}}}{\sum_{\{i:\mathrm{x}_{ji}y\mathrm{t}_{hi}existen\}} \mathrm{t}^2_{hi}}$

2.2.2: Normaliza p_h a norma 1.

2.2.3: Para $i = 1,2,\dots,n$: $\mathrm{t}_{\mathrm{hi}} = \dfrac{\sum_{\{j:\mathrm{x}_{ji}y\mathrm{t}_{hi}existen\}} \mathrm{x}_{(h-1)ji}\mathrm{p}_{\mathrm{hj}}}{\sum_{\{j:\mathrm{x}_{ji}y\mathrm{t}_{hi}existen\}} \mathrm{p}^2_{hj}}$

2.3: $\mathrm{X}_\mathrm{h} = \mathrm{X}_{\mathrm{h}-1} - \mathrm{t}_\mathrm{h}\ \mathrm{p}^{t}_{h}$.

La idea más importante del algoritmo NIPALS reside en la interpretación de las etapas 2.2.1 y 2.2.3 donde se calculan las pendientes de las rectas mínimo cuadráticas, pasando por el origen, de las nubes de puntos sobre los datos disponibles, $\{(\mathrm{t}_{hi}, \mathrm{x}_{(h-1)ji}),\ i = 1,2,\dots,n\}$ donde t_{hi} y x_{ji} existen y $\{(\mathrm{p}_{hj}, \mathrm{x}_{(h-1)ji}),\ j = 1,2,\dots,p\}$ donde x_{ji} existe. La salida del algoritmo puede proporcionar los pseudo-valores propios de λ_h siempre definidos por la varianza de la componente $\mathrm{t}_h(\frac{1}{\mathrm{n}-1}\ \mathrm{t}^{t}_{h}\mathrm{t}_1)$.

Además, el algoritmo NIPALS permite estimar los datos ausentes utilizando la fórmula de reconstitución habitual:

$$\hat{x}_{ji} = \sum_{k=1}^{h} t_{ki}\, p_{kj}$$

(j es la variable, i la observación y k la componente). No obstante, el algoritmo funciona eficientemente sin estimación de datos ausentes.

En la práctica la convergencia del algoritmo se asegura siempre que no haya demasiados datos ausentes.

CAPITULO 6: REGRESION PLS EN R.

1.-Introducción.

El paquete pls (*cran.r-project.org/web/packages/pls)* implementa Regresión de Componentes Principales (PCR) y Partial Least Square Regression (PLSR) en R, y está disponible gratuitamente en el sitio web de CRAN. El paquete implementa PCR y varios algoritmos para PLSR, describiremos el paquete y cómo se utiliza para el análisis de datos; así como puede ser utilizado como parte de otros paquetes.

En este apartado se explica un ejemplo para obtener una visión general del paquete. Se describen fórmulas y marcos de datos como se utilizan en pls. Analizando cómo se ajustan los modelos y la elección del mismo por validación cruzada; inspeccionando y trazando el modelo y por último se realizan predicciones de futuras observaciones.

2.-Ejemplo.

En la práctica apenas hay diferencias entre el uso de PLSR y PCR; en la mayoría de los casos, los métodos consiguen precisiones de predicción similares, aunque PLSR por lo general necesita menos variables latentes que PCR, es decir, con el mismo número de variables latentes, PLSR cubrirá más la variación en Y y PCR cubrirá más la variación en X.

Hay que tener en cuenta que en algunos casos PLSR parece aumentar la varianza de los coeficientes de regresión individuales, por lo que no siempre es mejor que PCR.

Se analiza con un ejemplo para obtener una visión general del paquete.

Se carga el paquete para poder utilizarlo con el siguiente argumento:

```
library(pls)
```

Con esta función se carga el paquete pls en R; existen tres ejemplos de conjuntos de datos que se incluyen en este paquete:

yarn Un conjunto de datos con 28 espectros de infrarrojo cercano (NIR) de hilos PET, medidos a 268 ondas longitudinales como predictores y la densidad como respuesta (**density**). El conjunto de datos también incluye una variable lógica **train** que puede utilizarse para dividir los datos en un subconjunto de datos de tamaño 21 y un subconjunto de datos denominados **test** de tamaño 7.

oliveoil Un conjunto de datos con 5 mediciones de calidad (**chemical**) y panel sensorial de 6 variables paneles (**sensory**) hechas en 16 muestras de aceite de oliva.

gasoline Un conjunto de datos consistentes en el índice de octano (**octane**) y el espectro NIR (NIR) de 60 muestras de gasolina. Cada espectro NIR consta de 401 mediciones de reflectancia de 900 a 1700nm.

Estos conjuntos de datos se utilizarán en los siguientes ejemplos; para ello primero se ha de cargar con los siguientes comandos:

```
data("yarn")
data("oliveoil")
data("gasoline")
```

A partir de ahora se supone que el paquete y los conjuntos de datos han sido cargados como se ha indicado. En este apartado, se realiza un PLSR (regresión PLS) con los datos de gasolina para ilustrar el uso de PLS.

Se comienza dividiendo el conjunto de datos en datos entrenamiento (**gastrain**) y conjunto de datos prueba (**gastest**) con los siguientes comandos:

```
gastrain<-gasoline[1:50,]
gastest<-gasoline[51:60,]
```

Para ajustar los datos a un modelo PLSR se realiza usando el comando:

```
gas1<-plsr(octane~ NIR, ncomp=10, data=gastrain, validation="LOO")
```

Con ello se indica el ajuste a un modelo con 10 componentes, incluyendo leave-one-out (LOO) por validación cruzada. Se puede obtener una visión general de los resultados de ajuste y validación con el comando:

```
summary(gas1)
```

Obteniendo los siguientes resultados:

```
Data:   X dimension: 50 401
        Y dimension: 50 1
Fit method: kernelpls
Number of components considered: 10
```

```
VALIDATION: RMSEP
Cross-validated using 50 leave-one-out segments.

      (Intercept)   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
CV          1.545     1.357   0.2966   0.2524   0.2476   0.2398   0.2319   0.2386   0.2316   0.2449    0.2673
adjCV       1.545     1.356   0.2947   0.2521   0.2478   0.2388   0.2313   0.2377   0.2308   0.2438    0.2657

TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X         78.17    85.58    93.41    96.06    96.94    97.89    98.38    98.85    99.02     99.19
octane    29.39    96.85    97.89    98.26    98.86    98.96    99.09    99.16    99.28     99.39
```

Los resultados de la validación son Root Mean Squared of Prediction (RMSEP). Existen dos estimaciones de validación cruzada: CV es la estimación CV común y adjCV es un sesgo corregido de la estimación CV. (Para una validación cruzada LOO, prácticamente no hay diferencias).

A menudo es más fácil analizar los RMSEP por gráficos, usamos para ello el siguiente comando:

```
plot(RMSEP(gas1), legendpos = "topright")
```

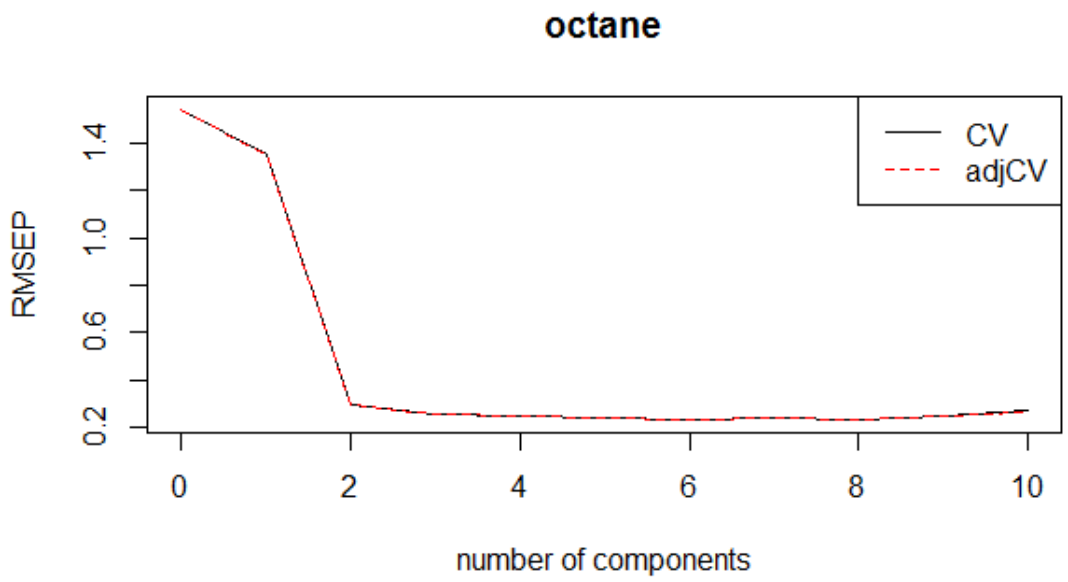

Figura1: Curvas RMSEP con validación cruzada para los datos de gasolina

Este gráfico muestra los RMSEP como funciones del número de componentes. Con el argumento **legendpos** se indica al gráfico que añada una leyenda en la posición indicada. Se puede analizar con el gráfico que dos componentes parecen ser suficiente ya que se obtiene un RMSEP de 0.297. Como se ha indicado anteriormente, la principal diferencia entre PCR y PLSR es que la PCR a menudo necesita más componentes que PLSR para

lograr el mismo error de predicción. En este caso en concreto, la PCR necesitaría tres componentes para lograr el mismo RMSEP.

Una vez elegido el número de componentes, se pueden analizar los diferentes aspectos del ajuste mediante el gráfico de las predicciones, residuos…etc. El gráfico de predicción se puede obtener con el siguiente comando:

```
plot(gas1, ncomp=2, asp=1, line=TRUE)
```

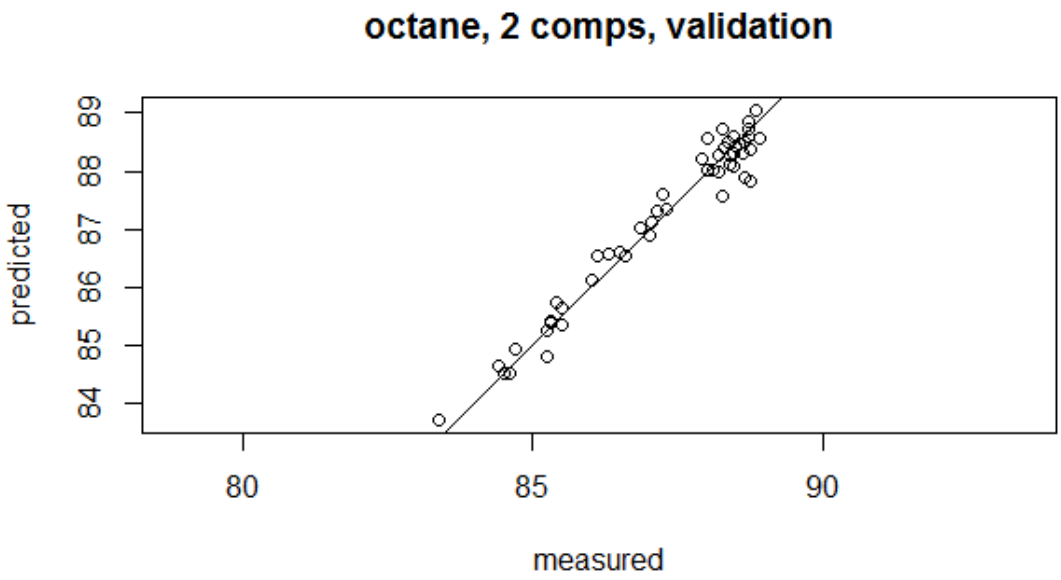

Figura 2: gráfico para las predicciones de los datos de gasolina.

