Chapter 13
Genomic Selection using Partial Least Squares

1. Introduction

The partial least squares (PLS) regression method fits models by using any one of a number of linear predictive methods. Ordinary least squares regression, as implemented in SAS/STAT procedures such as PROC GLM and PROC REG, has the single goal of minimizing sample response prediction error, seeking linear functions of the predictors that explain as much variation in each response as possible. The techniques implemented in the PLS procedure have the additional goal of accounting for variation in the predictors, under the assumption that directions in the predictor space that are well sampled should provide better prediction for *new* observations when the predictors are highly correlated. All of the techniques implemented in the PLS procedure work by extracting successive linear combinations of the predictors, called *factors* (also called *components*, *latent vectors*, or *latent variables*), which optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking factors that explain both response and predictor variation. The PROC PLS in SAS is particularly designed to perform PLS analysis for high dimensional data where the number of independent variables can be substantially larger than the sample size.

Note that the name "partial least squares" also applies to a more general statistical method that is *not* implemented in this procedure. The partial least squares method was originally developed in the 1960s by an econometrician named Herman Wold (1966) for modeling "paths" of causal relation between any number of "blocks" of variables. However, the PLS procedure fits only *predictive* partial least squares models, with one "block" of predictors and one "block" of responses.

All of the predictive methods implemented in PROC PLS work essentially by finding linear combinations of the predictors (factors) to use to predict the responses linearly. The methods differ only in how the factors are derived, as explained in the following sections.

2. Theory of PLS

2.1. Nonlinear Iterative Partial Least Squares (NIPALS)

Let $X_{n\times m}$ be an $n\times m$ matrix of predictors and $Y_{n\times s}$ be an $n\times s$ response variables. The overall purpose of PLS is to predict *Y* using *X*. Let us denote $X_1 = X$ as the standardized *X* matrix (centered and scaled) and $Y_1 = Y$ as the standardized *Y* matrix (centered and scaled). Note that X_1 and Y_1 are not the first columns of the corresponding matrices *X* and *Y*. Since there are *m* variables in X_1, we need to find a linear combination of the *m* variables called T_1 (an $n \times 1$ vector). We then find a linear combination of the *s* variables in Y_1 called U_1 (an $n \times 1$ vector). The T_1 and U_1 are called the first *X* score and the first *Y* score, respectively, and these scores are defined as

$$\begin{aligned} T_1 &= X_1 W_1 \\ U_1 &= Y_1 Q_1 \end{aligned} \tag{1}$$

Note that W_1 is an $m \times 1$ vector of weights for *X* and Q_1 is an $s \times 1$ vector of weights for *Y*. The two weight vectors (W_1 and Q_1) are found in such a way that they maximize the squared covariance between T_1 and Q_1. Mathematically, they are expressed as

$$(W_1, Q_1) = \arg \max_{(W_1, Q_1) \in \Omega} \left[\operatorname{cov}^2(T_1, U_1) \right] \tag{2}$$

The solution for W_1 is the first eigenvector of matrix $X_1^T Y_1 Y_1^T X_1$ and the solution for Q_1 is the first eigenvector of matrix $Y_1^T X_1 X_1^T Y_1$. Once we find these two vectors, we then perform regression analyses for X_1 and Y_1 using T_1 as the predictor (independent variable)

$$\begin{aligned} X_1 &= T_1 P_1 + E \\ Y_1 &= T_1 C_1 + F \end{aligned} \tag{3}$$

which leads to

$$\begin{aligned} P_1 &= (T_1^T T_1)^{-1} T_1^T X_1 \\ C_1 &= (T_1^T T_1)^{-1} T_1^T Y_1 \end{aligned} \tag{4}$$

We then predict X_1 and Y_1 using

$$\begin{aligned} \hat{X}_1 &= T_1 P_1 \\ \hat{Y}_1 &= T_1 C_1 \end{aligned} \tag{5}$$

This accounts for how the first PLS factor is extracted. The second factor is extracted in the same way by replacing X_1 and Y_1 with the *X* residuals and the *Y* residuals from the first factor, i.e.,

$$\begin{aligned} X_2 &= X_1 - \hat{X}_1 = X_1 - T_1 P_1 \\ Y_2 &= Y_1 - \hat{Y}_1 = Y_1 - T_1 C_1 \end{aligned} \tag{6}$$

These residuals are called the deflated *X* and *Y* blocks. The process of extracting a score vector and deflating the data matrices is repeated for as many extracted factors as are wanted. Let $f \le m$ be the number of factors and f is often smaller than *m*. We will discuss how to determine f later.

