

Partial Least Squares Regression

-- an introductory tutorial to some of the most important ideas in multivariate regression.

Instructor: Nam Sun Wang

Partial Least Squares (PLS).

In principal component regression, we calculate the principal components based on the largest variation in the X data alone, without any consideration paid to Y. However, not all independent variables affect the dependent variable in the same manner or with the same intensity. Some independent variables may vary a lot but have very little influence on the dependent variable y, while others do not vary as much but dominate the outcome. The difference in the variance among different independent variables can be normalized with variance-scaling. The last example above exposes a problem with principal component regression. Now, we want to determine which component in the X space explains the most variation in Y. This is the component we want to regress against first so that we capture most of the variation in Y with just the first few components. A major task in partial least squares is finding the vector that lives in the X space such that it explains as much variation in Y as possible. In partial least squares regression, we break down the independent and dependent matrices X and Y into scores and loadings. We follow the notation of P. Geladi and B. R. Kowalski (*Analytica Chimica Acta*, **185**, p1-17, 1986) with the exception that our vectors are column vectors, whereas, Geladi and Kowalski's are a mix of row vectors and column vectors. In Mathcad, mathematical manipulations are much easier with column vectors. Some people also like to work strictly with column vectors. The outer relations are:

$$X = t \cdot p \quad Y = u \cdot q \quad \text{Our column vector notation:} \quad X = t \cdot p^T \quad Y = u \cdot q^T$$

We regress t (the score vector of X) and u (the score vector of Y) to find the regression coefficient b.

$$u = t \cdot b$$

The next table is a summary of the scores and loadings and their relationship to each other in partial least squares regression. The first set of relationships contain both row and column vectors. This notation graphically resembles the physical dimensions of the X and Y matrices. The second set of relationships contain only column vectors.

vectors	original notation	column vector notation
1. direction of projection for X:	$w = u^T \cdot X$	$w = \alpha_w \cdot X^T \cdot u$
2. score (column) vector for X:	$t = X \cdot w^T$	$t = X \cdot w$
3. loading (row) vector for Y:	$q = t^T \cdot Y$	$q = \alpha_q \cdot Y^T \cdot t$
4. score (column) vector for Y:	$u = Y \cdot q^T$	$u = Y \cdot q$
5. loading (row) vector for X:	$p = t^T \cdot X$	$p = X^T \cdot t = X^T \cdot X \cdot w$

The interpretation of each vector is as follows:

1. w is a directional vector whose components are proportional to the extent of overlap between X and u. Since w is normalized, there really should be a normalization factor $\alpha_w = 1/|X^T \cdot u|$ in the equation.
2. t is clearly the projection of X along w.
3. q is a directional vector whose components are proportional to the extent of overlap between Y and t. Since q is normalized, there should be a normalization factor $\alpha_q = 1/|Y^T \cdot t|$ in the equation.
4. u is clearly the projection of Y along q.

we see that u depends on q, which in turn depends on t, which in turn depends on w, which in turn depends on u. Thus, we end with the variable u that we have started with. (Bad English grammar here, but the sentence illustrates that these four variables (w, u, t, and q) trace a full circle. If we string these variables together, we see that w is the eigenvector of $X^T \cdot Y \cdot Y^T \cdot X$ that corresponds to an eigenvalue of $1/(\alpha_w \cdot \alpha_q) = |X^T \cdot u| \cdot |Y^T \cdot t|$, and so on.

vector	eigenvector of	explanation #1	explanation #2	eigenvalue
w	$X^T \cdot Y \cdot Y^T \cdot X$	$X^T \cdot X$ weighted by $Y \cdot Y^T$	$(Y^T \cdot X)^T \cdot (Y^T \cdot X)$	$ X^T \cdot u \cdot Y^T \cdot t $
t	$X \cdot X^T \cdot Y \cdot Y^T$	$X \cdot Y^T$ weighted by $X^T \cdot Y$		
q	$Y^T \cdot X \cdot X^T \cdot Y$	$Y^T \cdot Y$ weighted by $X \cdot X^T$	$(X^T \cdot Y)^T \cdot (X^T \cdot Y)$	
u	$Y \cdot Y^T \cdot X \cdot X^T$	$Y \cdot X^T$ weighted by $Y^T \cdot X$		

One way is to view the covariance matrix $X^T \cdot Y \cdot Y^T \cdot X$ as the weighed covariance matrix $X^T \cdot X$, where the weighting factor is $Y \cdot Y^T$. The data points in X that contribute more heavily to Y are weighted more heavily. Since principal component regression is based on the eigenvectors of $X^T \cdot X$, we can view partial least squares as weighted principal component regression. Another interpretation is that $(X^T \cdot Y) \cdot (Y^T \cdot X) = (Y^T \cdot X)^T \cdot (Y^T \cdot X)$ is the covariance matrix of $Y^T \cdot X$, which is the inner products of Y and X along the directions of $x^{<0>}$, $x^{<1>}$, $x^{<2>}$, ... Qualitatively, it is the length of the shadow of Y projected onto X, or, equivalently, the shadow of X projected onto Y. In principal component regression, the eigenvector that corresponds to the largest eigenvalue of $X^T \cdot X$ captures the most variations in the $X^T \cdot X$ covariance matrix. Similarly, the eigenvector that corresponds to the largest eigenvalue of $(Y^T \cdot X)^T \cdot (Y^T \cdot X)$ captures the most variations in the $(Y^T \cdot X)^T \cdot (Y^T \cdot X)$ matrix. In other words, this eigenvector gives the direction in the X space that accounts for the most variation in the Y space.

In summary, the steps for partial least squares regression are:

- Step 0. Provide X and Y vectors.
- Step 1. Mean-center X and Y. If necessary, normalize the columns of X and Y with the respective covariance (variance-scaling).
- Step 2. Find the basis vectors w by examining X and Y. These are the eigenvectors of $X^T \cdot Y \cdot Y^T \cdot X$.
- Step 3. Find the portion of the X matrix that lies along the basis vector w. $score_x = X \cdot w$
Find the portion of the Y matrix that lies along the basis vector q. $score_y = Y \cdot q^T$
- Step 4. Apply the scalar normal equation to find regression coefficient b:
 $b = (score_x^T \cdot score_x)^{-1} \cdot score_x^T \cdot score_y$.
- Step 5. Compute the residual of X and Y.
- Step 6. The regression equation is: $y_{regress} = score_x \cdot b = X \cdot w \cdot b$.
(Take care of mean-centering and variance-scaling as needed.)
- Step 7. Steps 2-6 are performed with one basis vector at a time.
Repeat Steps 2-6 for each additional basis vector.