En este gráfico se muestran las predicciones con validación cruzada con dos componentes en comparación con los valores medidos. Se ha seleccionado una relación asp=1 y se indica que aparezca la línea objetivo. Se puede observar que los puntos se adaptan muy bien a la línea objetivo y no se aprecian curvaturas u otras anomalías.

También se puede realizar el gráfico de los residuos usando el argumento plottype de la siguiente forma:

```
plot(gas1, plottype = "scores", comps=1:3)
```

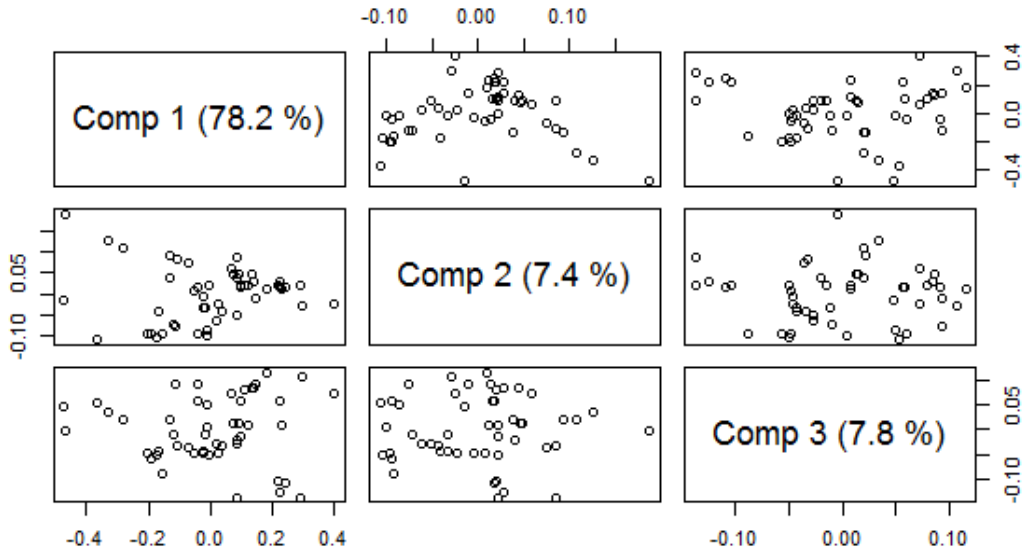

Figura 3: Gráfico de residuos para los datos de gasolina.

Se obtiene un gráfico de residuos para las tres primeras componentes. Los gráficos de residuos se usan a menudo para buscar patrones o valores atípicos en los datos. En este ejemplo no existen indicios de valores atípicos. Los datos que aparecen entre paréntesis después de cada componente corresponden al porcentaje de varianza explicada por cada componente para la variable X. También se puede obtener explícitamente la varianza explicada con el siguiente comando:

```
explvar(gas1)
```

Obteniendo los siguientes resultados:

```
 Comp 1    Comp 2     Comp 3    Comp 4    Comp 5    Comp 6    Comp 7    Comp 8    Comp 9    Comp 10
78.17076   7.41222    7.82415   2.65777   0.87682   0.94663   0.49215   0.47232   0.16882   0.16937
```

Es frecuente utilizar el modelo para predecir los valores de respuesta de las nuevas observaciones. Con el siguiente comando se realiza una predicción para las diez observaciones en gastest, utilizando dos componentes:

```
predict(gas1, ncomp = 2, newdata = gastest)
```

Obteniendo los siguientes resultados:

```
, , 2 comps

     octane
51 87.94125
52 87.25242
53 88.15832
```

```
54 84.96913
55 85.15396
56 84.51415
57 87.56190
58 86.84622
59 89.18925
60 87.09116
```

Debido a que se conocen los verdaderos valores de respuesta para estas muestras, se puede calcular el conjunto de pruebas RMSEP con el siguiente comando:

```
RMSEP(gas1, newdata=gastest)
```

Obteniendo los siguientes resultados:

```
(Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
  1.5369     1.1696   0.2445   0.2341   0.3287   0.2780   0.2703   0.3301   0.3571   0.4090   0.6116
```

Para dos componentes, obtendremos 0.244, que es bastante cercano a la estimación indicada anteriormente de 0.297.

3.-Fórmulas y marco de datos.

El paquete pls tiene unas fórmulas que funcionan de forma similar, en la mayoría de los casos, a la regresión lineal simple en R denominada lm. En este apartado se indican descripciones de fórmulas y marco de datos (data frame) para pls.

3.1-Fórmulas.

Una fórmula consiste en un *left hand side* (lhs), una tilde (~) y un *right hand side* (rhs). Los lhs consisten en un solo término, que representa la respuesta. El rhs consiste en uno o más términos separados por +, representando la variable explicativa. Por ejemplo, en la fórmula a~b+c+d, a es la variable respuesta y b, c y d son las variables explicativas. El término *intercept* se incluye automáticamente por lo que no es necesario especificarlo en la fórmula.

Cada término representa una matriz, un vector numérico o un factor (se debe tener en cuenta que la variable respuesta no debe ser un factor). Si el término respuesta puede ser una matriz obteniendo así un modelo multidimensional. También es posible usar transformaciones de las variables como, por ejemplo, log(y)~Z que especifica una regresión del logaritmo de y sobre la variable Z. Si las transformaciones contienen

símbolos, (+,* o ^) lo términos de ir acompañados de la función I() de la siguiente forma:y~x1 + I(x2+x3). En este caso se indican dos variables respuesta, una x1 y otra la suma de x2+x3.

3.2-Marco de datos (Data frame).

Las funciones de ajustes de modelos buscan primero las variables especificadas en un marco de datos suministrados, y es aconsejable recoger todas las variables de este modo. De esta forma se hace más fácil saber qué datos se ha utilizado para ajustar, para mantener variantes diferentes de los datos y para predecir nuevos datos.

Para crear un marco de datos, se puede usar la función data.frame: si v1, v2 y v3 son factores o vectores numéricos, mydata <-data.frame (y=v1, a=v2, b=v3) dará como resultado un marco de datos con variables denominada y, a y b.

PLSR y PCR a menudo se utilizan con una matriz como el término único predictor. Además, los modelos de respuesta múltiple requieren una matriz como respuesta. Si Z es una matriz, tiene que ir acompañada por la "*protect function*" I() cada vez que se haga mención en la función data.frame: mydata<-data.frame(…, Z=I(Z)). De lo contrario, separa las variables para cada columna, y no habrá ninguna variable llamada Z en el marco de datos, por lo que no se puede usar Z en la fórmula. También se puede añadir a un marco de datos existente:

Mydata<- data.frame(…)

Mydata$Z<-Z

De esta forma también se evita que Z se divida en variables separadas. Finalmente, también se puede hacer mención al comando **cbind** con el que se puede combinar vectores y matrices. Es muy útil en la variable respuesta, por ejemplo, cbind(y1, y2)~X.

Las variables en un marco de datos se pueden acceder con el operador $, por ejemplo, mydata$y. Sin embargo, las funciones en pls acceden a las variables automáticamente, por lo que el usuario nunca debe usar $ en las fórmulas.

4.-Ajustes de modelos.

Las funciones principales para ajustar los modelos son **pcr** y **plsr**. Se va a utilizar plsr en los ejemplos en este apartado, pero se podría haber usado igualmente las funciones **pcr** (o **mvr**).

En su forma más simple, la función para ajustar los modelos es **plsr** (fórmula, ncomp, datos). La fórmula de argumento es una fórmula como se ha descrito anteriormente, ncomp es el número de componentes que se desea ajustar, y los datos son el marco de datos que contienen las variables a utilizar en el modelo. La función devuelve un modelo ajustado que puede ser analizado o utilizado para predecir nuevas observaciones. Por ejemplo:

```
dens1<-plsr(density~NIR, ncomp=5, data=yarn)
```

Si el término de respuesta de la fórmula es una matriz, se ajusta un modelo de respuesta múltiple, por ejemplo:

```
dim(oliveoil$sensory)
```

```
[1] 16  6
```

```
plsr(sensory~chemical, data=oliveoil)
```

```
Partial least squares regression , fitted with the kernel algorithm.
Call:
plsr(formula = sensory ~ chemical, data = oliveoil)
```

Se puede comprobar que la salida nos indica que tipo de modelo es y como se denomina la función de ajuste. El argumento ncomp es opcional; si se omite, toma por defecto el número máximo de componentes. También los datos son opcionales, y si se omiten, las variables especificadas en la fórmula se buscan en el entorno global (espacio de trabajo del usuario). Por lo general, es preferible mantener las variables en los marcos de datos, pero a veces puede ser conveniente tenerlos en un entorno global. Si las variables residen en un marco de datos, por ejemplo, **yarn**, no se usa la fórmula "yarn$density~yarn$NIR" se utiliza **density~NIR** y se especifica el marco de datos con **data=yarn** como se ha indicado.

Para utilizar sólo una parte de las muestras en un conjunto de datos, los primeros 20, se puede usar el comando **subset=1:20** o bien **data=yarn[1:20,].** Además, si se desea probar diferentes alternativas del modelo se puede hacer con el comando **update:**

```
trainind<-which(yarn$train==TRUE)
```

```
dens2<-update(dens1, subset=trainind)
```

Vuelve a montar el modelo **dens1** usando solo las observaciones que están marcadas con TRUE en **yarn$train** y

```
dens3<-update(dens1, ncomp=10)
```

Cambiará el número de componentes a 10. Otros argumentos, como la fórmula, también pueden ser cambiados por actualizaciones.

La ausencia de datos a veces puede ser un problema, los algoritmos PLSR y PCR actualmente implementados en pls no trabajan los valores ausentes intrínsecamente, por lo que las observaciones ausentes deben ser eliminadas. Esto se puede hacer con el argumento **na.action**. Usamos **na.action=Na.omit** (el valor predeterminado), cualquier observación con valores ausentes se eliminará del modelo completamente. Con **na.action=na.exclude**, se eliminarán del proceso de ajuste, pero se incluyen como NA en los valores residuales y ajustados. Si se necesita un error explícito cuando exista ausencia de datos, se usa na.action=na.fail. El na.action predeterminado se puede establecer con options(), por ejemplo, options(na.action=quote(na.fail)).