Once we finish all the extraction and deflation process, we place all the *T* vectors together to form a $T = \left[T_1 \,||\, T_2 \,||\, \cdots \,||\, T_f \right]$ matrix. The dimension of matrix *T* is $n \times f$ and this matrix is called the *X* score matrix. We define $P = \left[P_1 \,/\,/ P_2 \,/\,/ \cdots /\,/ P_f \right]$ as an $f \times m$ matrix called the *X* loadings and $C = \left[C_1 \,/\,/ C_2 \,/\,/ \cdots /\,/ C_f \right]$ as an $f \times s$ matrix called the *Y* loadings. Similarly, $W = \left[W_1 \,||\, W_2 \,||\, \cdots \,||\, W_f\right]^T$ is the weight matrix (*f* rows and *m* columns) for *X* and $Q = \left[Q_1 \,||\, Q_2 \,||\, \cdots \,||\, Q_f\right]^T$ is the weight matrix (*f* rows and *s* columns) for *Y*. Finally, the prediction model for all *Y* is

$$\hat{Y} = TC = T(T^T T)^{-1} T^T Y = XW^T (WX^T XW^T)^{-1} WX^T Y = XB \tag{7}$$

where

$$B = W^T (WX^T XW^T)^{-1} WX^T Y \tag{8}$$

is the multiple regression coefficients of *Y* on *X* from the PLS analysis. This method of fitting the PLS model is called the Nonlinear Iterative Partial Least Squares (NIPALS). In PROC PLS, the `method = PLS` option allows you to perform PLS analysis. This is the default method.

2.2. Statistically Inspired Modification of PLS (SIMPLS)

Note that each extracted PLS factor is defined in terms of different *X*-variables X_i. This leads to difficulties in comparing different scores, weights, and so forth. There is another method called Statistically Inspired Modification of PLS (SIMPLS). This method was developed by de Jong (1993) and it overcomes these difficulties by computing each score $T_i = XW_i$ in terms of the original (centered and scaled) predictors *X*. Similarly, $U_i = YQ_i$ is the score of the response variables in terms of the original *Y*. The first SIMPLS *X*-weight vector W_1 and the first SIMPLS *Y*-weight vector Q_1 are found by maximizing the squared covariance between T_1 and U_1. The solution for T_1 happens to be the first eigenvector of matrix $X^T Y Y^T X$ and the solution for Q_1 is the first eigenvector of matrix $Y^T X \; X^T Y$. Therefore, the first SIMPLS scores are the same as the first PLS scores. The second scores $T_2 = XW_2$ and $U_2 = YQ_2$ are found by

$$(W_2, Q_2) = \arg\max_{(W_2, Q_2) \in \Omega} \left[\text{cov}^2(T_2, U_2)\right] \tag{9}$$

subject to

$$\begin{aligned} W_2^T X^T XW_2 &= Q_2^T Y^T YQ_2 = 1 \\ W_1^T X^T XW_2 &= Q_1^T Y^T YQ_2 = 0 \end{aligned} \tag{10}$$

The third scores $T_3 = XW_3$ and $U_3 = YQ_3$ are found by

$$(W_3, Q_3) = \arg\max_{(W_3, Q_3) \in \Omega} \left[\text{cov}^2(T_3, U_3)\right] \tag{11}$$

subject to

$$\begin{gathered} W_3^T X^T XW_3 = Q_3^T Y^T YQ_3 = 1 \\ W_1^T X^T XW_3 = W_2^T X^T XW_3 = Q_1^T Y^T YQ_3 = Q_2^T Y^T YQ_3 = 0 \end{gathered} \tag{12}$$

This process continues until we extract the desired number of factors *f*. In the end, we have

$$\hat{Y} = XW^T (WX^T XW^T)^{-1} WX^T Y = XB \tag{13}$$

where

$$B = W^T (WX^T XW^T)^{-1} WX^T Y \tag{14}$$

You can see that SIMPLS and NIPALS differ only by the way that the W matrix is drawn. When there is only one response variable, $s = 1$, the two methods are identical.