Below, we will numerically demonstrate these steps.

Step 0. Generate X and Y Data. The structure of X is identical to the last worksheet in this series, namely, the first two independent variables $x^{<0>}$ and $x^{<1>}$ are mostly dependent, $x^{<0>}$ being 10 times of $x^{<1>}$.

Number of points: $N := 50$ $i := 0..N$

Dimension: $m := 2$ $j := 0..m$

$$X^{<i>} := (\text{rnd}(1) - 0.5) \cdot \begin{pmatrix} 10 \\ 1 \\ 0 \end{pmatrix} + (\text{rnd}(1) - 0.5) \cdot \begin{pmatrix} 0 \\ 0.001 \\ 0 \end{pmatrix} + (\text{rnd}(1) - 0.5) \cdot \begin{pmatrix} 0 \\ 0 \\ 0.1 \end{pmatrix} \quad X := X^T$$

↑ Without this small noise term, $x^{<0>}$ and $x^{<1>}$ are completely dependent and $X^T \cdot X$ is singular.

Following the same example as in the last worksheet in this series, the dependent variable Y depends most heavily on $x^{<2>}$ and to a minor extent on $x^{<1>}$.

$$Y_i := 0.1 \cdot X_{i,1} + 100 \cdot X_{i,2} + (\text{rnd}(0.1) - 0.05) \quad y(x) := x \cdot \begin{pmatrix} 0 \\ 0.1 \\ 100 \end{pmatrix}$$

Step 1a. Mean-Centering:

$$X_{\text{save}} := X \quad Y_{\text{save}} := Y \quad (\text{save them for use later.})$$

$$x_{\text{mean}_j} := \text{mean}(X^{<j>}) \quad x_{\text{mean}} := x_{\text{mean}}^T \quad X^{<j>} := X^{<j>} - x_{\text{mean}_{0,j}}$$

$$x_{\text{mean}} = (-0.318 \quad -0.032 \quad 0.003)$$

$$y_{\text{mean}} := \text{mean}(Y) \quad Y := Y - y_{\text{mean}} \quad y_{\text{mean}} = 0.311$$

Step 1b. Variance-Scaling:

$$x_{\text{stdev}_j} := \text{stdev}(X^{<j>}) \quad x_{\text{stdev}} := x_{\text{stdev}}^T \quad X^{<j>} := \frac{X^{<j>}}{x_{\text{stdev}_{0,j}}}$$

$$x_{\text{stdev}} = (2.813 \quad 0.281 \quad 0.028)$$

Step 2. Find the eigenvalues and eigenvectors of the mean-centered and variance-scaled covariance matrix $X^T \cdot Y \cdot Y^T \cdot X$.

Covariance matrix:

$$W := X^T \cdot Y \cdot Y^T \cdot X \quad W = \begin{bmatrix} 2.456 \cdot 10^3 & 2.455 \cdot 10^3 & 7.012 \cdot 10^3 \\ 2.455 \cdot 10^3 & 2.454 \cdot 10^3 & 7.008 \cdot 10^3 \\ 7.012 \cdot 10^3 & 7.008 \cdot 10^3 & 2.002 \cdot 10^4 \end{bmatrix} \quad |W| = 0 \quad \leftarrow \text{Singular.}$$

Eigenvalue/eigenvector

$$\lambda := \text{reverse}(\text{sort}(\text{eigenvals}(W))) \quad \lambda^T = (2.493 \cdot 10^4 \quad -1.698 \cdot 10^{-12} \quad -1.819 \cdot 10^{-12})$$

↑ Only one nonzero eigenvalue.

$$w^{<j>} := \text{eigenvec}(W, \lambda_j) \quad w = \begin{pmatrix} 0.314 & -0.59 & 0.542 \\ 0.314 & -0.675 & 0.716 \\ 0.896 & 0.443 & -0.44 \end{pmatrix}$$

Note that w appears not to be orthogonal, although the eigenvectors of a symmetric real matrix should theoretically be orthogonal, i.e., $w^T \cdot w = w \cdot w^T = 1$. This is because Mathcad returns an almost identical eigenvector for an almost repeated eigenvalue, despite that we should get

different, linearly independent, orthogonal eigenvectors even when eigenvalues are repeated.

$$w^T \cdot w = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -0.998 \\ 0 & -0.998 & 1 \end{pmatrix} \quad |w| = -0.064$$

The diagonal terms' being unity tells us that the eigenvectors are normalized to unity. The off-diagonal terms being zero means that the corresponding eigenvectors are orthogonal. Alternatively, we can test one vector at a time.

$$w^{<0>T} \cdot w^{<1>} = 0 \quad w^{<0>T} \cdot w^{<2>} = 0 \quad \leftarrow w^{<0>} \text{ and } w^{<1>} \text{ are orthogonal.}$$

$$w^{<1>T} \cdot w^{<2>} = -0.998 \quad \leftarrow \text{But, Mathcad says } w^{<1>} \text{ and } w^{<2>} \text{ obtained numerically are highly dependent, which is theoretically false. This is a numerical error because of the closely repeated eigenvalues.}$$

Check: Let us try to extract the most amount of information in U, score of Y.

