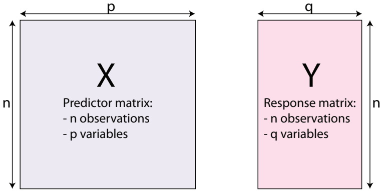
Introduction to PLS

B. Liquet

Partial Least Square (PLS)

PLS is a family of multivariate statistical techniques based on dimension reduction developed by S. Wold and H. Wold (1966, 1983). It can be seen as a supervised version of PCA.



X

Predictor matrix:
- n observations
- p variables

Y

Response matrix:
- n observations
- q variables

Modelling Aims

Partial Least Square is also a Dimension reduction method with the two following aims:

- Symmetric relationship: analyse the shared information.
- Asymmetric relationship: X = predictors, and Y = response.

Modelling Aims

When Y is only one column (univariate case), PLS can be summarized as

- Dimension reduction method: p dimension space $\Rightarrow$ K dimension space ($K << p$)
- PLS looks the best components the most correlated to the response variable
- The PLS components are linear combinations of the variables

$$C^k = u_1 \times SNP_1 + u_2 \times SNP_2 + \ldots + u_p \times SNP_p$$

- It is a supervised approach

Modelling Aims

In more general case (Y multivariate $q > 1$):

- PLS finds pairs of latent (score) vectors $\xi = \mathbf{Xu}$, $\omega = \mathbf{Yv}$

$$\xi = u_1 \times \mathrm{gene}_1 + u_2 \times \mathrm{gene}_2 + \cdots + u_p \times \mathrm{gene}_p$$
$$\omega = v_1 \times \mathrm{pheno}_1 + v_2 \times \mathrm{pheno}_2 + \cdots + v_p \times \mathrm{pheno}_q$$

- Symmetric relationship. Analyse the shared information.
- Asymmetric relationship. There is a set of response and predictor variables that can be used for prediction.

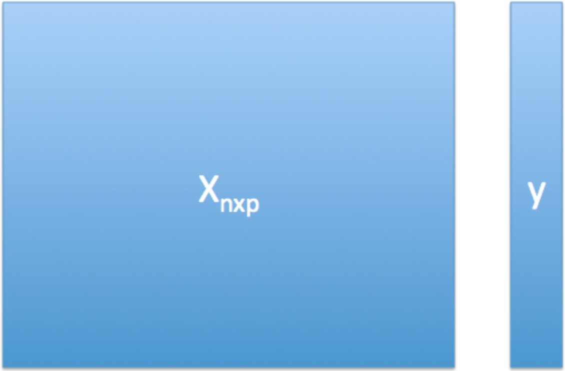
Objective function:

$$\max_{\|\boldsymbol{u}_h\|=1,\|\boldsymbol{v}_h\|=1} cov(X_h\boldsymbol{u}_h, Y_h\boldsymbol{v}_h) \qquad h = 1 \ldots H$$

Principle:

- Iterative procedure ↦ orthogonal component (latent variable $\boldsymbol{\xi}_h = X_h\boldsymbol{u}_h$).
- successive local regressions on the latent variables.
- X and Y are successively deflated.

Univariate case $Y \in \Re^n$



Univariate case $Y \in \Re^n$

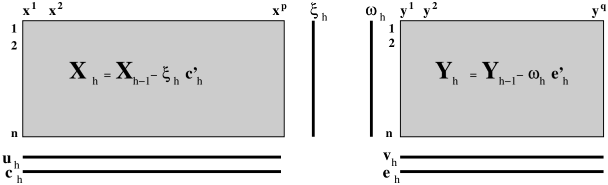
Univariate case $Y \in \Re^n$: step 1 :$\max_{||\boldsymbol{u}||=1} cov(X\boldsymbol{u}, Y)$

$$\begin{aligned} cov(X\boldsymbol{u}, Y) &= (X\boldsymbol{u})^T Y \\ &= < \boldsymbol{u}, X^T Y >= ||X^T Y|| cos(\boldsymbol{u}, X^T Y) \end{aligned}$$

$\hookrightarrow \boldsymbol{u} = \frac{X^T Y}{||X^T Y||}$

- Step 2: find a new linear combination no correlated to $\xi = X\boldsymbol{u}$ which explain the residuals $Y - d\xi$ where d is the regression of Y on $\boldsymbol{\xi} = X\boldsymbol{u}$
- Deflated step: $Y \leftarrow Y - d\xi$ and $X \leftarrow X - \boldsymbol{\xi}\boldsymbol{c}^T$ where $\boldsymbol{c}$ is the regression of X on $\boldsymbol{\xi} = X\boldsymbol{u}$;
- the columns of the new X are orthogonal (no correlated) to $X\boldsymbol{u}$.