3. Cross validation

None of the regression methods implemented in the PLS procedure fit the observed data any better than ordinary least squares (OLS) regression; in fact, all of the methods approach OLS as more factors are extracted. The crucial point is that, when there are many predictors, OLS can *overfit* the observed data; biased regression methods with fewer extracted factors can provide better predictability of *future* observations. However, as the preceding observations imply, the quality of the observed data fit cannot be used to choose the number of factors to extract; the number of extracted factors must be chosen on the basis of how well the model fits observations not involved in the modeling procedure itself.

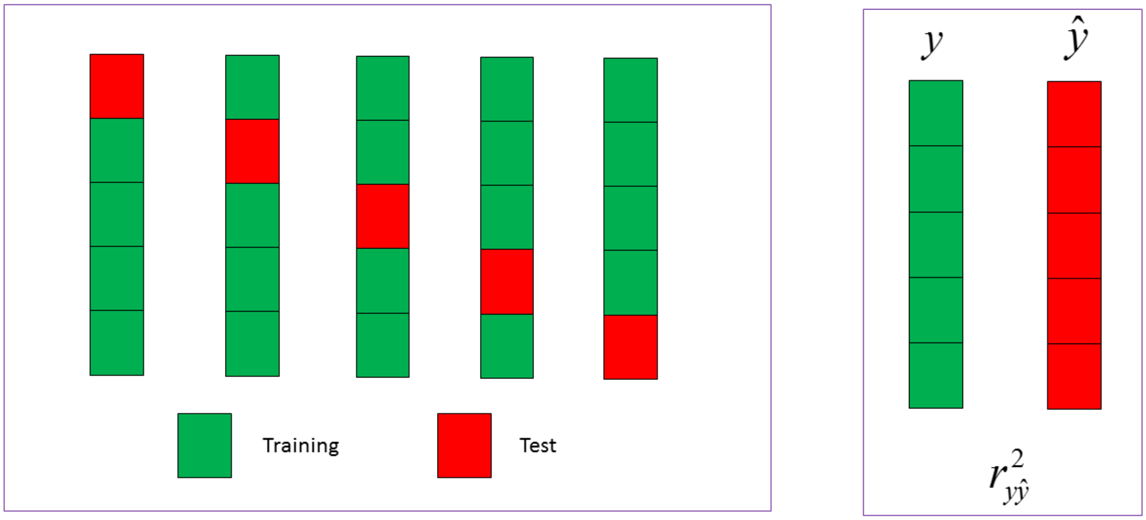
One method of choosing the number of extracted factors is to fit the model to only part of the available data (the *training set*) and to measure how well models with different numbers of extracted factors fit the other part of the data (the *test set*). This is called *test set validation*. However, it is rare that you have enough data to make both parts large enough for pure test set validation to be useful. Alternatively, you can make several different divisions of the observed data into training set and test set. This is called *cross validation*, and there are several different types. In *one-at-a-time* cross validation, the first observation is held out as a single-element test set, with all other observations as the training set; next, the second observation is held out, then the third, and so on. Another method is to hold out successive blocks of observations as test sets—for example, observations 1 through 7, then observations 8 through 14, and so on; this is known as *blocked* validation. A similar method is *split-sample* cross validation, in which successive groups of widely separated observations are held out as the test set—for example, observations {1, 11, 21, …}, then observations {2, 12, 22, …}, and so on. Finally, test sets can be selected from the observed data randomly; this is known as *random sample* cross validation.