Cross-Covariance matrix: $U := Y \cdot Y^T \cdot X \cdot X^T$

Eigenvalue/eigenvector of U

$$\lambda_u := \text{reverse}(\text{sort}(\text{eigenvals}(U)))$$

$$\lambda_u^T = \begin{array}{|c|c|c|c|} \hline & 0 & 1 & 2 \\ \hline 0 & 2.493 \cdot 10^4 & 3.934 \cdot 10^{-13} & 2.664 \cdot 10^{-13} + 8.342 \cdot 10^{-14}i \\ \hline \end{array}$$

$$u := \text{eigenvec}(U, \lambda_{u_0})$$

↑ Only one nonzero eigenvalue.

$$w := X^T \cdot u \quad \text{normalize: } w := \frac{w}{|w|}$$

$$w = \begin{pmatrix} -0.314 \\ -0.314 \\ -0.896 \end{pmatrix} \quad \leftarrow \text{This is identical to (the negative of) the 0th } w \text{ vector.} \rightarrow w^{<0>} = \begin{pmatrix} 0.314 \\ 0.314 \\ 0.896 \end{pmatrix}$$

Check: Let us compare the eigenvector of $X \cdot X^T \cdot Y \cdot Y^T$ to the score vector of x.

Cross-Covariance matrix: $T := X \cdot X^T \cdot Y \cdot Y^T$

Eigenvalue/eigenvector of T

$$\lambda_t := \text{reverse}(\text{sort}(\text{eigenvals}(T)))$$

$$\lambda_t^T = \begin{array}{|c|c|c|c|} \hline & 0 & 1 & 2 \\ \hline 0 & 2.493 \cdot 10^4 & 6.208 \cdot 10^{-13} & 2.992 \cdot 10^{-13} \\ \hline \end{array}$$

$$t := \text{eigenvec}(T, \lambda_{t_0})$$

↑ Only one nonzero eigenvalue.

$$t \cdot \frac{(X \cdot w^{<0>})_0}{t_0} = \begin{array}{|c|c|} \hline & 0 \\ \hline 0 & -0.87 \\ \hline 1 & -1.423 \\ \hline 2 & -0.888 \\ \hline 3 & -2.054 \\ \hline 4 & -0.799 \\ \hline 5 & -2.213 \\ \hline \end{array}$$

← Identical value when t (the score vector of X is not normalized) →

$$t = X \cdot w$$

$$X \cdot w^{<0>} = \begin{array}{|c|c|} \hline & 0 \\ \hline 0 & -0.87 \\ \hline 1 & -1.423 \\ \hline 2 & -0.888 \\ \hline 3 & -2.054 \\ \hline 4 & -0.799 \\ \hline 5 & -2.213 \\ \hline \end{array}$$

Check: Of course, Y is not decomposed at all or described with a rotated set of coordinates when there is only one dependent variable.

Cross-Covariance matrix: $Q := Y^T \cdot X \cdot X^T \cdot Y \quad \leftarrow Q \text{ is simply a } 1 \times 1 \text{ matrix.}$

Eigenvalue/eigenvector of Q

$$\lambda_Q := \text{reverse}(\text{sort}(\text{eigenvals}(Q))) \quad \lambda_Q^T = 2.493 \cdot 10^4$$

$$q := \text{eigenvec}(Q, \lambda_{q_0}) \quad \uparrow \text{Only one (nonzero) eigenvalue, period.}$$

$$q = 1 \quad \leftarrow \text{Just a scalar when the number of variables in Y is one.}$$

Since Mathcad fails to return two distinct eigenvectors for repeated eigenvalues, we need to manually find the second eigenvector for repeated eigenvalues, perhaps through an orthogonalization process. At any rate, at this point, we only need one eigenvector that corresponds to the largest eigenvalue. This 0th eigenvector tells us the initial direction to project the independent variable X. In partial least squares, we do not find all the directions simultaneously *a priori*. Instead, we take away from X its 0th score, i.e., the projection of X onto the 0th eigenvector. Similarly, and we take away from Y the part that is explained by regressing against the 0th score of X. We then repeat an identical process with the residuals of X and Y.

Step 3. Score and loading vectors for X and Y.

$$\text{score (column) vector for X: } t^{<0>} := X \cdot w^{<0>}$$

$$\text{loading (row) vector for Y: } q^{<0>} := Y^T \cdot t^{<0>}$$

$$\text{normalize: } \alpha_q := |q^{<0>}| \quad q^{<0>} := \frac{q^{<0>}}{|q^{<0>}|} \quad q = 1$$

$$\alpha_q = 157.887$$

$$\text{score (column) vector for Y: } u^{<0>} := Y \cdot q^{<0>}$$

$$\text{loading (row) vector for X: } p^{<0>} := X^T \cdot t^{<0>}$$

$$\text{normalize: } \alpha_p := |p^{<0>}| \quad p^{<0>} := \frac{p^{<0>}}{|p^{<0>}|} \quad p = \begin{pmatrix} 0.541 \\ 0.541 \\ 0.644 \end{pmatrix}$$

$$\alpha_p = 87.828$$

Check: Find w as the inner product between X and u and compare this to the eigenvector.

$$\omega^{<0>} := X^T \cdot u^{<0>}$$

$$\text{normalize: } \alpha_\omega := |\omega^{<0>}| \quad \omega^{<0>} := \frac{\omega^{<0>}}{|\omega^{<0>}|}$$

$$\alpha_\omega = 157.887$$

$$\omega^{<0>} = \begin{pmatrix} 0.314 \\ 0.314 \\ 0.896 \end{pmatrix} \quad \leftarrow \text{Compare} \rightarrow \quad w^{<0>} = \begin{pmatrix} 0.314 \\ 0.314 \\ 0.896 \end{pmatrix} \quad \overrightarrow{(\omega^{<0>} = w^{<0>})} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Check: The eigenvalue is the product of the two normalization factors α_ω and α_q .

$$\alpha_\omega \cdot \alpha_q = 2.493 \cdot 10^4 \quad \leftarrow \text{Compare} \rightarrow \quad \lambda_0 = 2.493 \cdot 10^4 \quad (\alpha_\omega = \alpha_q) = 1$$

Note that $p^{<0>} = X^T \cdot t^{<0>}$ is the projection of X in the $t^{<0>}$ direction (in X space); it indicates the extent of commonality between the X data and the score vector for X. Whereas, $w^{<0>} = X^T \cdot u^{<0>}$ is the projection of X in the $u^{<0>}$ direction (in Y space); it indicates the extent of commonality between the X data and the score vector for Y. They are related by a factor of $X^T \cdot X$ because $p^{<0>} = X^T \cdot t^{<0>} / |X^T \cdot t^{<0>}| = X^T \cdot X \cdot w^{<0>} / |X^T \cdot X \cdot w^{<0>}|$.