A menudo, se necesitan estandarización y otros pretratamientos de las variables predictoras; en pls, las variables predictoras siempre están centradas, como parte del algoritmo de ajuste. La escala se establece con el argumento scale. Si la escala es TRUE, cada variable es estandarizada por estar divida por su desviación estándar, y si la escala es un vector numérico, cada variable se divide por el número correspondiente. Por ejemplo:

```
olive1<-plsr(sensory~chemical, scale=TRUE, data=oliveoil)
```

Como se mencionó anteriormente, **msc** se implementa en pls como una función msc que se puede utilizar en fórmulas:

```
gas2<-plsr(octane~msc(NIR), ncomp=10, data=gastrain)
```

Esta dispersión corrige NIR antes de la adaptación, y se encarga de que los nuevos espectros sean automáticamente scatter corregido (usando el mismo espectro de referencia que cuando se ajustan) con la función predict sería:

```
predict(gas2, ncomp=3, newdata = gastest)
```

Hay otros argumentos que se pueden usar como son: **validation** es para seleccionar y también el argumento **mvrCv.**

5.-Elegir el número de componentes por Validación Cruzada.

Validación cruzada, se utiliza con frecuencia para determinar el número óptimo de componentes a tener en cuenta, se controla mediante el argumento **validation** en las

funciones de modelado (mvr, plsr y pcr). El valor predeterminado es **"none"** los argumento "CV" o "LOO" se usan en el procedimiento de modelado con el argumento

mvrCv para realizar validación cruzada. "LOO" utiliza validación cruzada dejando un caso fuera, mientras que "CV" divide los datos en segmentos. Por defecto utiliza diez segmentos, seleccionados aleatoriamente, pero también segmentos de objetos consecutivos o segmentos intercalados, solo hay que indicarlo en el argumento **segment.type.**

Cuando se realiza la validación del modelo de esta forma, el modelo contendrá un elemento que comprende información sobre las predicciones fuera de la bolsa (en forma de valores predichos, como por ejemplo el MSEP y R^2). Los resultados de la validación se pueden visualizar usando el argumento **plottype="validation"** en la función de gráficos estándar. Como ejemplo lo se puede visualizar en la Figura 1 para los datos de gasolina; normalmente se seleccionaría un número de componentes y se puede comprobar que el error de validación cruzada no muestra una disminución significativa.

La decisión de cuántos componentes seleccionar será en cierta medida subjetiva; sin embargo, especialmente cuando se construye un gran número de modelos (como ejemplo, en estudios de simulación), puede ser crucial tener una estrategia consistente en cómo elegir el número "optimo" de componentes.

Cuando se aplica un pretratamiento que depende de la composición del conjunto de entrenamiento, el procedimiento de validación cruzada como se ha descrito anteriormente no es óptimo, ya que, los errores de validación están sesgados por debajo. Siempre y cuando el único propósito sea seleccionar el óptimo número de componentes, este sesgo no se considera muy importante, pero tampoco sería demasiado difícil de evitar. Las funciones de modelado tienen un argumento **scale** que puede ser usado para los segmentos; sin embargo, los métodos más elaborados como MSC necesitan usar los segmentos de formas más explícita. Para esto, la función **crossval** es muy válida, toma un mvr object y realiza cros-validation, aplicando en pretratamiento para cada segmento. Los resultados se pueden analizar en una gráfica similar a la figura 1 o bien ver el resumen de datos.

```
gas2.cv<-crosval(gas2, segments=10)

plot(MSEP(gas2.cv), legendpos="topright")

summary(gas2.cv, what="validation")
```

```
Data:   X dimension: 50 401
        Y dimension: 50 1
Fit method: kernelpls
Number of components considered: 10
```

```
VALIDATION: RMSEP

Cross-validated using 10 random segments.
(Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps 8 comps  9 comps  10 comps
CV     1.545   1.303   0.2824   0.2538   0.2377   0.2358   0.2469   0.2442 0.2559   0.2736    0.2886
adjCV 1.545   1.301   0.2806   0.2529   0.2347   0.2323   0.2410   0.2388 0.2492   0.2646    0.2777
```

Aplicar MSC en este caso conduce a casi idénticas predicciones por validación cruzada. Cuando la escala no depende de la división de los datos en segmentos, las funciones crossval y mvrCv dan los mismos resultados; sin embargo, crossval es mucho más lento.

6.-Análisis de ajuste de modelo.

La validación cruzada puede ser computacionalmente exigente (especialmente cuando se utiliza la función crossval). Un análisis más exhaustivo al ajuste del modelo puede revelar interesantes acuerdos o desacuerdo con las informaciones sobre las relaciones entre X e Y.

6.1-Gráfico.

Se puede acceder a todas las funciones de trazado a través del argumento **plottype** de la función plot. La gráfica por defecto es una gráfica de predicción (predplot), que muestra la predicción de valores sobre los valores medidos. Las predicciones de conjuntos de pruebas se utilizan si se suministra un conjunto de pruebas con el argumento newdata. Por otra parte, si el modelo fue construido utilizando validación cruzada, se utilizan las predicciones de validación cruzada, de lo contrario pueden anularse las predicciones del conjunto de entrenamiento.

Para evaluar cuántos componentes son óptimos, un diagrama de validación (**validationplot**) se puede usar como se muestra en la figura 1. Es una muestra de medida del rendimiento de las predicciones (RMSEP, MSEP, R^2) frente al número de componentes. En general, seleccionamos el primer mínimo local en lugar de elegir el mínimo absoluto en la curva para evitar un ajuste excesivo.

Los coeficientes de regresión se pueden visualizar usando **plottype=”coef”** o bien directamente a través de la función **coefplot.** Esto permite trazar simultáneamente la regresión de vectores para varios números diferentes de componentes a la vez. Los vectores de regresión para el conjunto de datos de gasolina utilizando msc se muestran en

la figura 4 usando el comando:

```
plot (gas1, plottype = "coef", ncomp=1:3, legendpos="bottomleft",
      labels="numbers", xlab="nm")
```

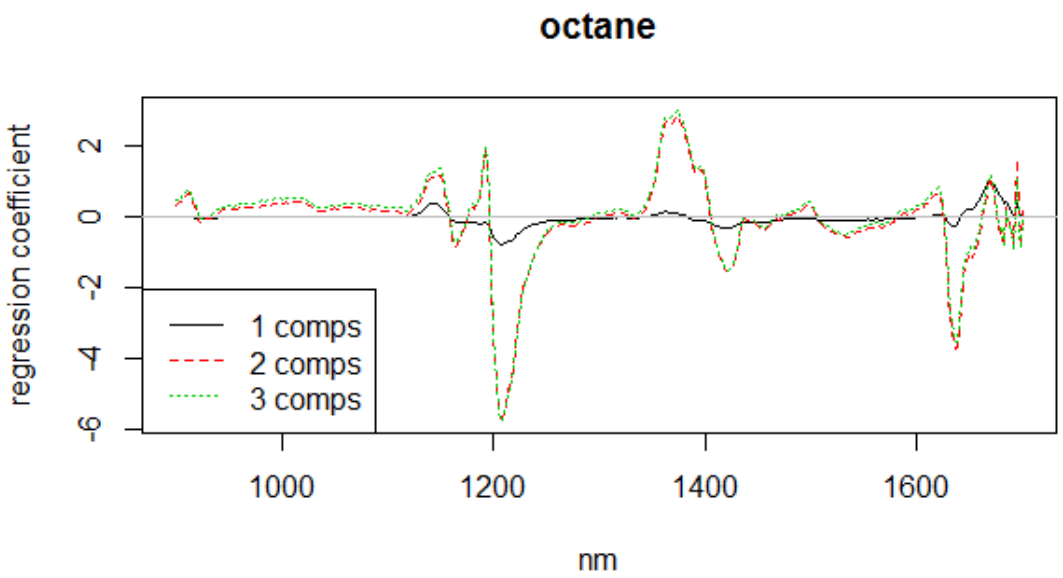

Figura 4: Coeficientes de regresión para los datos de la gasolina

Obsérvese que los coeficientes para dos componentes y tres componentes son similares, es porque la tercera componente influye muy poco en las predicciones. Los RMSEP y las predicciones para dos y tres componentes son bastante similares como habíamos comprobado anteriormente.

Las puntuaciones (score) y las cargas (loadings) se pueden representar mediante las funciones **scoreplot** y **loadplot.** Se puede indicar el número de componentes con el argumento **comps**, si se dan más de dos componentes el gráfico de puntuaciones será un diagrama de pares, de lo contrario será un diagrama de dispersión. Para **loadplot**, el valor predeterminado es utilizar gráficos lineales.

Finalmente, un diagrama de cargas de correlación "correlation loadings" (función **corrplo**t o plottype="correlation") muestra las correlaciones entre cada variable y las componentes seleccionadas. Son gráficos de dispersión de dos series de puntuaciones con círculos concéntricos de radios dados **radii**. Cada punto corresponde a una variable X, el cuadrado de la distancia entre el punto y el origen es igual a la fracción de la varianza de la variable explicada por las componentes en el panel. Los valores predeterminados para radii corresponden a 50% y 100% que explican la varianza.

La función plot acepta la mayoría de los parámetros de los gráficos más comunes, tales como col y pch. Si el modelo tiene varias respuestas o seleccionamos más de un tamaño de modelo, por ejemplo, ncomp=4:6, en algunas funciones del gráfico (en particular en gráficos de predicción, gráficos de validación y gráficos de coeficientes) el gráfico se divide en ventanas y un gráfico se muestra para cada combinación de respuesta y el tamaño del modelo. El número de filas y columnas son elegidos de forma automática, pero puede especificarse explícitamente con argumentos **nrows** y **ncols**. Si hay varios gráficos que se ajustan a la pantalla de diagramas, presionando retorno vemos el resto de gráficos.

6.2-Extracción.

Los coeficientes de regresión se pueden extraer utilizando la función genérica coef; la función toma varios argumentos, indicando el número de componentes a tener en cuenta, y si se necesita la intersección por defecto es FALSE.

Las puntuaciones y las cargan pueden ser extraídas utilizando funciones score y loadings de X, y los datos Yscores e Yloadings de Y, estos también devuelven el porcentaje de varianza explicada como atributos en plsr, los pesos pueden ser extraídos utilizando la función loading.weights. Cuando se aplica a un modelo pcr la función devuelve NULL.

Nota: los comandos con argumento plot(score(gas1)) son correctos y muestran el mismo gráfico que si usamos el argumento scoreplot.

6.3-Resumen de datos.

El método de impresión para un objeto de la clase “mvr” muestra el tipo de regresión utilizado, quizás indica la forma de validación empleada y muestra la función a la que hacemos referencia. La función **summary** nos muestra la cantidad de varianza explicada por el método utilizado (el número de variables latentes para todas las opciones). Summary tiene un argumento adicional **what** para poder obtener los resultados de la fase entrenamiento o la validación, respectivamente. Por defecto muestra ambos datos.

7.-Predicción de nuevas observaciones.