Which validation you should use depends on your data. Test set validation is preferred when you have enough data to make a division into a sizable training set and test set that represent the predictive population well. You can specify that the number of extracted factors be selected by test set validation by using the CV= TESTSET(*data set*) option in PROC PLS of SAS, where *data set* is the name of the data set containing the test set. If you do not have enough data for test set validation, you can use one of the cross validation techniques. The most common technique is one-at-a-time validation (which you can specify with the CV= ONE option or just the CV option), unless the observed data are serially correlated, in which case either blocked or split-sample validation might be more appropriate (CV= BLOCK or CV= SPLIT); you can specify the number of test sets in blocked or split-sample validation with a number in parentheses after the CV= option. Note that CV= ONE is the most computationally intensive of the cross validation methods, since it requires a recomputation of the PLS model for every input observation. Also, note that using random subset selection with CV= RANDOM might lead two different researchers to produce different PLS models on the same data (unless the same seed is used).

Whichever validation method you use, the number of factors chosen is usually the one that minimizes the predicted residual sum of squares (PRESS); this is the default choice if you specify any of the CV methods with PROC PLS.

The most commonly adopted CV method is the K-fold cross validation. You can set the CV = SPLIT(K) option in the PROC PLS statement. For example, when K = 10, the method is called the 10-fold cross validation. **Figure 1** illustrates a 5-fold cross validation and how to calculate the model predictability.

Figure 1
Sketch of a K-fold cross validation



4. PROC PLS in SAS

The PLS procedure in SAS provides partial least squares regression analysis. In addition, PROC PLS can also perform principal component regression and several other regression methods that can handle a very large number of independent variables.

We now use a sample data to demonstrate PROC PLS. The example in this section illustrates basic features of the PLS procedure. The data are reported in Umetrics (1995); the original source is Lindberg, Persson, and Wold (1983). Suppose that you are researching pollution in the Baltic Sea, and you would like to use the spectra of samples of seawater to determine the amounts of three compounds present in samples from the Baltic Sea: lignin sulfonate (*ls*: pulp industry pollution), humic acids (*ha*: natural forest products), and optical whitener from detergent (*dt*). Spectrometric calibration is a type of problem in which partial least squares can be very effective. The predictors are the spectra emission intensities at different frequencies in sample spectrum, and the responses are the amounts of various chemicals in the sample.

For the purposes of calibrating the model, samples with known compositions are used. The calibration data consist of 16 samples of known concentrations of *ls*, *ha*, and *dt*, with spectra based on 27 frequencies (or, equivalently, wavelengths). Part of the data are shown in **Table 1**. The following statements read the data (pollution.xlsx) and perform PLS analysis.

Table 1

The pollution data with 27 predictors and three response variables

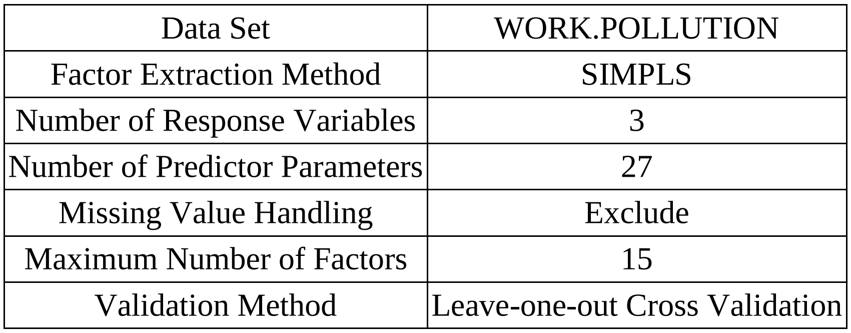
obs	v1	v2	…	v27	ls	ha	dt
EM1	2766	2610	…	1017	3.011	0	0
EM2	1492	1419	…	50	0	0.4005	0
EM3	2450	2379	…	50	0	0	90.63
EM4	2751	2883	…	582	1.482	0.158	40
EM5	2652	2691	…	507	1.116	0.4104	30.45
EM6	3993	4722	…	1227	3.397	0.3032	50.82
EM7	4032	4350	…	957	2.428	0.2981	70.59
EM8	4530	5190	…	1380	4.024	0.1153	89.39
EM9	4077	4410	…	963	2.275	0.504	81.75
EM10	3450	3432	…	468	0.9588	0.145	101.1
EM11	4989	5301	…	1167	3.19	0.253	120
EM12	5340	5790	…	1470	4.132	0.5691	117.7
EM13	3162	3477	…	855	2.16	0.436	27.59
EM14	4380	4695	…	1119	3.094	0.2471	61.71
EM15	4587	4200	…	714	1.604	0.2856	108.8
EM16	4017	4725	…	1257	3.162	0.7012	60

```
%let dir=C:\Users\SHXU\STAT-231B-2016\Text\Chapter 18;
proc import out=pollution datafile="&dir\pollution.xlsx" dbms=xlsx
replace;
run;
proc pls data=pollution method=simpls cv=one details;
   model ls ha dt = v1-v27/solution;
   output out=pred p=p_ls p_ha p_dt;
   ods output CenScaleParms=b XWeights =weight;
run;
```