$$\frac{X^T \cdot X \cdot w^{<0>}}{|X^T \cdot X \cdot w^{<0>}|} = \begin{pmatrix} 0.541 \\ 0.541 \\ 0.644 \end{pmatrix} \quad \leftarrow \text{Compare} \rightarrow \quad p^{<0>} = \begin{pmatrix} 0.541 \\ 0.541 \\ 0.644 \end{pmatrix}$$

Note that in principal component regression, because the weighting vector v is the eigenvector of $X^T \cdot X$, the weighting vector is identical to the loading vector (after normalization to unity):

$$\text{load}^{<0>} = X^T \cdot \text{score}^{<0>} = X^T \cdot X \cdot v^{<0>} = \lambda_0 \cdot v^{<0>} \rightarrow v^{<0>}. \quad (\text{True in PCR})$$

In partial least squares, the weighting vector w and the loading vector p are not identical. The following is not true because $w^{<0>}$ is not an eigenvector of $X^T \cdot X$.

$$p^{<0>} = \text{load}^{<0>} = X^T \cdot \text{score}^{<0>} = X^T \cdot t^{<0>} = X^T \cdot X \cdot w^{<0>} = \lambda_{w0} \cdot w^{<0>} \rightarrow w^{<0>}. \quad (\text{Not true in PLS})$$

As a result, $w^{<0>}$ and $p^{<0>}$, as calculated above during the 0th step of partial least squares regression, are not perfectly aligned with each other.

$$w^{<0>} \cdot p^{<0>} = 0.917 \quad \theta := \text{acos}(w^{<0>} \cdot p^{<0>}) \quad \theta = 23.564 \cdot \text{deg}$$

Because the eigenvalue-eigenvector relationship gives $\lambda \cdot w = X^T \cdot Y \cdot Y^T \cdot X \cdot w$ while the recursive relationship gives $w = X^T \cdot Y \cdot Y^T \cdot X \cdot w / |q|_{\text{old}}$, one of the vectors have to be modified by a constant.

Here, we choose to increase $w^{<0>}$ so that the projection of $w^{<0>}$ into $p^{<0>}$ results in a unit vector. ($w^{<0>}$ itself will no longer be a unit vector.)

$$t^{<0>} \cdot p^{<0>}^T = \frac{X \cdot w^{<0>}}{w^{<0>} \cdot w^{<0>}} \cdot w^{<0>}^T$$

$$w^{<0>} := w^{<0>} \cdot \frac{1}{w^{<0>} \cdot p^{<0>}} \quad w^{<0>} = w^{<0>} \cdot \frac{\alpha_p}{\left(w^{<0>}^T \cdot X^T \cdot X \cdot w^{<0>} \right)_0} = w^{<0>} \cdot \frac{|X^T \cdot X \cdot w^{<0>}|}{|w^{<0>}^T \cdot X^T \cdot X \cdot w^{<0>}|}$$

Corrected weighting vector.

$$w^{<0>} = \begin{pmatrix} 0.342 \\ 0.342 \\ 0.978 \end{pmatrix} \quad |w^{<0>}| = 1.091$$

↑ An equivalent form that is toggled off because $p^{<0>} = X^T \cdot t^{<0>} = X^T \cdot (X \cdot w^{<0>}) = (X^T \cdot X) \cdot w^{<0>}$.

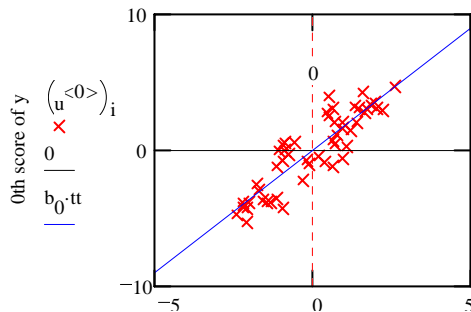
Re-calculate the score vectors for X and Y, starting with the corrected vector. Since the loading vectors have already been normalized, there is no need to repeat the calculation for a change arising from a constant factor.

$$\text{score (column) vector for X: } t^{<0>} := X \cdot w^{<0>}$$

Step 4. Regression.

$$b_0 := \text{slope}(t^{<0>}, u^{<0>}) \quad b = 1.798$$

$$\text{Check: } \text{intercept}(t^{<0>}, u^{<0>}) = 0 \quad \text{tt} := -5, -4.9 \dots 5$$



$(t^{<0>})_{i,tt}$
0th score of x

Step 5. Compute the residual matrices.

$$E := X - t^{<0>} \cdot p^{<0>T} \quad \text{In projection notation,} \quad E := X - X \cdot w^{<0>} \cdot p^{<0>T}$$

$$F := Y - t^{<0>} \cdot q^{<0>T} \cdot b_0$$

Check: Naturally, after all the components that are lined up with $w^{<0>}$ have been taken out from X, the remaining residual E has nothing in common with $w^{<0>}$.

$$E \cdot w^{<0>} = \begin{array}{|c|c|} \hline & 0 \\ \hline 0 & 0 \\ \hline 1 & 0 \\ \hline 2 & 0 \\ \hline 3 & 0 \\ \hline 4 & 0 \\ \hline \end{array} \quad E = \begin{array}{|c|c|c|} \hline & 0 & 1 \\ \hline 0 & -1.146 & -1.147 \\ \hline 1 & 0.421 & 0.422 \\ \hline 2 & 1.385 & 1.385 \\ \hline 3 & 0.072 & 0.073 \\ \hline \end{array} \quad E = \begin{array}{|c|c|c|} \hline & 0 & 1 \\ \hline 0 & -1.146 & -1.147 \\ \hline 1 & 0.421 & 0.422 \\ \hline 2 & 1.385 & 1.385 \\ \hline 3 & 0.072 & 0.073 \\ \hline \end{array}$$

Step 6. Goodness of fit.

$$\begin{aligned} \text{sse}_{\text{old}} &:= Y \cdot Y & \text{sse}_{\text{old}} &= 392.614 \\ \text{sse} &:= F \cdot F & \text{sse} &= 82.964 \\ r^2 &:= \frac{\text{sse}_{\text{old}} - \text{sse}}{\text{sse}_{\text{old}}} & r^2 &= 78.869\% \\ r &:= \sqrt{r^2} & r &= 88.808\% \end{aligned}$$

Note that in principal component regression, the 0th component/factor captured only about $r^2=12\%$ of the variations in Y. With variance-centering, the amount captured by the 0th component increased to about $r^2=31\%$. The 1st component then went on to capture most of Y. This is because while $x^{<2>}$ heavily affected Y, it did not vary as much as $x^{<0>}$ and $x^{<1>}$. Principal component regression identifies and selects a component in the dependent variable X that accounts for most variation in the X data. On the other hand, partial least squares picks a component in the dependent variable X that accounts for most variation in the Y data. In partial least squares regression here, the 0th factor captured $r^2=79\%$.