El ajuste de modelos a menudo se utiliza para predecir futuras observaciones, y pls tiene implementado **predict** para modelos plsr y pcr. La forma más común de llamar a esta función es predict (mymod, ncomp=myncomp, newdata=mynewdata) donde mymod es

un modelo ajustado, myncomp especifica el tamaño del modelo y mynewdata es un marco de datos con las nuevas observaciones de X. El marco de datos también puede contener

mediciones de respuesta para las nuevas observaciones, que puede ser utilizado para comparar los valores predichos a los medidos, o para estimar la capacidad de predicción global del modelo. Si la función no encuentra newdata, predict utiliza los datos utilizados para encajar el modelo, es decir, devuelve valores ajustados.

Si la función no reconoce ncomp, predict devuelve predicciones para los modelos con 1 componente, 2 componentes, ⋯, a componentes. donde a es el número de componentes utilizados para ajustar el modelo. De lo contrario, se utiliza el tamaño del modelo indicado en ncomp. Por ejemplo, para obtener las predicciones del modelo construido anteriormente con dos y tres componentes usaríamos:

```
predict(gas1, ncomp = 2:3, newdata = gastest[1:5,])
```

```
, , 2 comps

     octane
51 87.94125
52 87.25242
53 88.15832
54 84.96913
55 85.15396

, , 3 comps

     octane
51 87.94907
52 87.30484
53 88.21420
54 84.86945
55 85.24244
```

(Se predice sólo las cinco primeras observaciones de prueba). Las predicciones con dos y tres componentes son bastantes similares. Podría ser de esperar, dado que los vectores de regresión, así como los RMSEP estimados para los dos tamaños de modelo fueron similares.

También se puede especificar de forma explícita que las componentes a utilizar en la predicción, se realiza especificando las componentes en las composiciones de argumento. (si se especifican ncomp y composiciones comps, tiene prioridad comps sobre ncomp). Por ejemplo, para obtener predicciones de un modelo con solo dos componentes, se puede utilizar:

```
predict(gas1, comps = 2, newdata = gastest[1:5,])
```

```
     octane
51 87.53322
52 86.30322
```

```
53 87.35217
54 85.81561
55 85.31952
```

Los resultados son diferentes de las predicciones con dos componentes (es decir, los componentes de una y dos anterior) (el intercept se incluye siempre en las predicciones, se puede eliminar sustrayendo mymod$Ymeans de los valores pronosticados).

La función predict devuelve una matriz tridimensional, en la que la entrada (i, j, k) es el valor predicho para i observación, la respuesta de j y el tamaño del modelo k. Hay que tener en cuenta que la predicción de cinco observaciones para un modelo uni-respuesta con ncomp=3 da una matriz 5 x 1 x 1, no es un vector de longitud cinco. Esto es para que sea más fácil para distinguir entre las predicciones de los modelos con una de la respuesta y predicciones con el tamaño del modelo. Se puede usar las dimensiones explícitamente mediante el uso de **drop(predict(..)).**

```
drop(predict(gas1, ncomp = 2:3, newdata = gastest[1:5,]))
```

```
    2 comps  3 comps
51 87.94125 87.94907
52 87.25242 87.30484
53 88.15832 88.21420
54 84.96913 84.86945
55 85.15396 85.24244
```

Los valores perdidos en newdata se modifican en NA en las predicciones previstas, de manera predeterminada, se puede modificar con el argumento na.action.

El newdata no tiene que ser un marco de datos. Reconociendo el hecho de que the right hand side de PLS y PCR muy a menudo son un solo término matriz, la función predict permite utilizar una matriz como newdata por lo que en lugar de

newdataframe <- data.frame (X = newmatrix)

predict(…, newdata = newdataframe)

se puede decir directamente:

predict(…, newdata = newmatriz)

Sin embargo, hay un par de advertencias: en primer lugar, solo se puede aplicar en la función predict, otras funciones que tome un argumento newdata (como RMSEP) debe tener un marco de datos, ya que también necesitará los valores respuesta. En segundo lugar, cuando newdata es un marco de datos, predict es capaz de realizar más pruebas en los datos suministrados, tales como las dimensiones y tipos de variables. Por último, con la excepción del posicionamiento (especificado con el argumento scale al ajustar el modelo), cualquier transformación o codificación de factores e interacciones tienen que ser realizadas manualmente si newdata es una matriz.

A menudo es interesante predecir las puntuaciones de nuevas observaciones, en lugar de los valores de respuesta, se puede realizar mediante la especificación en el argumento

tipo=”scores” en la función predict. Se obtiene de esta forma una matriz con las puntuaciones correspondientes a las componentes especificadas en comps (ncomp es aceptado como sinónimo de comps a la hora de predecir las puntuaciones).

Existe un gráfico de predicciones con la función **predplot;** esta función es genérica y también se puede utilizar para el gráfico de predicciones a partir de otros tipos de modelos, como lm. Dicha función se usa con el siguiente argumento:

```
predplot(gas1, ncomp = 2, newdata = gastest, asp=1, line=TRUE)
```

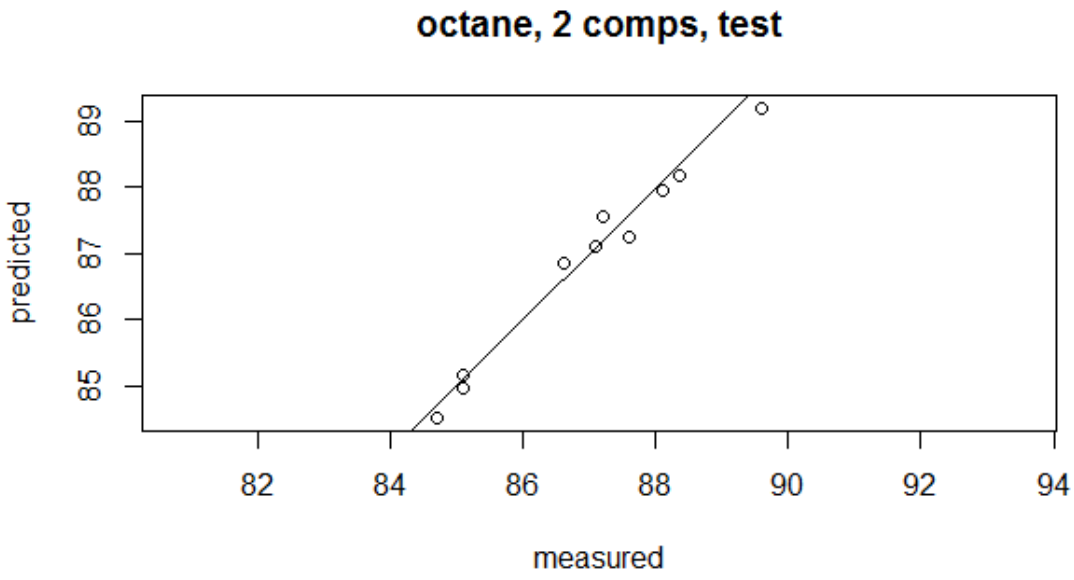

Figura 5: Test de predicciones

Este gráfico predice, con dos componentes frente a valores de respuesta medios. (Hay que tener en cuenta que newdata debe ser un marco de datos con las dos variables X e Y).

8.-Otros.

En este apartado se detallan un par de temas pocos técnicos para el uso más avanzado del paquete.

Existen varios algoritmos PLSR, y el paquete pls actualmente implementa tres de ellos:

el algoritmo de núcleo para matrices, the kernel algorithm, (muchas observaciones, pocas variables), el clásico algoritmo de puntuaciones, también conocido como NIPALS y el algoritmo SIMPLS. El algoritmo Kernel y el algoritmo de puntuaciones producen los mismos resultados, aunque el algoritmo Kernel suele ser más rápido para la mayoría de los problemas. El algoritmo SIMPLS produce el mismo ajusto cuando se trata de una sola respuesta; pero existe una ligera diferencia para los modelos de respuesta múltiple. También es generalmente más rápido que el algoritmo NIPALS.

El valor por defecto es utilizar el algoritmo de Kernel. Se puede especificar el algoritmo que queramos utilizar con el argumento: method =”oscorespls”.

Cada vez que se desee cambiar el algoritmo, usar la opción de anotarlo puede resultar un poco tedioso por lo que existe la opción de cambiar la configuración que existe por defecto con la función pls.options; devuelve la configuración actual como una lista denominada:

```
> pls.options()

$mvralg
[1] "kernelpls"

$plsralg
[1] "kernelpls"

$cpplsalg
[1] "cppls"

$pcralg
[1] "svdpc"

$parallel
NULL

$w.tol
[1] 2.220446e-16

$X.tol
[1] 1e-12
```

Las opciones especifican el algoritmo de ajuste por defecto del mvr, plsr y pcr. Para que devuelve sólo una opción específica se utiliza la función pls.options(“plsralg”). Para cambiar el algoritmo predeterminado para plsr el resto de la sesión, se puede utilizar:

```
pls.options(plsralg="oscorespls")
```

Hay que tener en cuenta que este cambio de opción solo dura mientras dure la sesión de R.

8.1-Validación cruzada paralela.

La validación cruzada es un procedimiento exigente computacionalmente. Las funciones de ajuste subyacentes se han optimizado, y la implementación de validación cruzada que se usa cuando se especifica el argumento de validación para mvr trata de evitar cualquier cálculo que no sean necesarios; aun así, la validación puede llevar mucho tiempo, en los modelos con grandes matrices, muchos componentes o varios segmentes.

Desde la versión 2.14.0, R ha trabajado con **package parallel** para ejecutar cálculos paralelamente, en las máquinas multi-CPU o en varias máquinas. El paquete pls puede utilizar sistemas de ese tipo para ejecutar las validaciones cruzadas en paralelo.

8.2-Paquete de diseño (package design).

El paquete pls está diseñado de tal manera que una función mvr utiliza la fórmula y los datos, y llama a una función subyacente de ajuste (y posiblemente una función de validación cruzada) para hacer el trabajo real.

Los paquetes de gráficos están implementados de forma similar. Las funciones son independientes del gráfico destinado a estar interactivo, ya que hay personas que les gusta usar la función de gráfico genérica; mientras que a otros les gusta usar funciones separadas para cada tipo de gráfico. También existen gráficos para algunos de los componentes de los modelos equipados que se pueden obtener con funciones de extracción, como puntuación y matrices de carga.

8.3-Utilizar funciones de ajuste directamente.

Las funciones subyacentes de ajustes se denominan kernelpls.fit, oscorespls.fit y simpls.fit para los modelos plsr, y svdpc.fit para el modelo pc. Todas ellas toman argumentos X, Y, ncomp y stripped. Denominamos a los argumentos X, Y cómo matrices, no marco de datos y ncomp cómo el número de componentes. El valor por defecto del argumento stripped es FALSO. Cuando es TRUE, los cálculos son el mínimo requerido para la media de la X y la media de la Y y los coeficientes de regresión. Se utiliza para acelerar los procedimientos de validación cruzada.

Las funciones de ajuste pueden ser llamadas directamente, por ejemplos, cuando se necesita evitar la sobrecarga de las fórmulas y utilización de datos en repetidos ajustes.

8.4-Utilización de fórmulas con más detalle.