PROC PLS first reports the information about the data and the procedure (shown in **Table 2**)

Table 2

Data and procedure information



Data Set	WORK.POLLUTION
Factor Extraction Method	SIMPLS
Number of Response Variables	3
Number of Predictor Parameters	27
Missing Value Handling	Exclude
Maximum Number of Factors	15
Validation Method	Leave-one-out Cross Validation

The most important information is the PRESS value profile (amount of change as the number of extracted factors increases). Zero extracted number of factors represents the PRESS of the null model. The CV = ONE option allows the leave-one-out cross validation and generates the following table.

Table 3
Root Mean PRESS

Cross Validation for the Number of Extracted Factors	
Number of Extracted Factors	Root Mean PRESS
0	1.066667
1	0.892851
2	0.824549
3	0.592936
4	0.528524
5	0.658519
6	0.479886
7	0.428608
8	0.477809
9	0.472736
10	0.481212
11	0.480967
12	0.523949
13	0.530066
14	0.531553
15	0.531553

The smallest root mean PRESS occurs when the number of factors is 7, which is 0.428608. As the number of factors increases after 7, the PRESS value starts to increase. The PRESS value at 7 factors is $\text{Press}(7) = 0.428608^2 = 0.183705$. The estimated residual error sum of squares is $\text{PRESS}(0)=1.066667^2 = 1.137778$. Therefore, the predictability is

$$R^2 = 1 - \frac{\text{PRESS}(7)}{\text{PRESS}(0)} = 1 - \frac{0.183705}{1.137778} = 0.838541 \qquad (15)$$

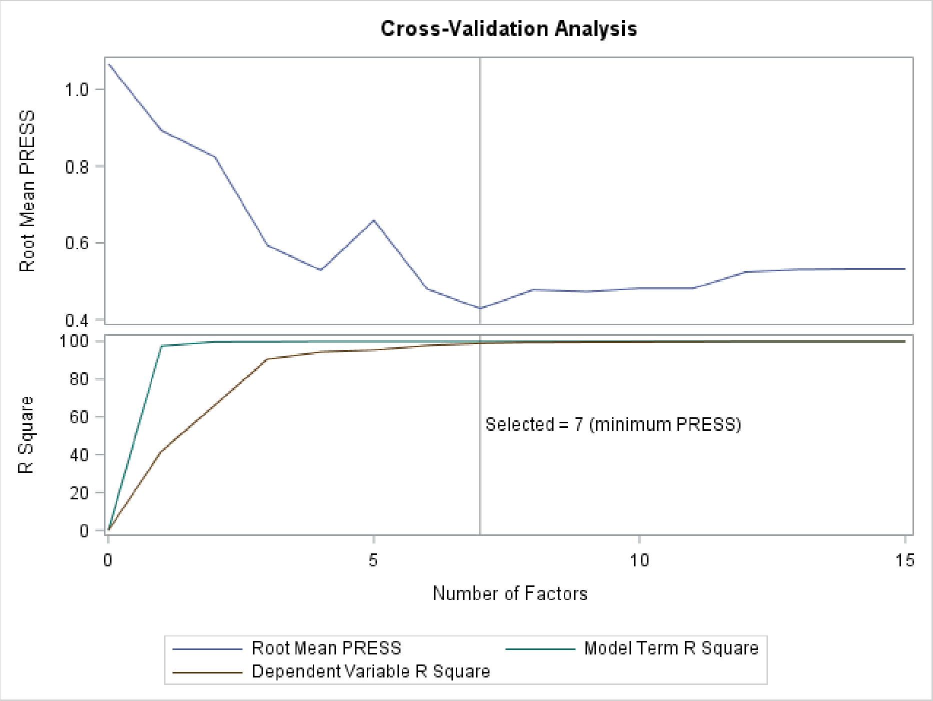
The next result is the percent variation accounted for the SIMPLS factors (**Table 4**).