Partial Least Squares: regression mode (multivariate case: Y $(n \times q)$)



For each iteration h, $h = 1..H$, decompose X and Y into:

1. Loadings vectors $\mathbf{u}_h$ and $\mathbf{v}_h$, p- and q- dimensional vectors
2. Latent variables $\boldsymbol{\xi}_h$ and ω_h, n-dimensional vectors
3. Regression of X_{h-1} and Y_{h-1} on $\boldsymbol{\xi}_h$ reg. coeff. $\mathbf{c}_h$ and $\mathbf{e}_h$
4. Residual matrices: deflation step of X_{h-1} and Y_{h-1}

Algorithm: regression mode

Objective function:

$$\max_{||\boldsymbol{u}_h||=1, ||\boldsymbol{v}_h||=1} cov(X_h \boldsymbol{u}_h, Y_h \boldsymbol{v}_h) \qquad h = 1 \dots H$$

Start: set w to the first column of Y

1. $\boldsymbol{u} = \frac{X^T w}{w^T w}$, scale $\boldsymbol{u}$ to one. $\boldsymbol{u}$ is the loading vector associated to X
2. $\xi = X\boldsymbol{u}$ is the latent variable associated to X
3. $\boldsymbol{v} = \frac{Y^T \xi}{\xi^T \xi}$, scale $\boldsymbol{v}$ to one. $\boldsymbol{v}$ is the loading vector associated to Y
4. $w = Y\boldsymbol{v}$ is the latent variable associated to Y.
5. If convergence then 6 else 1
6. $\boldsymbol{c} = \frac{X^T \xi}{\xi^T \xi}$, $\boldsymbol{e} = \frac{Y^T \xi}{\xi^T \xi}$ are the partial regression coefficients from the regression of X (Y) onto ξ.
7. Deflation step: Compute the residual matrices $X \leftarrow X - \xi \boldsymbol{c}^T$ and $Y \leftarrow Y - \xi \boldsymbol{e}^T$

PLS family

PLS = Partial Least Squares or Projection to Latent Structures $ $\
Four main methods coexist in the literature:

(i) Partial Least Squares Correlation (PLSC) also called PLS-SVD;

(ii) PLS in mode A (PLS-W2A, for Wold's Two-Block, Mode A PLS);

(iii) PLS in mode B (PLS-W2B) also called Canonical Correlation Analysis (CCA);

(iv) Partial Least Squares Regression (PLSR, or PLS2).

- (i),(ii) and (iii) are symmetric while (iv) is asymmetric.
- Different objective functions to optimise.
- Good news: all use the singular value decomposition (SVD).

PLS connected to Singular Value Decomposition (SVD)

Let a matrix $M : p \times q$ of rank r:

$$M = U\Delta V^T = \sum_{l=1}^{r} \delta_l u_l v_l^T,$$

- $U = (u_l) : p \times p$ and $V = (v_l) : q \times q$ are two orthogonal matrices which contain the normalised left (resp. right) singular vectors
- $\Delta = \mathrm{diag}(\delta_1, \ldots, \delta_r, 0, \ldots, 0)$: the ordered singular values $\delta_1 \geqslant \delta_2 \geqslant \cdots \geqslant \delta_r > 0$.

Connexion between SVD and maximum covariance

We were able to describe the optimization problem of the **four** PLS methods as:

$$(u^*, v^*) = \underset{\|u\|_2=\|v\|_2=1}{\operatorname{argmax}} \; Cov(X_{h-1}u, Y_{h-1}v), \qquad h = 1, \ldots, H$$

Matrices X_h and Y_h are obtained recursively from X_{h-1} and Y_{h-1}.

The four methods differ by the deflation process, chosen so that the above scores or weight vectors satisfy given constraints.

The solution at step h is obtained by computing **only the first** triplet (δ_1, u_1, v_1) of singular elements of the SVD of $M_{h-1} = X_{h-1}^T Y_{h-1}$:

$$(u^*, v^*) = (u_1, v_1)$$

PLS in practice: the `nutrimouse` study

The ‘nutrimouse‘ study contains the expression levels of genes potentially involved in nutritional problems and the concentrations of hepatic fatty acids for forty mice. The data sets come from a nutrigenomic study in the mouse, in which the effects of five regimens with contrasted fatty acid compositions on liver lipids and hepatic gene expression in mice were considered.