El manejo de las fórmulas y variables en el modelo de ajuste es muy similar a lo que ocurre en la función lm: las variables especificadas en la fórmula se buscan en la trama de datos dada en el argumento de datos de la función de ajuste (plsr, pcr o mvr), o en el entorno de la llamada si no se encuentra en la trama de datos. Los factores se codifican en una o más de las columnas, dependiendo del número de niveles, y en la opción de contrastes. Todas las variables son recogidas en una matriz numérica del modelo. Esta matriz es entonces entregada al ajuste subyacente o funciones de validación cruzada. Una manipulación similar se utiliza en el método de predecir.

El intercept es tratado de forma especial en los pls. Después de la matriz del modelo se haya construido, se elimina la columna intercept. Esto asegura que cualquier factor se codifique como si el intercept estuviese presente. Las funciones de ajuste que subyacen a continuación, centran el resto de las variables como parte del proceso de adaptación (esto es intrínseco en los algoritmos plsr y pcr). El intercept se usa por separado; una consecuencia de eso es que en las fórmulas se especifican explícitamente sin el intercept (por ejemplo y~a+b-1), solo dará lugar a la codificación de cualquier factor para cambiar; el intercept seguirá siendo ajustado.

CAPITULO 7: CASO PRACTICO DE REGRESION PLS EN R.

1.-Introducción.

Como se ha indicado anteriormente en la librería específica que tiene R para el algoritmo pls, existen varios ejemplos, en este caso en concreto se va analizar los datos Gasolina y con ellos se realizan un análisis del algoritmo desarrollado anteriormente utilizando los comandos y argumentos especificados en el apartado anterior.

gasoline Un conjunto de datos consistentes en el índice de octano (**octane**) y el espectro NIR (NIR) de 60 muestras de gasolina. Cada espectro NIR consta de 401 mediciones de reflectancia de 900 a 1700nm.

Se realiza una carga del paquete para poder utilizarlo con el siguiente argumento:

```
library(pls)
```

Posteriormente también se cargan los datos con el siguiente comando:

```
data("gasoline")
```

A partir de ahora se supone que el paquete y los conjuntos de datos han sido cargados como como se ha indicado. En este apartado, se realiza una PLSR con los datos de gasolina para ilustrar el uso de PLS.

En primer lugar, se divide el conjunto de datos en datos entrenamiento y conjunto de datos prueba con los siguientes comandos; en el desarrollo anterior se ha dividido la muestra simplemente nombrando los 50 primeros datos y los 50 siguientes, en este caso, para que el ejemplo sea más práctico, se realiza con una semilla, de forma aleatoria y usando para los datos entrenamiento el 75% de los datos y el resto 25% los datos test, se usan los siguientes comandos para el desarrollo indicado:

```
set.seed(1)
n=nrow(gasoline)
train=sample(1:n,  floor(n*0.75))
test=setdiff(1:n,train)
```

Con ello se obtienen los datos preparados para su posterior estudio.

2.-Estudio de componentes principales PCR.

A continuación, se realiza un análisis de componentes principales para posteriormente realizar una comparativa del estudio, para ello usamos los siguientes argumentos:

```
pcr.fit=pcr(octane~., data=gasoline,subset=train,
            scale=TRUE, validation="CV")
```

Con esta función se ajustan los datos a un modelo de componentes principales. Con el argumento "`data=gasoline`" se indican los datos que se desean ajustar, en concreto con el argumento "`subset=train`" solo se usan un conjunto de esos datos en concreto el denominado train en la anterior división, con "`scale=TRUE`" se indica que los datos sean tipificados y por último se desea usar validación cruzada y para ello se usa el siguiente argumento `validation="CV".` Se obtiene un modelo ajustado y denominado "`pcr.fit`".

A continuación, para el análisis del modelo, lo mejor es obtener el gráfico de las predicciones de los cuadrados medios, Mean Squared of Prediction ("MSEP") y de esta forma se analiza el modelo.

```
validationplot(pcr.fit,val.type="MSEP")
```

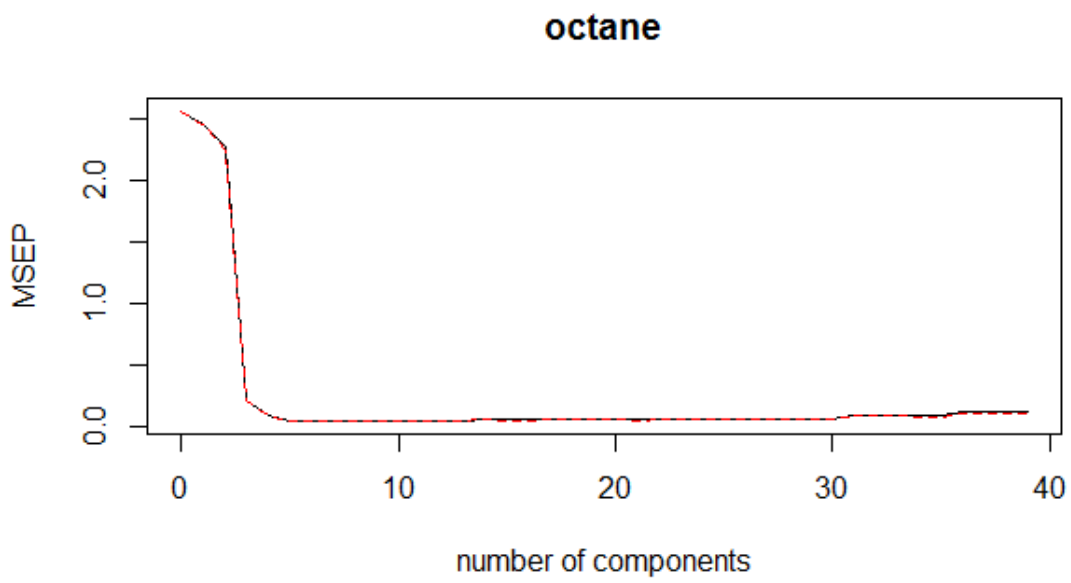

Figura1: Curva MSEP con validación cruzada para los datos de gasolina

Este gráfico muestra los MSEP como funciones del número de componentes.

Con este gráfico se puede analizar que dos componentes parecen ser suficiente ya que, una vez elegido el número de componentes, se pueden sacar conclusiones de los diferentes aspectos del ajuste mediante el gráfico de residuos.

```
plot(pcr.fit, plottype = "scores", comps = 1:3)
```

En este caso, se indica el tipo de gráfico que se desea analizar “scores” (gráfico de residuos y se indica el número de componentes a analizar 1:3 y obteniendose el siguiente gráfico:

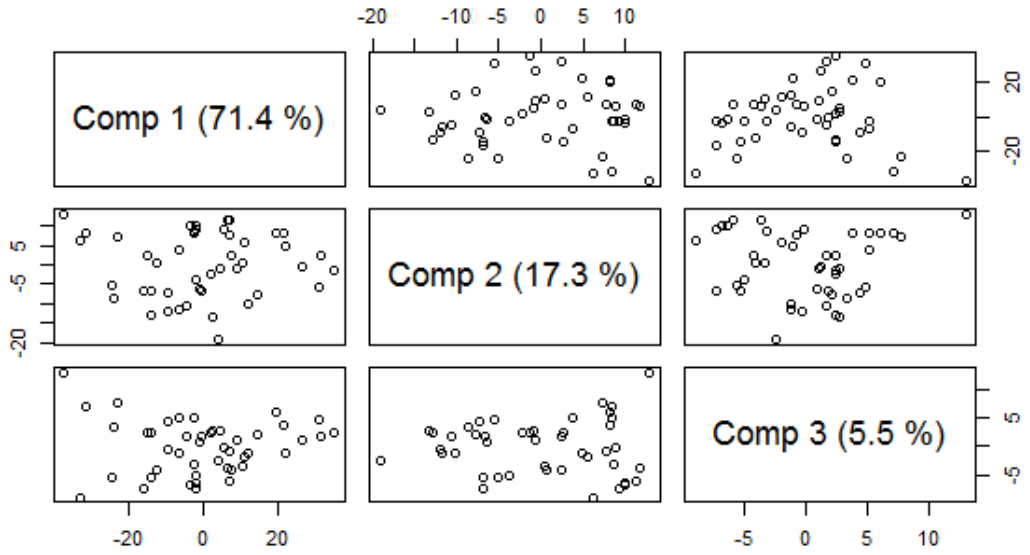

Figura 2: Gráfico de residuos para los datos de gasolina.

Se obtiene de esta forma un gráfico de residuos para las tres primeras componentes. Los gráficos de residuos se usan a menudo para buscar patrones o valores atípicos en los datos. En este ejemplo no existen indicios de valores atípicos. Los datos que aparecen entre paréntesis después de cada componente corresponden al porcentaje de varianza explicada por cada componente para la variable X. También se puede obtener explícitamente la varianza explicada con el siguiente comando:

```
explvar(pcr.fit)
cumsum(explvar(pcr.fit))
```

Y con esta función se realiza una suma acumulada, obteniendo:

```
  Comp 1   Comp 2   Comp 3   Comp 4   Comp 5   Comp 6   Comp 7   Comp 8   Comp 9   Comp 10
71.41713 88.71433 94.25020 97.74322 98.57829 98.98205 99.20757 99.38855 99.50176  99.60099

Comp 11  Comp 12  Comp 13  Comp 14  Comp 15  Comp 16  Comp 17  Comp 18  Comp 19  Comp 20
99.66653 99.72865 99.77014 99.80685 99.83683 99.85971 99.87573 99.89081 99.90375  99.91560

Comp 21  Comp 22  Comp 23  Comp 24  Comp 25  Comp 26  Comp 27  Comp 28  Comp 29  Comp 30
99.92594 99.93515 99.94354 99.95114 99.95741 99.96244 99.96736 99.97119 99.97453  99.97777

Comp 31  Comp 32  Comp 33  Comp 34  Comp 35  Comp 36  Comp 37  Comp 38  Comp 39
99.98051 99.98317 99.98535 99.98740 99.98919 99.99091 99.99237 99.99368 99.99492
```

```
plot(pcr.fit, "loadings", comps = 1:2, legendpos = "bottomleft",
+      labels = "numbers", xlab = "nm")
 abline(h = 0)
 grid()
 colnames(gasoline$NIR)
```

Con estos argumentos se realiza el gráfico de las cargas indicando que añada leyendas y cuadrículas para el gráfico y de esta forma poder realizar un buen análisis del mismo:

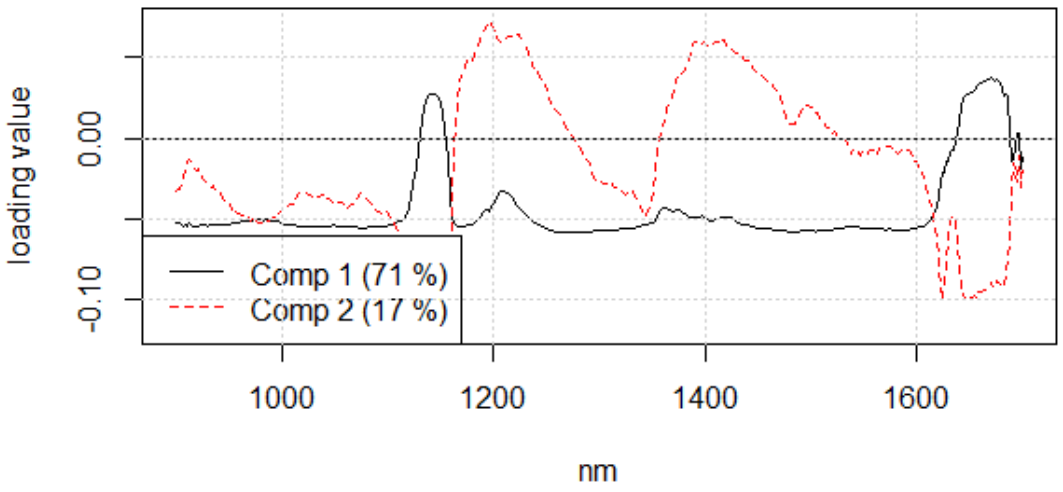

Figura 3: Gráfico de cargas para los datos de gasolina.