Table 4
Percent variation accounted for by SIMPLS factors

Percent Variation Accounted for by SIMPLS Factors				
Number of Extracted Factors	Model Effects		Dependent Variables	
	Current	Total	Current	Total
1	97.4607	97.4607	41.9155	41.9155
2	2.1830	99.6436	24.2438	66.1592
3	0.1779	99.8216	24.5573	90.7166
4	0.1198	99.9413	3.7695	94.4861
5	0.0416	99.9829	0.9906	95.4767
6	0.0105	99.9935	2.2984	97.7751
7	0.0017	99.9952	1.1584	98.9335

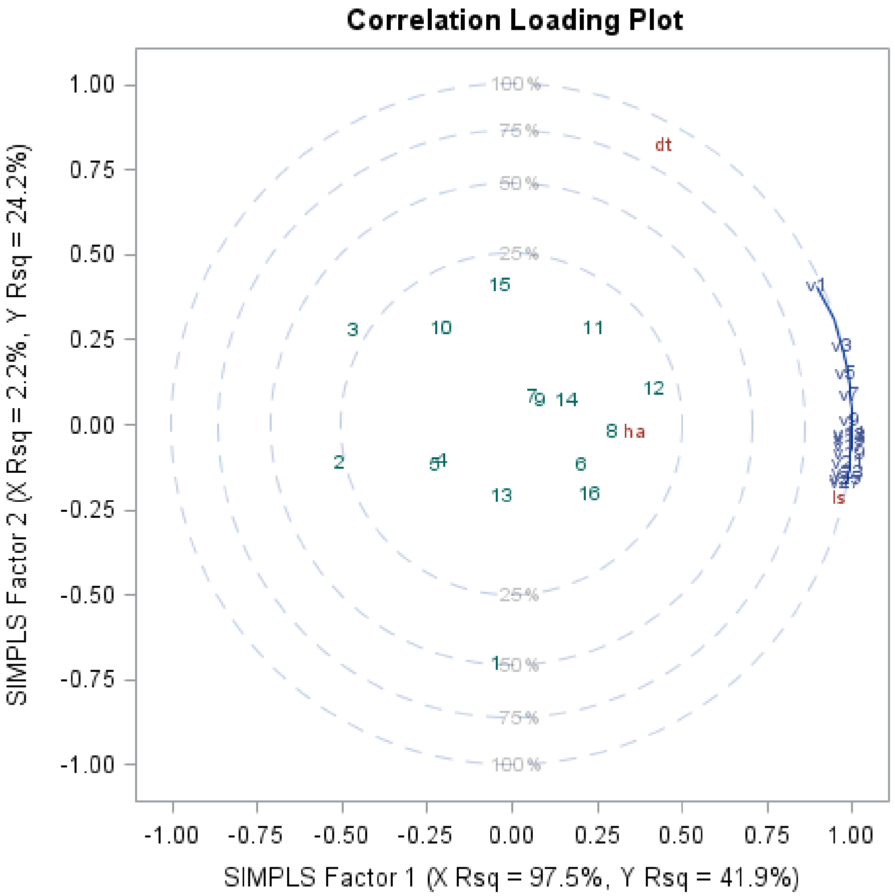
This table shows that the first factor contributes 97.4% of variation for the 27 predictors but it only contributes 41.9% of the variation for the three response variables. The first two factors contribute 99.64% of variation of the predictors and 66.16% of variation for the response variables.