PLS in practice: the `nutrimouse` study

Two sets of variables were measured on 40 mice:

- `gene`: the expression levels of 120 genes measured in liver cells, selected among (among about 30,000) as potentially relevant in the context of the nutrition study.
- `lipid`: concentration (in percentage) of 21 hepatic fatty acids measured by gas chromatography.
- `diet`: a 5-level factor. Oils used for experimental diets preparation were corn and colza oils (50/50) for a reference diet (REF), hydrogenated coconut oil for a saturated fatty acid diet (COC), sunflower oil for an Omega6 fatty acid-rich diet (SUN), linseed oil for an Omega3-rich diet (LIN) and corn/colza/enriched fish oils for the FISH diet (43/43/14).
- `genotype` 2-levels factor indicating either wild-type (WT) and PPARα -/- (PPAR).

PLS in practice: the `nutrimouse` study

To illustrate PLS, we will integrate the gene expression levels (`gene`) with the concentrations of hepatic fatty acids (`lipid`).

Set up the data

We first set up the data as X expression matrix and Y as the lipid abundance matrix. We also check that the dimensions are correct and match:

```r
library(mixOmics)
data(nutrimouse)
X <- nutrimouse$gene
Y <- nutrimouse$lipid
dim(X); dim(Y)
```

```
[1]  40 120
```

```
[1] 40 21
```

Quick start

```r
MyResult.pls <- pls(X,Y, ncomp=10)
plotIndiv(MyResult.pls)
```

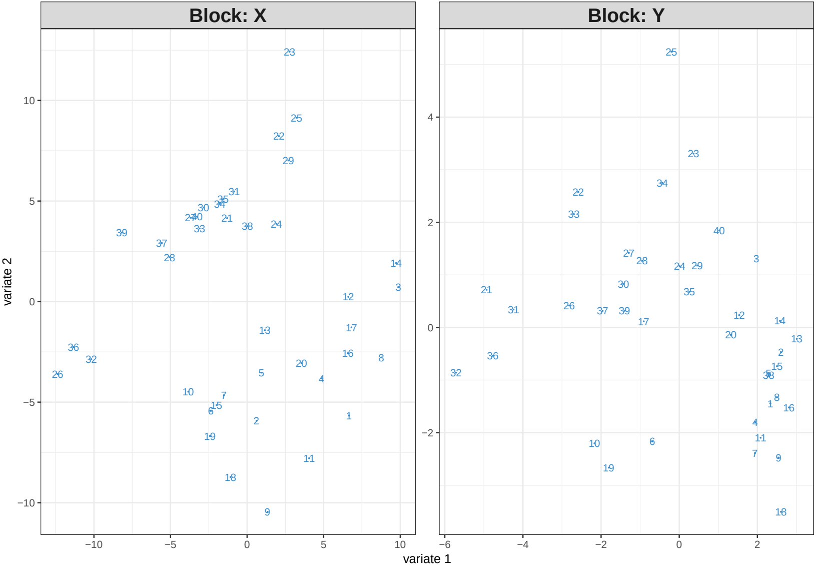
```r
plotVar(MyResult.pls)
```

If you were to run `pls` with minimal code, you would be using the following default values:

- `ncomp = 2`: the first two PLS components are calculated and are used for graphical outputs;
- `scale = TRUE`: data are scaled (variance = 1, strongly advised here);
- `mode = "regression"`: by default a PLS regression mode should be used.