2nd Iteration. If we want to capture even more information in Y, we iterate with the next factor until the sse value flattens out. We start with the residual independent variables E and F.

Covariance matrix: $W := E^T \cdot F \cdot F^T \cdot E$

$$W = \begin{bmatrix} 1.903 \cdot 10^3 & 1.904 \cdot 10^3 & -1.333 \cdot 10^3 \\ 1.904 \cdot 10^3 & 1.904 \cdot 10^3 & -1.334 \cdot 10^3 \\ -1.333 \cdot 10^3 & -1.334 \cdot 10^3 & 933.904 \end{bmatrix}$$

Eigenvalue: $\lambda := \text{reverse}(\text{sort}(\text{eigenvals}(W)))$ $\lambda^T = (4.742 \cdot 10^3 \quad 2.575 \cdot 10^{-13} \quad -8.285 \cdot 10^{-13})$
 \uparrow Only one nonzero eigenvalue.

Eigenvector: $w^{<1>} := \text{eigenvec}(W, \lambda_0)$

$$w = \begin{pmatrix} 0.342 & 0.634 & 0.542 \\ 0.342 & 0.634 & 0.716 \\ 0.978 & -0.444 & -0.44 \end{pmatrix}$$

Note that we kept the original eigenvector $w^{<0>}$ unchanged and updated only the next eigenvector $w^{<1>}$ from the residual matrices. The updated eigenvector $w^{<1>}$ is not identical to that calculated earlier on this worksheet corresponding to the next largest eigenvalue of $X^T \cdot Y \cdot Y^T \cdot X$. These two eigenvectors ($w^{<0>}$ and $w^{<1>}$) are mutually orthogonal. This is necessarily so, because $w^{<1>}$ is calculated from the residual matrices, and the residual matrices do not contain any component aligned with $w^{<0>}$. (Remember that $E \cdot w^{<0>} = 0$.)

$$w^{<0>T} \cdot w^{<1>} = 0$$

score (column) vector for X: $t^{<1>} := E \cdot w^{<1>}$

loading (row) vector for Y: $q^{<1>} := F^T \cdot t^{<1>}$ normalize: $q^{<1>} := \frac{q^{<1>}}{|q^{<1>}|}$ $q^{<1>} = -1$

score (column) vector for Y: $u^{<1>} := F \cdot q^{<1>}$

loading (row) vector for X: $p^{<1>} := E^T \cdot t^{<1>}$ normalize: $p^{<1>} := \frac{p^{<1>}}{|p^{<1>}|}$ $p^{<1>} = \begin{pmatrix} 0.634 \\ 0.634 \\ -0.444 \end{pmatrix}$

The correction step is not necessary because the correction factor is unity.

In other words, $w^{<1>} = p^{<1>}$.

$$\frac{1}{w^{<1>} \cdot p^{<1>}} = 1$$

$$w^{<1>} = \begin{pmatrix} 0.634 \\ 0.634 \\ -0.444 \end{pmatrix} \quad \frac{E^T \cdot u^{<1>}}{|E^T \cdot u^{<1>}|} = \begin{pmatrix} 0.634 \\ 0.634 \\ -0.444 \end{pmatrix} \quad \leftarrow \text{Compare} \rightarrow \quad p^{<1>} = \begin{pmatrix} 0.634 \\ 0.634 \\ -0.444 \end{pmatrix}$$

On the other hand, although we calculate $w^{<1>}$ from $E^T \cdot u^{<1>}$ and $p^{<1>}$ from $E^T \cdot t^{<1>}$, but because $w^{<1>}$ and $p^{<1>}$ are normalized, $p^{<1>} = w^{<1>}$ does not imply that $t^{<1>}$ and $u^{<1>}$ are identical.

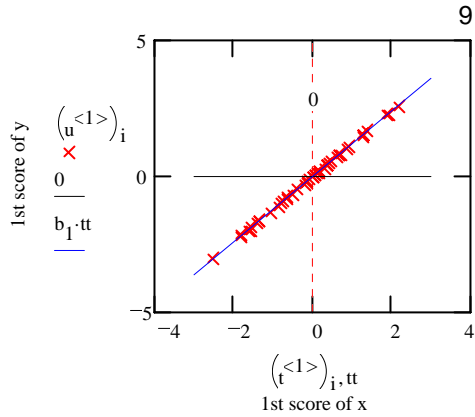
$t^{<1>} =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 20px;">0</td><td style="width: 50px;">-1.809</td></tr> <tr><td>1</td><td>0.665</td></tr> <tr><td>2</td><td>2.186</td></tr> <tr><td>3</td><td>0.114</td></tr> <tr><td>4</td><td>-1.832</td></tr> <tr><td>5</td><td>0.368</td></tr> </table>	0	-1.809	1	0.665	2	2.186	3	0.114	4	-1.832	5	0.368	$\leftarrow \text{Compare} \rightarrow$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 20px;">0</td><td style="width: 50px;">-2.156</td></tr> <tr><td>1</td><td>0.822</td></tr> <tr><td>2</td><td>2.593</td></tr> <tr><td>3</td><td>0.111</td></tr> <tr><td>4</td><td>-2.223</td></tr> <tr><td>5</td><td>0.434</td></tr> </table>	0	-2.156	1	0.822	2	2.593	3	0.111	4	-2.223	5	0.434
0	-1.809																										
1	0.665																										
2	2.186																										
3	0.114																										
4	-1.832																										
5	0.368																										
0	-2.156																										
1	0.822																										
2	2.593																										
3	0.111																										
4	-2.223																										
5	0.434																										

We note that $t^{<1>} = E \cdot w^{<1>} \neq X \cdot w^{<1>}$.