Figure 2
Cross validation plot of the pollution data PLS analysis



The cross validation plot in **Figure 2** gives a visual representation of the selection of the optimum number of factors discussed previously. PROC PLS also generates a correlation loading plot (**Figure 3**).

Figure 3
Correlation loading plot of the pollution data



The correlation loading plot is a compact summary of many features of the PLS model. For example, it shows that the first factor is highly positively correlated with all spectral values, indicating that it is approximately an average of them all; the second factor is positively correlated with the lowest frequencies and negatively correlated with the highest, indicating that it is approximately a contrast between the two ends of the spectrum. The observations, represented by their number in the data set on this plot, are generally spaced well apart, indicating that the data give good information about these first two factors.

Finally, the solution option in the model statement prints the estimated regression coefficients (the *B* matrix in $\hat{Y} = XB$) as shown in **Table 5** below. This matrix allows you to predict future observations where only *X* are available without *Y*.

Table 5
Estimated regression coefficients for 27 predictors and three response variables using seven extracted SIMPLS factors of the pollution data

Parameter Estimates			
	ls	ha	dt
Intercept	0.69785733	-0.08548999	-75.11263261
v1	-0.00010975	-0.00012225	0.01169590
v2	-0.00045520	-0.00002487	0.12082541
v3	-0.00018413	0.00010409	-0.04110417
v4	-0.00000601	0.00004259	-0.04730823
v5	0.00022086	-0.00010188	-0.01337337
v6	0.00038140	-0.00023437	0.04111064
v7	0.00030258	-0.00015594	0.02154072
v8	0.00026710	-0.00012388	0.03405598
v9	-0.00005032	0.00018954	-0.05697615
v10	-0.00017089	0.00020494	-0.00809670
v11	-0.00031866	0.00031922	-0.03323955
v12	-0.00043072	0.00034991	-0.01333096
v13	-0.00035335	0.00030847	-0.00683950
v14	-0.00022832	0.00021127	0.00496476
v15	-0.00032272	0.00030037	-0.03277549
v16	0.00019731	-0.00008702	0.07366920
v17	0.00013200	0.00007295	-0.02756192
v18	0.00019277	0.00001229	-0.00557267
v19	0.00057828	-0.00040427	0.15926784
v20	0.00068204	-0.00047475	0.14563307
v21	0.00015440	0.00003179	-0.09496604
v22	0.00034873	-0.00030877	0.06273182
v23	-0.00032749	0.00053233	-0.32511748
v24	0.00000810	-0.00002570	-0.07177708
v25	0.00002099	0.00002960	-0.22452086
v26	0.00118963	-0.00120612	0.32020365
v27	0.00052198	-0.00052672	0.00834452

Suppose that we have two new observed data points (**Table 6**). We can predict the three response variables for the two new observations.

Table 6
New observations of the 27 predictors

obs	v1	v2	…	v27
EM17	3933	4518	…	987
EM25	2904	2997	…	162

The new observations are stored in a file called “pollutNew.xlsx”. The following code will predict the three response variables for the two new observations.

```
proc import out=pollutNew datafile="&dir\pollutNew.xlsx" dbms=xlsx
replace;
run;

data all;
   set pollution pollutNew;
run;

proc pls data=all method=simpls cv=one details;;
   model ls ha dt = v1-v27;
   output out=prednew p=p_ls p_ha p_dt;
run;

proc export data=prednew outfile="&dir\predNew.xlsx" dbms=xlsx
replace;
run;
```

We combined the new data with the existing data. Because the new data do not have the response variables measured, they have missing values for the two new observations. We essentially reanalyzed the data with missing values. PROC PLS predicts the response variables regardless an individual observation has missing response variables or not. The predicted responses are listed in **Table 7** (next page) with the new observations highlighted.