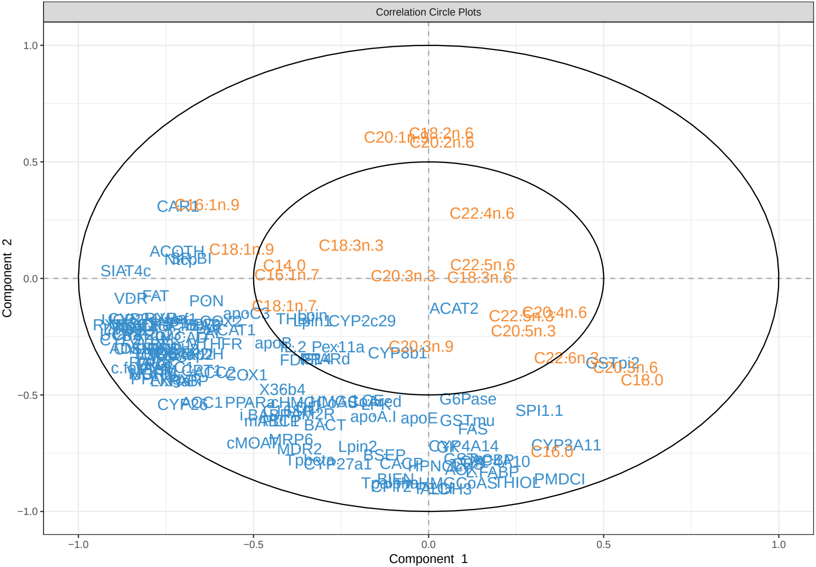
Plot the samples

```
plotIndiv(MyResult.pls)
```



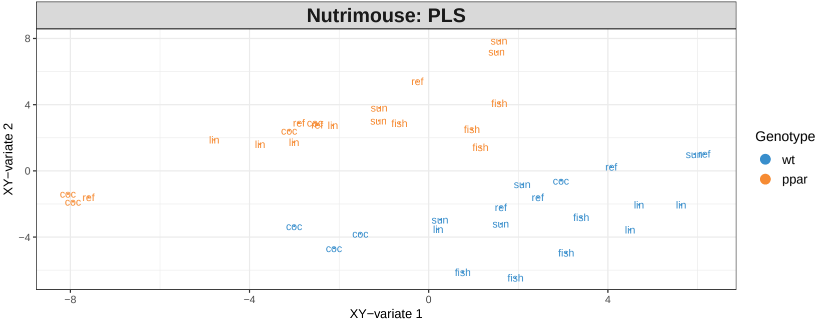
Plot the variables

```
plotVar(MyResult.pls)
```



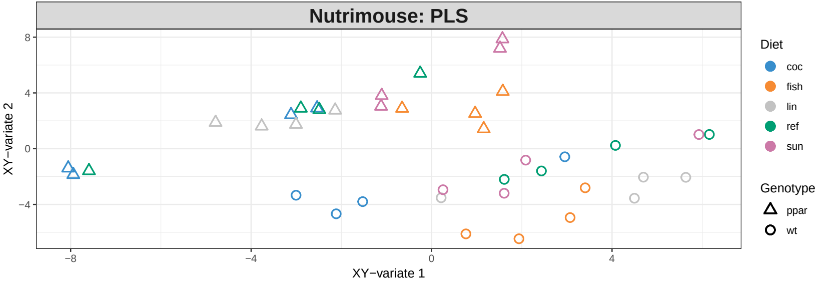
Customize sample plots

```r
plotIndiv(MyResult.pls, group = nutrimouse$genotype,
          rep.space = "XY-variate", legend = TRUE,
          legend.title = 'Genotype',
          ind.names = nutrimouse$diet,
          title = 'Nutrimouse: PLS')
```



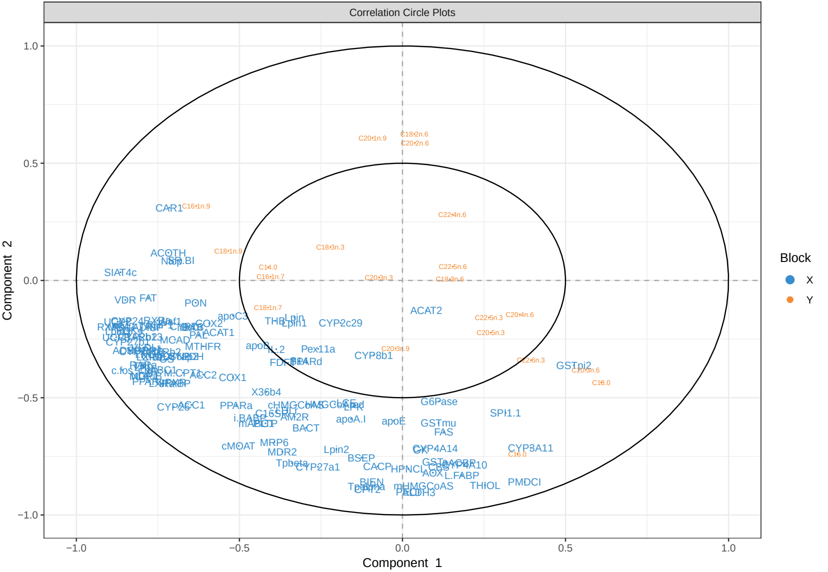
Customize sample plots

```
plotIndiv(MyResult.pls, group=nutrimouse$diet,
          pch = nutrimouse$genotype,
          rep.space = "XY-variate",  legend = TRUE,
          legend.title = 'Diet', legend.title.pch = 'Genotype',
          ind.names = FALSE,
          title = 'Nutrimouse: PLS')
```