A continuación, se realiza análisis de la estimación del cuadrado medio del modelo y se indica que muestre los datos con los siguientes argumentos:

```
ECM_PCR=MSEP(pcr.fit, estimate = "all")
str(ECM_PCR)
CVvalid=ECM_PCR$val[,1,]
CVvalid
```

Obteniéndose:

```
 model estimate
          (Intercept)   1 comps     2 comps     3 comps     4 comps      5 comps      6 comps
   train     2.453143  2.273629    1.902154   0.1740344  0.06916176   0.03977606   0.03460767
   CV        2.565917  2.464097    2.271297   0.2105787  0.08914745   0.05372683   0.05239669
   adjCV     2.565917  2.453602    2.252352   0.2143395  0.08769641   0.05274292   0.05139936

 model estimate
                          7 comps    8 comps     9 comps     10 comps     11 comps      12 comps
   train                0.03376480 0.02535815 0.02515796 0.02514342 0.02466519 0.02435999
   CV                   0.05308565 0.05061062 0.05084773 0.05392278 0.05196525 0.05150284
   adjCV                0.05353259 0.04867819 0.04886039 0.05183565 0.04991843 0.04983876

 model estimate
                          13 comps   14 comps    15 comps    16 comps     17 comps     18 comps
   train                0.02366687 0.02323907 0.01795919 0.01795905 0.01791816 0.01760391
   CV                   0.05241551 0.06105575 0.05695607 0.05782120 0.05950772 0.06166315
   adjCV                0.05064410 0.05974526 0.05363948 0.05480636 0.05667998 0.05906188


model estimate
                          19 comps   20 comps    21 comps    22 comps     23 comps     24 comps
   train                0.01687167 0.01477999 0.01399559 0.01324132 0.01293282 0.01292744
   CV                   0.05858866 0.05952941 0.05761943 0.06045486 0.06143858 0.06325470
   adjCV                0.05673255 0.05652990 0.05454096 0.05680784 0.05760577 0.05951753

model estimate
                          25 comps   26 comps    27 comps    28 comps     29 comps     30 comps
   train                0.01258128 0.01247073 0.01193994 0.01180810 0.01180662 0.01069634
   CV                   0.06737609 0.06930224 0.06693672 0.06307537 0.07038263 0.06809971
   adjCV                0.06320130 0.06511213 0.06276463 0.05976571 0.06713168 0.06468706

model estimate
                          31 comps   32 comps    33 comps     34 comps    35 comps     36 comps
   train                0.01065524 0.01054248 0.006303507 0.006168338 0.005130689 0.004936267
   CV                   0.09417571 0.09179947 0.095533149 0.087129941 0.087724972 0.120293001
   adjCV                0.09036911 0.08905808 0.088542857 0.081420417 0.081176703 0.111453410

model estimate
                          37 comps    38 comps    39 comps
   train                0.004737025 0.004624137 0.003963143
   CV                   0.123141823 0.117920030 0.123483384
   adjCV                0.114437174 0.110471912 0.115265466
```

Para un mejor análisis de estos datos obtenidos, se realiza un gráfico con ellos esta vez usando otro argumento para gráficos

```
matplot(t(CVvalid),type="l",ylab="ECM_VC")
```

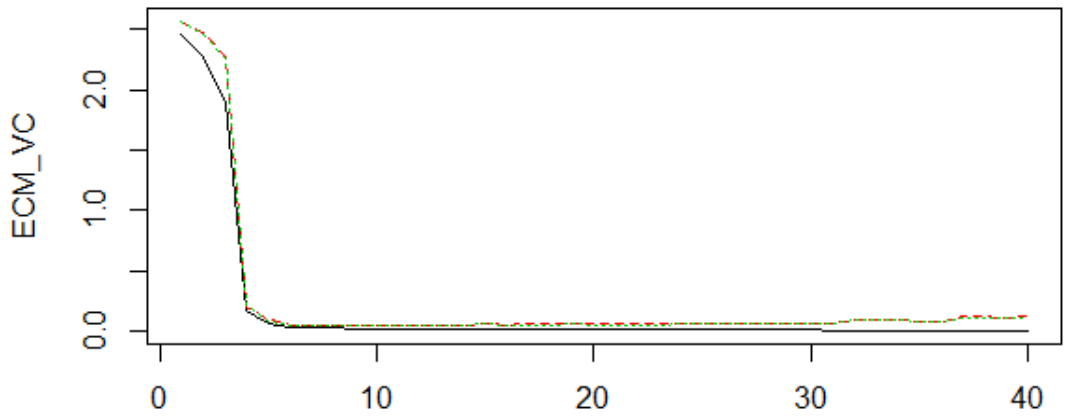

Figura4: Curva ECM con validación cruzada para los datos de gasolina

Se puede calcular el mínimo con el siguiente argumento:

```
which.min(CVvalid[2,])
```

```
8 comps
      9
```

A continuación, se realizan las predicciones y para ello se usa la submuestra test, calculando el error cuadrático medio e indicando el número de componente 8 como se ha calculado anteriormente y para el análisis del mismo se realiza un gráfico con los resultados con sus cuadrículas para el mejor análisis de los mismo:

```
pcr.pred=predict(pcr.fit,gasoline[test,],ncomp=8)
ECM_test_PCR=mean((pcr.pred-gasoline$octane[test])^2)
R2_test_PCR=cor(pcr.pred,gasoline$octane[test])^2
plot(gasoline$octane[test],pcr.pred,xlab="octanaje",ylab="Predic_octan
")
abline(a=0,b=1,col="blue")
grid()
```

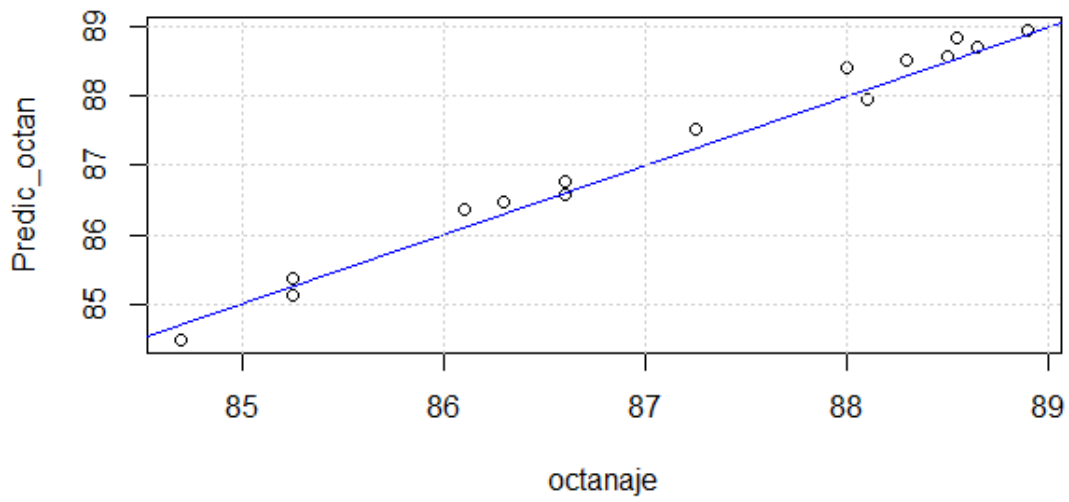

Figura 5: Test de predicciones

3.-Estudio de los datos con regresión PLS.

De igual forma que lo se ha descrito anteriormente se usan los comandos correspondientes para el análisis PLS usando los comandos:

```
pls. Fit=plsr(octane~., data=gasoline,subset=train,
              scale=TRUE, validation="CV")
```

Con esta función ajustamos los datos a un modelo PLS. Con el argumento "`data=gasoline`" indicamos los datos que queremos ajustar, en concreto con el argumento "`subset=train`" solo queremos usar un conjunto de esos datos en concreto el denominado train en la anterior división como hemos realizado anteriormente usamos los datos entrenamiento, con "`scale=TRUE`" indicamos que los datos sean tipificados y por último queremos usar validación cruzada y ello lo indicamos con el siguiente argumento `validation="CV".` Tenemos ajustado el modelo y denominado "`pls.fit`".

Como se ha realizado anteriormente se usa el gráfico “MSEP” para un mejor análisis del modelo ajustado y realizar una evaluación del número de componentes

```
validationplot(pls.fit,val.type="MSEP")
```

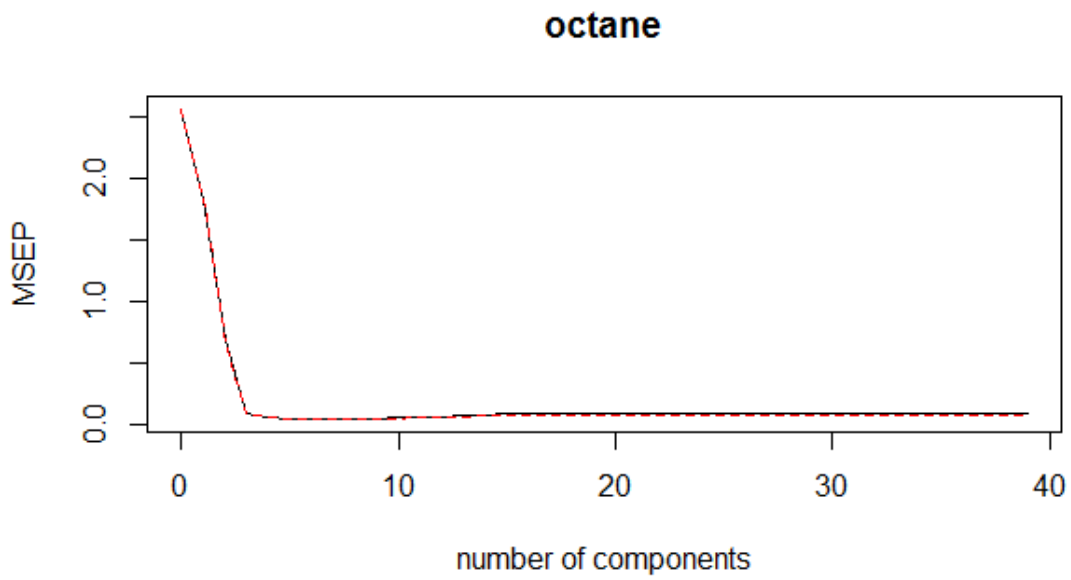

Figura 6: Curva MSEP con validación cruzada para los datos de gasolina

Este gráfico nos muestra los MSEP como funciones del número de componentes en este caso usando un modelo PLS. Se puede analizar con el gráfico que dos componentes parecen ser suficiente y es un gráfico muy similar al obtenido en el caso de PCR, se va a analizar los residuos con un gráfico y comprobar la situación.

```
plot(pls.fit, plottype = "scores", comps = 1:3)
```

Indicar el tipo de gráfico y usando el mismo número de componentes que en modelo anterior para analizar la diferencia.