$X \cdot w^{<1>} =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 20px;">0</td><td style="width: 50px;">-2.189</td></tr> <tr><td>1</td><td>0.044</td></tr> <tr><td>2</td><td>1.798</td></tr> <tr><td>3</td><td>-0.781</td></tr> <tr><td>4</td><td>-2.181</td></tr> <tr><td>5</td><td>-0.598</td></tr> </table>	0	-2.189	1	0.044	2	1.798	3	-0.781	4	-2.181	5	-0.598	$\leftarrow \text{Compare} \rightarrow$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 20px;">0</td><td style="width: 50px;">-1.809</td></tr> <tr><td>1</td><td>0.665</td></tr> <tr><td>2</td><td>2.186</td></tr> <tr><td>3</td><td>0.114</td></tr> <tr><td>4</td><td>-1.832</td></tr> <tr><td>5</td><td>0.368</td></tr> </table>	0	-1.809	1	0.665	2	2.186	3	0.114	4	-1.832	5	0.368
0	-2.189																										
1	0.044																										
2	1.798																										
3	-0.781																										
4	-2.181																										
5	-0.598																										
0	-1.809																										
1	0.665																										
2	2.186																										
3	0.114																										
4	-1.832																										
5	0.368																										

Regression: $b_1 := \text{slope}(t^{<1>}, u^{<1>})$ $b = \begin{pmatrix} 1.798 \\ 1.204 \end{pmatrix}$

Check: intercept($t^{<1>}, u^{<1>}$) = 0 $tt := -3, -2.9..3$



Residual matrices: $E := E - t^{<1>} \cdot p^{<1>T}$

$$F := F - t^{<1>} \cdot q^{<1>T} \cdot b_1$$

Check: Naturally, after all the components that are lined up with $w^{<0>}$ and $w^{<1>}$ have been taken out from X, the remaining residual E has nothing in common with neither $w^{<0>}$ nor $w^{<1>}$.

$E \cdot w^{<0>}$	=	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0</td></tr> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table>	0	0	1	2	3	4	5
0									
0									
1									
2									
3									
4									
5									

$E \cdot w^{<1>}$	=	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0</td></tr> <tr><td>$-5.769 \cdot 10^{-14}$</td></tr> <tr><td>$2.118 \cdot 10^{-14}$</td></tr> <tr><td>$6.927 \cdot 10^{-14}$</td></tr> <tr><td>$3.636 \cdot 10^{-15}$</td></tr> </table>	0	$-5.769 \cdot 10^{-14}$	$2.118 \cdot 10^{-14}$	$6.927 \cdot 10^{-14}$	$3.636 \cdot 10^{-15}$
0							
$-5.769 \cdot 10^{-14}$							
$2.118 \cdot 10^{-14}$							
$6.927 \cdot 10^{-14}$							
$3.636 \cdot 10^{-15}$							

Goodness of fit: $sse := F \cdot F$ $sse = 0.042$

$$r2 := \frac{sse_{old} - sse}{sse_{old}} \quad r2 = 99.989\%$$

$$r := \sqrt{r2} \quad r = 99.995\%$$

Almost all variations in y have been captured at this point.

3rd Iteration.

Covariance matrix: $W := E^T \cdot F \cdot F^T \cdot E$

$$W = \begin{bmatrix} 1.508 \cdot 10^{-10} & -1.508 \cdot 10^{-10} & -3.103 \cdot 10^{-14} \\ -1.508 \cdot 10^{-10} & 1.508 \cdot 10^{-10} & 3.103 \cdot 10^{-14} \\ -3.103 \cdot 10^{-14} & 3.103 \cdot 10^{-14} & 0 \end{bmatrix}$$

Eigenvalue: $\lambda := \text{reverse}(\text{sort}(\text{eigenvals}(W)))$ $\lambda^T = (3.016 \cdot 10^{-10} \quad 0 \quad 0)$
 ↑ No nonzero eigenvalue. There is really no more relationship between E and F that we can milk!

Eigenvector: $w^{<2>} := \text{eigenvec}(W, \lambda_0)$

$$w = \begin{bmatrix} 0.342 & 0.634 & -0.707 \\ 0.342 & 0.634 & 0.707 \\ 0.978 & -0.444 & 1.455 \cdot 10^{-4} \end{bmatrix}$$

check. These eigenvectors should be orthogonal, but they are not quite so here because of a verysmall degree of numerical error. At this point, the residuals represent just noise, and the numbers we calculate for the the last eigenvector are highly unstable. A small amount of noise might knock the eigenvector around if additional degree of freedom existed, say if the column dimension of X were higher than 3.

$$w^T \cdot w = \begin{bmatrix} 1.19 & 0 & -1.709 \cdot 10^{-13} \\ 0 & 1 & -3.052 \cdot 10^{-13} \\ -1.709 \cdot 10^{-13} & -3.052 \cdot 10^{-13} & 1 \end{bmatrix}$$

Let us manually create the last orthogonal vector via the Gram-Schmidt orthogonalization process.

Start with: $w_{start} := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

Gram-Schmidt process: $w^{<2>} := w_{start} - \frac{w^{<0>} \cdot w_{start}}{w^{<0>} \cdot w^{<0>}} \cdot w^{<0>} - \frac{w^{<1>} \cdot w_{start}}{w^{<1>} \cdot w^{<1>}} \cdot w^{<1>}$

Normalize: $w^{<2>} := \frac{w^{<2>}}{|w^{<2>}|}$ $w^{<2>} = \begin{bmatrix} -0.707 \\ 0.707 \\ 1.455 \cdot 10^{-4} \end{bmatrix}$

Check once more to make sure all w vectors are orthogonal.

$$w^T \cdot w = \begin{pmatrix} 1.19 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We can toggle on/off the next equation and examine what happens to the rest of the steps. We see that there is not much residual left after the last factor has been taken out in either case. So, we revert back to the original eigenvector.