Customize variable plots

```
plotVar(MyResult.pls, cex=c(3,2), legend = TRUE)
```



```
coordinates <- plotVar(MyResult.pls, plot = FALSE)
```

Customize variable plots

In this example, the figure is difficult to interpret and we would prefer to use a sparse vesrion of PLS to selecet the most important variable.

A cut-off can be set to display only the variables that mostly contribute to the definition of each component. Those variables should be located towards the circle of radius 1, far from the centre.

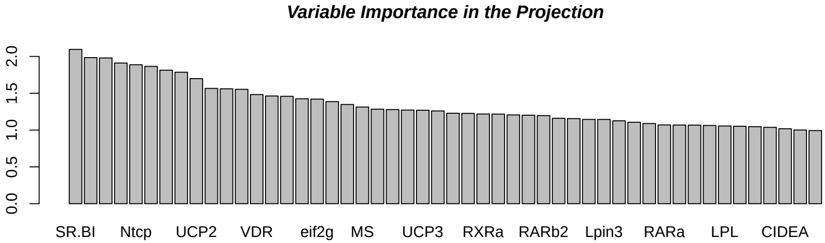
```
plotVar(MyResult.pls, cutoff=0.5)
```

In this particular case, no variable selection was performed. Only the display was altered to show a subset of variables.

Variable Importance in the Projection (VIP)

Variable importance in projection (VIP) coefficients reflect the relative importance of each X variable for each X variate in the prediction model.

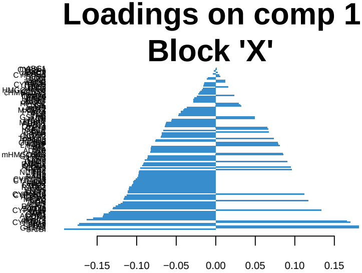
```r
my.vip <- sort(vip(MyResult.pls)[,1],decreasing = TRUE)
barplot(my.vip[1:50],
beside = FALSE,
ylim = c(0, max(my.vip)), legend = rownames(my.vip)[1:50],
main = "Variable Importance in the Projection", font.main = 4)
```



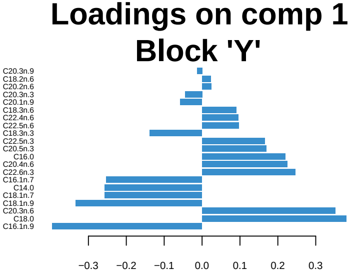
Loading plots

The loading plots help visualise the coefficients assigned to each selected variable on each component:

```
plotLoadings(MyResult.pls, comp = 1, size.name = rel(0.5))
```



Loadings on comp 1
Block 'X'



Loadings on comp 1
Block 'Y'