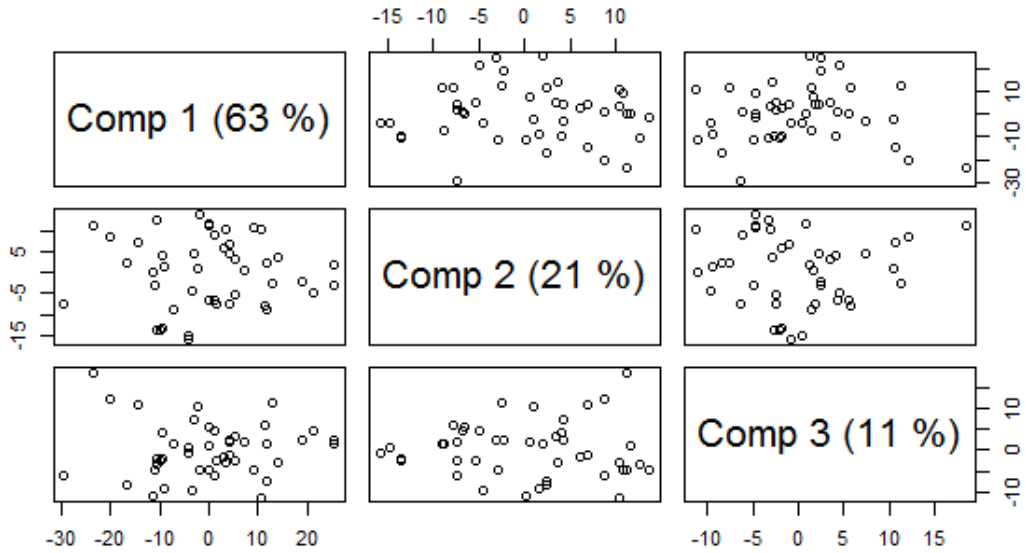

Figura 7: Gráfico de residuos para los datos de gasolina.

Se obtiene un gráfico de residuos para las tres primeras componentes y como se ha explicado antes también se pueden analizar patrones o valores atípicos. Existen diferencias entre el porcentaje de varianza explicada por cada componente por lo que se procede al análisis más explícito de la varianza y para ello se usan los siguientes comandos:

```
explvar(pls.fit)
cumsum(explvar(pls.fit))
```

Y con esta función se realiza una suma acumulada, para una mejor visión del análisis obteniendo los siguientes resultados:

```
 Comp 1   Comp 2   Comp 3   Comp 4   Comp 5   Comp 6   Comp 7   Comp 8   Comp 9
62.54337 83.65065 94.16813 97.40623 98.52853 98.85734 99.13593 99.21152 99.38746

 Comp 10  Comp 11  Comp 12  Comp 13  Comp 14  Comp 15  Comp 16  Comp 17  Comp 18
99.47669 99.54584 99.60376 99.63859 99.70249 99.75113 99.79431 99.82256 99.85014

 Comp 19  Comp 20  Comp 21  Comp 22  Comp 23  Comp 24  Comp 25  Comp 26  Comp 27
99.86682 99.87571 99.88381 99.89904 99.91139 99.92119 99.93032 99.93831 99.94572

 Comp 28  Comp 29  Comp 30  Comp 31  Comp 32  Comp 33  Comp 34  Comp 35  Comp 36
99.95289 99.95804 99.96371 99.96690 99.97109 99.97616 99.97942 99.98314 99.98526

 Comp 37  Comp 38  Comp 39
99.98704 99.98934 99.99180
```

Se puede verificar esa diferencia que se ha detectado en el gráfico entre la primera y segunda componente; a partir de la tercer la diferencia es mínima.

Con el gráfico de cargas también se detectan diferencia con respecto al anterior modelo:

```
loadingplot(pls.fit, comps = 1:2, legendpos = "bottomleft",
          labels = "numbers", xlab = "nm")
abline(h = 0)
grid()
colnames(gasoline$NIR)
```

Obteniendo el siguiente gráfico:

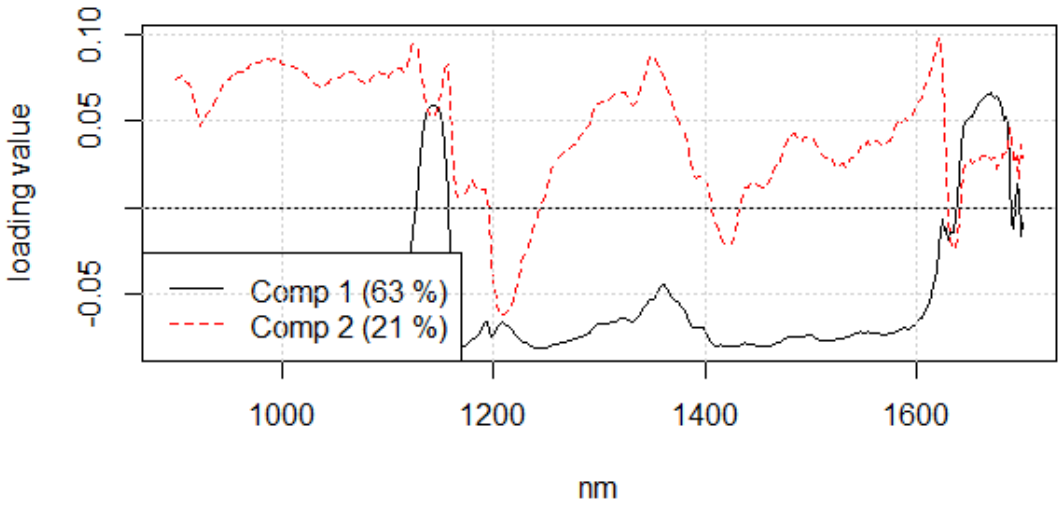

Figura 8: Gráfico de cargas para los datos de gasolina.

En este gráfico se puede verificar una gran diferencia en las cargas con el modelo PLS

A continuación, se realiza análisis de la estimación del cuadrado medio del modelo y se indica que muestre los datos con los siguientes argumentos:

```
ECM_PLS=MSEP(pls.fit, estimate = "all")
str(ECM_PLS)
CVvalid=ECM_PLS$val[,1,]
CVvalid
```

Obteniéndose los siguientes resultados:

```
model estimate
          (Intercept)    1 comps    2 comps     3 comps     4 comps     5 comps     6 comps
   train    2.453143    1.663984   0.4963545   0.06427954  0.04460421  0.02908233  0.02218196
   CV       2.565917    1.840476   0.6813242   0.08929267  0.06080300  0.04877431  0.04713937
   adjCV    2.565917    1.861459   0.6728186   0.08633782  0.06148731  0.04791427  0.04566517


model estimate
                        7 comps    8 comps     9 comps     10 comps     11 comps    12 comps
   train               0.01890722  0.01279882  0.01142196  0.00915487  0.00691527  0.00528252
   CV                  0.04701504  0.04976482  0.05161870  0.05261358  0.05881769  0.06445205
   adjCV               0.04547240  0.04669696  0.04881265  0.04921196  0.05461107  0.05946520

model estimate
                        13 comps    14 comps    15 comps    16 comps    17 comps     18 comps
   train               0.00319901  0.00199587  0.00124958  0.00094316  0.00057484  0.00040212
   CV                  0.07208761  0.08114620  0.08440270  0.08916552  0.08854300  0.08737416
   adjCV               0.06562944  0.07358398  0.07623670  0.08045257  0.07968150  0.07860331

model estimate
                      19 comps    20 comps      21 comps      22 comps     23 comps   24 comps
   train             0.0002125874 9.110379e-05 3.397931e-05 2.041129e-05 1.085343e-05 5.535279e-06
   CV                0.0888591758 8.888574e-02 8.978188e-02 9.023793e-02 9.017042e-02 9.038432e-02
   adjCV             0.0798184220 7.977788e-02 8.054459e-02 8.095410e-02 8.088973e-02 8.107921e-02

model estimate
                      25 comps     26 comps     27 comps      28 comps     29 comps   30 comps
   train             2.779388e-06 1.025251e-06 3.013997e-07 1.075688e-07 3.917004e-08 1.072957e-08
   CV                9.034939e-02 9.038283e-02 9.043748e-02 9.042729e-02 9.043922e-02 9.044385e-02
   adjCV             8.104630e-02 8.107550e-02 8.112428e-02 8.111511e-02 8.112599e-02 8.113013e-02

model estimate
                      31 comps     32 comps     33 comps     34 comps     35 comps     36 comps
   train             2.920409e-09 7.473883e-10 8.721328e-11 1.404451e-11 2.815117e-12 5.600475e-14
   CV                9.044728e-02 9.044620e-02 9.044555e-02 9.044573e-02 9.044567e-02 9.044567e-02
   adjCV             8.113319e-02 8.113223e-02 8.113164e-02 8.113180e-02 8.113175e-02 8.113175e-02

model estimate
                      37 comps     38 comps    39 comps
   train             3.260838e-15 2.281324e-16  1.265666e-17
   CV                9.044567e-02 9.044567e-02  9.044567e-02
   adjCV             8.113175e-02 8.113175e-02  8.113175e-02
```

En los resultados de la validación existen dos estimaciones de validación cruzada: CV es la estimación CV común y adjCV es un sesgo corregido de la estimación CV.