$w^{<2>} := \text{eigenvec}(W, \lambda_0)$

score (column) vector for X: $t^{<2>} := E \cdot w^{<2>}$

loading (row) vector for Y: $q^{<2>} := F^T \cdot t^{<2>}$ normalize: $q^{<2>} := \frac{q^{<2>}}{|q^{<2>}|}$ $q^{<2>} = 1$

score (column) vector for Y: $u^{<2>} := F \cdot q^{<2>}$

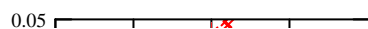
loading (row) vector for X: $p^{<2>} := E^T \cdot t^{<2>}$ normalize: $p^{<2>} := \frac{p^{<2>}}{|p^{<2>}|}$ $p^{<2>} = \begin{bmatrix} -0.707 \\ 0.707 \\ 1.455 \cdot 10^{-4} \end{bmatrix}$

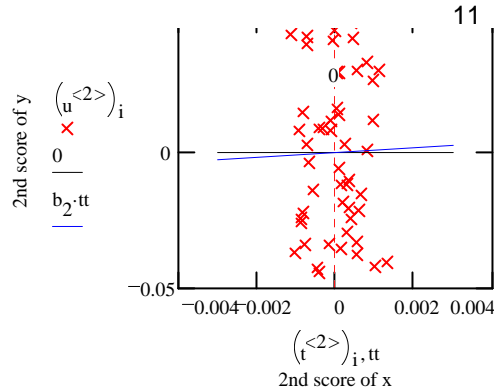
The correction step is not necessary because the correction factor is unity.

$$\frac{1}{w^{<2>} \cdot p^{<2>}} = 1$$

Regression: $b_2 := \text{slope}(t^{<2>}, u^{<2>})$ $b = \begin{pmatrix} 1.798 \\ 1.204 \\ 0.876 \end{pmatrix}$

Check: $\text{intercept}(t^{<2>}, u^{<2>}) = 0$ $tt := -0.003, -0.0029 .. 0.003$





Residual matrices. (Theoretically, after all m components have been taken out, where m is the number of independent variables in the original X , the residual of X should be exactly 0. That is to say that when the number of factors equal the rank of X , all the variation in X should be accounted for. However, this does not mean that we will capture all variations in Y through regression. For example, if Y depends on the product of $x^{<0>}x^{<1>}$, then we must expand the X matrix to include the cross term in order to capture as much variation in Y as possible. Of course, when all N components have been taken out, where N is the number of data points, the residuals in Y should be exactly zero. In other words, with N terms, we can fit any given N points, even an elephant.)

$$E := E - t^{<2>} \cdot p^{<2>T}$$

$$F := F - t^{<2>} \cdot q^{<2>T} \cdot b_2$$

Check: $\sum_{j=0}^2 (E^T \cdot E)_{j,j} = 0 \leftarrow \text{should be exactly 0.}$

Goodness of fit: $sse := F \cdot F$ $sse = 0.042 \leftarrow \text{not quite exactly zero.}$

$$r2 := \frac{sse_{old} - sse}{sse_{old}} \quad r2 = 99.989\%$$

$$r := \sqrt{r2} \quad r = 99.995\%$$

Note that we should not hope to achieve $sse=0$ because we do not want to capture the noise in Y . At the beginning of this worksheet, we generated the noise in Y with a random number generator (rnd). As a double check, we can expect the level of residual sse to reflect the following noise level.

$$\text{noise}_i := \text{rnd}(0.1) - 0.05 \quad \text{noise} \cdot \text{noise} = 0.052$$

$$(Y_{\text{save}} - y(X_{\text{save}})) \cdot (Y_{\text{save}} - y(X_{\text{save}})) = 0.046 \quad \text{The two numbers on the left should be similar.}$$

Step 6. Regression Model. (Be sure to take care of both mean-centering and variance-scaling.)

Since there is not much improvement in going from two to three factors, stop at the end of the 2nd iteration.

Although we do not really have to include q here because it is unity when there is only one dependent variable y , we include it anyway so that we can handle the possibility of multiple dependent variables.

$$q = (1 \ -1 \ 1)$$

Variance-scaling: $x_{\text{stdev_inv},j,j} := \frac{1}{x_{\text{stdev}_{0,j}}}$

The next equation is simple-minded and incorrect because $w^{<0>}$ is not normalized, although w vectors are mutually orthogonal. Furthermore, $t^{<1>} = E \cdot w^{<1>} \neq X \cdot w^{<1>}$, etc.

$$y_{\text{regress}}(x) := \sum_{j=0}^1 \left[(x - x_{\text{mean}}) \cdot x_{\text{stdev_inv}} \cdot w^{<j>} \right] \cdot q^{<j>T} \cdot b_j + y_{\text{mean}}$$

Examples:

$$y_{\text{regress}}((5 \ 0.5 \ 0.05)) = 3.632 \longleftrightarrow y((5 \ 0.5 \ 0.05)) = 5.05 \leftarrow \text{No good.}$$

$$y_{\text{regress}}((5 \ -0.5 \ 0.05)) = 4.158 \longleftrightarrow y((5 \ -0.5 \ 0.05)) = 4.95 \leftarrow \text{No good.}$$

Step-by-step reconstruction of the dependent variable y from independent variable x (since the final model expression is somewhat complicated). y is the sum of the individual features captured by each regression factor.

$$y_{\text{regress}} = \sum_{j=0}^1 t^{<j>} \cdot q^{<j>T} \cdot b_j$$

where $t^{<j>} = E \cdot w^{<j>}$

Thus, we have:

$$y_{\text{regress}} = \sum_{j=0}^1 E \cdot w^{<j>} \cdot q^{<j>T} \cdot b_j$$

Note that E , which is the residual matrix of X , is not constant but changes as we include each new additional factor.

$$E = X \quad \text{for } j=0$$

$$E = X - t^{<0>} \cdot p^{<0>T} = X - (X \cdot w^{<0>}) \cdot p^{<0>T} = X \cdot (I - w^{<0>} \cdot p^{<0>T}) \quad \text{for } j=1$$