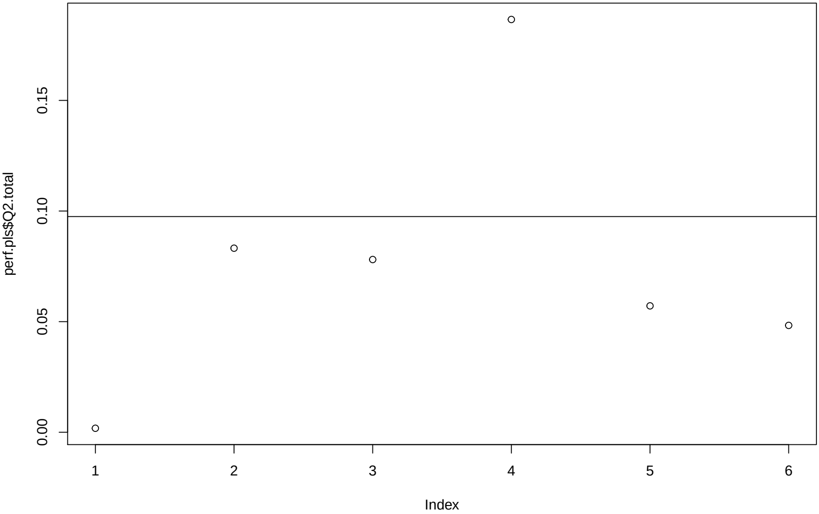
Tuning parameters and numerical outputs

- choose the number of components to retain `ncomp`.
- `perf` function and repeated k-fold cross-validation to calculate the Q^2 criterion used in the SIMCA-P software.
- The rule of thumbs is that a PLS component should be included in the model if its value is $\leqslant 0.0975$. Here we use 5-fold CV repeated 10 times.
- We run a PLS model with a sufficient number of components first, then run `perf` on the object.

```r
MyResult.pls <- pls(X,Y, ncomp = 6)
set.seed(30) # for reproducbility
perf.pls <- perf(MyResult.pls, validation = "Mfold", folds = 5,
                  progressBar = FALSE, nrepeat = 10)
```

Q^2

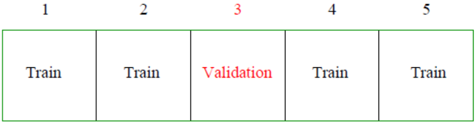
```r
plot(perf.pls$Q2.total)
abline(h = 0.0975)
```



Reminder of Cross-Validation

- One idea is to split the data set into two fractions, then use one portion to fit the model and the other to evaluate how well the estimated modelpredicted the observations in the second portion.
- The problem with this solution is that we rarely have so much data that we can freely part with half of it solely for the purpose of choosing tuning parameters.
- To finesse this problem, cross-validation splits the data into K folds, fits the data on $K - 1$ of the folds, and evaluates risk on the fold that was left out.

K-fold cross validation



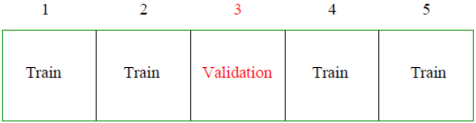
1	2	3	4	5
Train	Train	Validation	Train	Train

Let $k : 1, \ldots, N \rightarrow 1, \ldots, K$ the function indicating the partition to which observation i is allocated:

$$CV = \frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{f}^{-k(i)}(\boldsymbol{X}_i))^2$$

where $\widehat{f}^{-k(i)}$ is the prediction of the subject i based on a model fitted with the $k(i)$th part of the data removed.

M-K-fold cross validation



1	2	3	4	5
Train	Train	Validation	Train	Train

Let $k : 1, \ldots, N \to 1, \ldots, K$ the function indicating the partition to which observation i is allocated:

$$CV = \frac{1}{M} \sum \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{f}^{-k(i)}(\boldsymbol{X}_i))^2 \right)$$

where $\widehat{f}^{-k(i)}$ is the prediction of the subject i based on a model fitted with the $k(i)$th part of the data removed. \end{frame}

Cross-validation: Leave One Out (loo)

- $\widehat{Y}_i$ is the prediction of the i-th observation obtained with the model fitted on all the observation.
- $\widehat{Y}_i^{(-i)}$ is the prediction of the i-th observation obtained with the model fitted on all the observation except the i-th observation.

The cross-validation approach compares predictions $\widehat{Y}_i^{(-i)}$ to observations Y_i.

Choice of the number of latent variables H using Q_H^2

Determine $\widehat{H}$ by cross-validation

For each $H = 1 \ldots n$:

1. Evaluate $\widehat{Y}_i^H$ and $\widehat{Y}_i^{H(-i)}$
2. Evaluate *Residual Sum of Squares* : $RSS_H = \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i^H\right)^2$
3. Evaluate *PRediction Error Sum of Squares* :
   $PRESS_H = \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i^{H(-i)}\right)^2$
4. Evaluate $Q_H^2 = 1 - \dfrac{PRESS_H}{RSS_{H-1}}$

Take Home Message: PLS

- Dimension Reduction approach for 2 blocs of Data

- Supervised method

- Finds successive pairs of latent (score) vector which are most correlated

- Symmetric relationship. Analyse the shared information.

-Asymmetric relationship. There is a set of response and predictor variables that can be used for prediction

Take Home Message: PLS

- Dimension Reduction approach for 2 blocs of Data

- Supervised method

- Finds successive pairs of latent (score) vector which are most correlated

- Symmetric relationship. Analyse the shared information.

-Asymmetric relationship. There is a set of response and predictor variables that can be used for prediction

Difficult to interpret latent variables when there are too many variables.

How to make variable selection with PLS ?