A menudo es más fácil analizarlos por gráficos, usamos para ello el siguiente comando:

```
matplot(t(CVvalid),type="l",ylab="ECM_VC")
```

Obteniendo el siguiente gráfico:

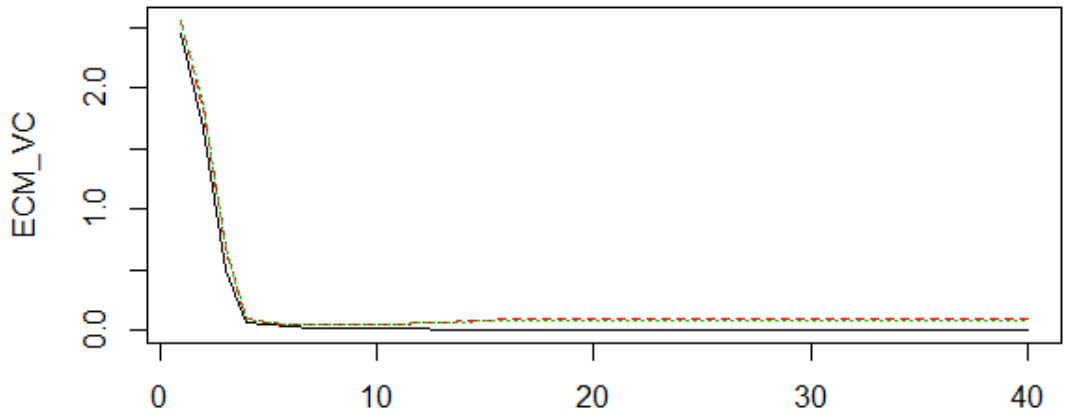

Figura 9: Curva ECM con validación cruzada para los datos de gasolina

Se puede calcular el mínimo con el siguiente argumento:

```
which.min(CVvalid[2,])
```

```
7 comps
      8
```

Comprobando de esta forma que PLS mejora el modelo anteriormente usado ya que se obtienen iguales resultado con menor número de componentes.

Se procede al cálculo de las predicciones y de igual forma se usa la submuestra test y en este caso se usa menor número de componentes como se ha calculado, para ello como se ha realizado hasta ahora se describe un gráfico con cuadrículas para un mejor análisis del mismo y se añade las predicciones obtenidas en el modelo PCR para una mejor comparativa

```
pls.pred=predict(pls.fit,gasoline[test,],ncomp=7)
ECM_test_PLS=mean((pls.pred-gasoline$octane[test])^2)
ECM_test_PLS=mean((pls.pred-gasoline$octane[test])^2) #ECM
R2_test_PLS=cor(pls.pred,gasoline$octane[test])^2
plot(gasoline$octane[test],pls.pred,xlab="octanaje",
      ylab="Predic_octan",col="red")
points(gasoline$octane[test],pcr.pred,col="blue")
abline(a=0,b=1,lty=2)
grid()
legend("topleft",col=c("blue","red"),pch=1,legend=c("PCR","PLS"))
```

Obtenemos el siguiente gráfico:

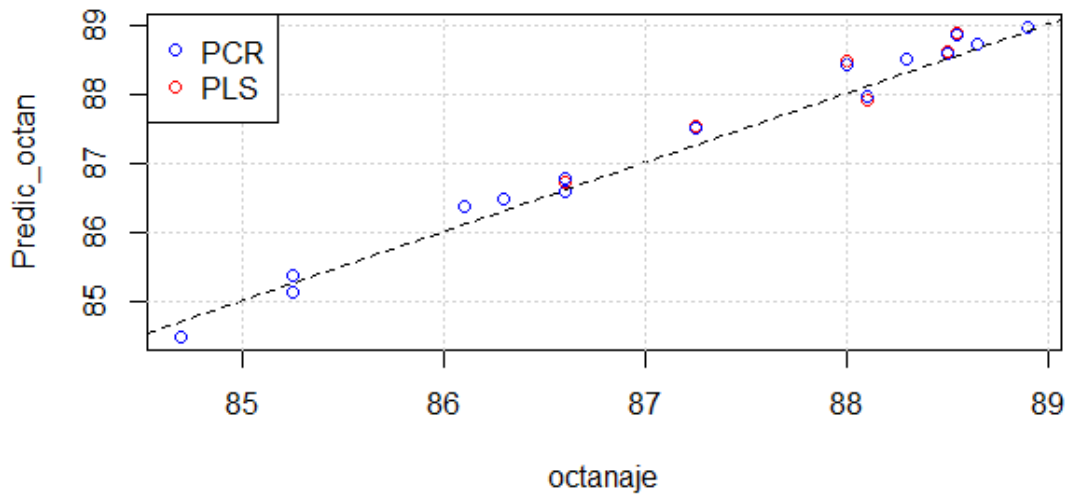

Figura 10: Test de predicciones.

En este gráfico se puede comprobar que se obtienen las mismas predicciones con el modelo PLS y sin embargo se necesitan menor número de componentes para obtener los mismos resultados por lo que el modelo estudiado mejora al anterior.

Por último, se muestran los datos de ambos para una mejora visión:

```
cbind(ECM_test_PCR,ECM_test_PLS)
```

```
     ECM_test_PCR       ECM_test_PLS
[1,]    0.0420405         0.04589339
```

```
cbind(R2_test_PCR,R2_test_PLS)
```

```
     R2_test_PCR        R2_test_PLS
[1,]   0.9869682          0.9857485
```

Con ello queda demostramos que el modelo funciona de forma muy similar usando menor número de componentes.

4.-Conclusiones sobre regresión PLS.

Como se ha estudiado en la parte teórica, el modelo de regresión PLS tiene ventajas sobre el modelo PCR. En la teoría en el estudio de componentes principales cuando X está altamente correlacionado con Y o bien no se seleccionan suficientes componentes, obtenemos malas predicciones; usando el modelo de regresión PLS tenemos la ventaja de solucionar el problema de la correlación entre las variables. Sin embargo, en la práctica apenas hay diferencias entre el uso del modelo de regresión PLS y el modelo PCR; en la mayoría de los casos, los métodos consiguen precisiones de predicción similares, aunque el modelo de regresión PLS por lo general necesita menos variables latentes que el modelo PCR, es decir, con el mismo número de variables latentes, el modelo de regresión PLS cubrirá más la variación en Y y el modelo PCR cubrirá más la variación en X.

Hay que tener en cuenta que en algunos casos el modelo de regresión PLS parece aumentar la varianza de los coeficientes de regresión individuales, por lo que no siempre es mejor que el modelo PCR.

ANEXOS. "Fichero de instrucciones del capítulo 7"

Se incluye Script completo que se ha ejecutado para el análisis de los modelos para el caso práctico:

```
############################################################
#REGRESION ACP Y PLS                          #          #
############################################################
#install.packages("pls")
library(pls)
data(gasoline)
#DIVIDIR LOS DATOS EN ENTRENAMIENTO/TEST
set.seed(1)
n=nrow(gasoline)
train=sample(1:n,  floor(n*0.75))
test=setdiff(1:n,train)

#PRIMERO REGRESION SOBRE C.P., PARA COMPARAR
pcr.fit=pcr(octane~., data=gasoline,subset=train,
            scale=TRUE, validation="CV")
validationplot(pcr.fit,val.type="MSEP")
plot(pcr.fit, plottype = "scores", comps = 1:3)
explvar(pcr.fit)
cumsum(explvar(pcr.fit))
plot(pcr.fit, "loadings", comps = 1:2, legendpos = "bottomleft",
    labels = "numbers", xlab = "nm")
abline(h = 0)
grid()
colnames(gasoline$NIR)

ECM_PCR=MSEP(pcr.fit, estimate = "all")
str(ECM_PCR)
CVvalid=ECM_PCR$val[,1,]
CVvalid
matplot(t(CVvalid),type="l",ylab="ECM_VC")
which.min(CVvalid[2,])
```

```
pcr.pred=predict(pcr.fit,gasoline[test,],ncomp=8)
ECM_test_PCR=mean((pcr.pred-gasoline$octane[test])^2) #ECM
R2_test_PCR=cor(pcr.pred,gasoline$octane[test])^2
plot(gasoline$octane[test],pcr.pred,xlab="octanaje",ylab="Predic_octan")
abline(a=0,b=1,col="blue")
grid()

#REGRESIÓN PLS
###############
pls.fit=plsr(octane~., data=gasoline,subset=train,
             scale=TRUE, validation="CV")
validationplot(pls.fit,val.type="MSEP")
plot(pls.fit, plottype = "scores", comps = 1:3)
explvar(pls.fit)
cumsum(explvar(pls.fit))
loadingplot(pls.fit, comps = 1:2, legendpos = "bottomleft",
       labels = "numbers", xlab = "nm")
abline(h = 0)
grid()
colnames(gasoline$NIR)

ECM_PLS=MSEP(pls.fit, estimate = "all")
str(ECM_PLS)
CVvalid=ECM_PLS$val[,1,]
CVvalid
matplot(t(CVvalid),type="l",ylab="ECM_VC")
which.min(CVvalid[2,])

pls.pred=predict(pls.fit,gasoline[test,],ncomp=7)
ECM_test_PLS=mean((pls.pred-gasoline$octane[test])^2) #ECM
R2_test_PLS=cor(pls.pred,gasoline$octane[test])^2
plot(gasoline$octane[test],pls.pred,xlab="octanaje",
    ylab="Predic_octan",col="red")
points(gasoline$octane[test],pcr.pred,col="blue")
abline(a=0,b=1,lty=2)
grid()
legend("topleft",col=c("blue","red"),pch=1,legend=c("PCR","PLS"))

cbind(ECM_test_PCR,ECM_test_PLS)
cbind(R2_test_PCR,R2_test_PLS)
```

BIBLIOGRAFIA.

- SEVERA-FRANCÉS, D. ARTEAGA-MORENO, F. IRENE-GIL. (2011). «Estimación de modelos causales con PLS: una aplicación al valor logístico». , Vol. 53, pp. 93-126.

- CEPEDA CARRION, G. ROLDÁN SALGUEIRO, JL. «Aplicado en la práctica la Técnica PLS en la Administración de Empresas».

- MARTIN, JUAN MIGUEL (2015) «Análisis de Componentes Principales» Universidad de Carlos III Madrid.

- CARLOS E. ALCIATURI, MARCOS E. ESCOBAR, CARLOS DE LA CRUZ, CARLOS RINCON (2003). «Partial Least Squares (PLS) regression and its application to coal analysis». Revista Técnica de la Facultad de Ingeniería Universidad del Zulia vol.26 n.3

- YURNIER E. TEJEDA RODRIGUEZ, VALIA GUERRA ONES, JESUS E. SANCHEZ GARCIA, RAMÓN CARRASCO VELAR (2012) «Utilización combinada de métodos exploratorios y confirmatorios para el análisis de la actividad antibacteriana de la cefalosporina (parte II)». *Revista Investigación operacional* vol.32 n.1 pp 114-120

- VALENCIA DELFA, JOSE LUIS, DIAZ-LLANOS Y SAINZ CALLEJA, JAVIER (2003) «Regresión PLS en las Ciencias Experimentales»

- HERVE ABDI, LYNNE J. WILLIAMS (2013) «Partial Least Squares Methods: Partial Least Squares Correlation and Partial Least Square Regression» chapter.23

- SVANTE WOLD, MICHAEL SJÖSTRÖM, LENNART ERIKSSON (2001) «PLS-regression: a basic tool of chemometrics». Chemometrics and intelligent laboratory systems. pp 109-130

- BJORN-HELGE MEVIK, RON WEHRENS (2007) «The pls Package: Principal Component Regression and Partial Least Squares Regression en R». Journal of Statistical Sofware. vol 18.

- BJORN-HELGE MEVIK, RON WEHRENS AND KRISTIAN HOVEDE LILAND (2016) «Partial Least Squares and Principal Component Regression».(*cran.r-project.org/web/packages/pls)*

- URL http://www.R-project.org/