Substituting these E into y_{regress} , we have:

$$\begin{aligned} y_{\text{regress}} &= X \cdot w^{<0>} \cdot q^{<0>T} \cdot b_0 + E \cdot w^{<1>} \cdot q^{<1>T} \cdot b_1 \\ &= X \cdot w^{<0>} \cdot q^{<0>T} \cdot b_0 + \left[X \cdot (I - w^{<0>} \cdot p^{<0>T}) \right] \cdot w^{<1>} \cdot q^{<1>T} \cdot b_1 \\ &= X \cdot \left[w^{<0>} \cdot q^{<0>T} \cdot b_0 + (I - w^{<0>} \cdot p^{<0>T}) \cdot w^{<1>} \cdot q^{<1>T} \cdot b_1 \right] \end{aligned}$$

The variables y_{regress} and X in the above equation have been mean-centered and perhaps variance-scaled. We need to take care of these steps in the final regression equation. Since the recursive relationship becomes fairly complicated as the number of factors increases, we really ought to use the programming feature in version 6.0. Nevertheless, here it is for two factors:

$I_{j,j} := 1 \quad \dots$ an identity matrix.

$$y_{\text{regress}}(x) := (x - x_{\text{mean}}) \cdot x_{\text{stdev_inv}} \cdot \left[w^{<0>} \cdot q^{<0>T} \cdot b_0 + (I - w^{<0>} \cdot p^{<0>T}) \cdot w^{<1>} \cdot q^{<1>T} \cdot b_1 \right] \dots$$

+ y mean

Let us examine the slope and intercept with 2 terms (j=0,1).

Intercept:

$$y_{\text{mean}} - x_{\text{mean}} \cdot x_{\text{stdev_inv}} \cdot \left[w^{<0>} \cdot q^{<0>T} \cdot b_0 + \left(I - w^{<0>} \cdot p^{<0>T} \right) \cdot w^{<1>} \cdot q^{<1>T} \cdot b_1 \right] = -0.005$$

which is practically 0; thus, y=x-slope

slope:

$$x_{\text{stdev_inv}} \cdot \left[w^{<0>} \cdot q^{<0>T} \cdot b_0 + \left(I - w^{<0>} \cdot p^{<0>T} \right) \cdot w^{<1>} \cdot q^{<1>T} \cdot b_1 \right] = \begin{pmatrix} 0.006 \\ 0.06 \\ 99.923 \end{pmatrix}$$

← compare → $y(x) := x \cdot \begin{pmatrix} 0 \\ 0.1 \\ 100 \end{pmatrix}$

Examples:

$$y_{\text{regress}}((5 \ 0.5 \ 0.05)) = 5.052 \leftarrow \rightarrow y((5 \ 0.5 \ 0.05)) = 5.05 \leftarrow \text{O.K.}$$

$$y_{\text{regress}}((5 \ -0.5 \ 0.05)) = 4.992 \leftarrow \rightarrow y((5 \ -0.5 \ 0.05)) = 4.95 \leftarrow \text{O.K. only because } y \text{ is not a function of } x^{<1>}$$

Miscellaneous Stuff.

Check: orthogonality of p. (The vectors in p are not mutually orthogonal.)

$$p^T \cdot p = \begin{bmatrix} 1 & 0.4 & 3.399 \cdot 10^{-14} \\ 0.4 & 1 & 2.522 \cdot 10^{-7} \\ 3.399 \cdot 10^{-14} & 2.522 \cdot 10^{-7} & 1 \end{bmatrix}$$

Check: orthogonality of w. (The vectors in w are mutually orthogonal, but the 0th vector is not normalized.)

$$w^T \cdot w = \begin{bmatrix} 1.19 & 0 & -1.709 \cdot 10^{-13} \\ 0 & 1 & -3.052 \cdot 10^{-13} \\ -1.709 \cdot 10^{-13} & -3.052 \cdot 10^{-13} & 1 \end{bmatrix}$$

Check: X=t·p^T?

	0	1
0	-1.659	-1.661
1	-0.419	-0.418
2	0.861	0.861
3	-1.14	-1.139
4	-1.632	-1.632

	0	1
0	-1.659	-1.661
1	-0.419	-0.418
2	0.861	0.861
3	-1.14	-1.139
4	-1.632	-1.632

Check: Y≠u·q^T? (Y=u^{<0>}·q^{<0>T}, Y=Σ(t·q^T·b)

	0
0	0.45
1	2.612

	0
0	0.45
1	2.612

1

	0
0	0.472
1	2.501

$$Y = \begin{bmatrix} 1 & -3.612 \\ 2 & -4.334 \\ 3 & -4.139 \\ 4 & 0.657 \\ 5 & -4.774 \end{bmatrix}$$

$$u_{<0>} \cdot q_{<0>}^T = \begin{bmatrix} 1 & -3.612 \\ 2 & -4.334 \\ 3 & -4.139 \\ 4 & 0.657 \\ 5 & -4.774 \end{bmatrix}$$

$$\sum_{j=0}^{14} t_{<j>} \cdot q_{<j>}^T \cdot b_j = \begin{bmatrix} 1 & -3.571 \\ 2 & -4.373 \\ 3 & -4.166 \\ 4 & 0.64 \\ 5 & -4.783 \end{bmatrix}$$

The 0th column of u is Y and the last column of u is F before subtracting the last factor.

$$u = \begin{bmatrix} & 0 & 1 \\ 0 & 0.45 & -2.156 \\ 1 & -3.612 & 0.822 \\ 2 & -4.334 & 2.593 \\ 3 & -4.139 & 0.111 \\ 4 & 0.657 & -2.223 \end{bmatrix}$$

$$F = \begin{bmatrix} & 0 \\ 0 & -0.022 \\ 1 & -0.022 \\ 2 & 0.04 \\ 3 & 0.026 \\ 4 & 0.016 \\ 5 & 0.009 \end{bmatrix}$$

Save the data file on disk.

```
PRNPRECISION := 10    PRNCOLWIDTH := 17    WRITEPRN(pca_dat) := augment(Y, X)
```