Table 7

Observed and predicted response variables of the pollution data

obs	ls	ha	dt	pred_ls	pred_ha	pred_dt
EM1	3.011	0	0	2.983375216	0.0190381	-0.417914665
EM2	0	0.4005	0	-0.046012	0.391874851	0.541201924
EM3	0	0	90.63	-0.100535497	-0.0025125	90.17284932
EM4	1.482	0.158	40	1.528255834	0.118175593	35.41604381
EM5	1.116	0.4104	30.45	1.175089216	0.439465353	28.11291717
EM6	3.397	0.3032	50.82	3.37608275	0.341224015	49.15671604
EM7	2.428	0.2981	70.59	2.409791836	0.290276008	74.05968167
EM8	4.024	0.1153	89.39	4.088649946	0.093368214	87.96712473
EM9	2.275	0.504	81.75	2.28923279	0.547291322	75.73210704
EM10	0.9588	0.145	101.1	1.050024847	0.159957441	103.0432997
EM11	3.19	0.253	120	3.18300893	0.305577231	121.9067656
EM12	4.132	0.5691	117.7	4.06923713	0.542034711	116.115343
EM13	2.16	0.436	27.59	2.222051379	0.424589067	35.86399612
EM14	3.094	0.2471	61.71	3.025269401	0.241196761	63.91438192
EM15	1.604	0.2856	108.8	1.633999492	0.254733005	108.3686032
EM16	3.162	0.7012	60	3.146278944	0.660210755	60.5768834
EM17	.	.	.	2.501968346	0.309819401	80.70113402
EM18	.	.	.	-0.447358605	1.492702588	67.27107004

5. Genomic prediction in rice

The purpose of this study is to predict hybrid yields in rice using parental genotypes. The rice (*Oryza sativa*) population was derived from the cross between Zhenshan 97 and Minghui 63 (Hua et al. 2002; Xing et al. 2002), the parents of Shanyou 63, the most widely grown hybrid. There are two populations of rice and we only analyzed the second population. The first population consisted of 210 recombinant inbred lines (RIL) derived by single-seed descent from the cross between the two parents. The second population consisted of 278 crosses randomly paired among the 210 recombinant inbred lines. This population is called the immortalized F_2 (IMF2) population (Hua et al. 2002; Hua et al. 2003). There are four traits to be evaluated for the efficacy of hybrid prediction using omic data: (1) yield (YIELD), which is the most important trait in rice production but has a very low heritability; (2) 1000-grain weight (KGW), a highly heritable trait that can be relatively easily improved through artificial selection; (3) grain number per plant (GRAIN) and (4) tiller number per plant (TILLER). For the IMF2 population, each trait was measured in two consecutive years (1998 and 1999) and the phenotypic values are the ones measured in year 1999. For illustration purpose, we only presented the result from one trait, the KGW trait.

In this particular example of PLS analysis, we have only one response variable but we have 1619 predictors. The sample size is $n = 278$, the number of response variables is $s = 1$ and the number of predictors is $m = 1619$. Therefore, the Y matrix is $n \times s = 278 \times 1$ in dimension and the X matrix is $n \times m = 278 \times 1619$ in dimension. The data are stored in a file called "imf2.csv". Part of the data (first 10 observations) are shown in **Table 8**.

Table 8
The IMF2 rice data with 278 observations and 1619 predictors (genetic markers)

IMF2	RIL1	RIL2	yield	tiller	grain	kgw	bin1	bin2	…	Bin1619
F001	R161	R164	38.188	10.871	138.34	25.334	1	1	…	0
F002	R090	R186	43.951	14.24	133.451	22.432	-1	-1	…	-1
F003	R051	R228	47.406	13.227	139.652	26.411	-1	-1	…	0
F005	R043	R154	33.911	11.938	111.709	25.029	0	0	…	0
F006	R118	R239	33.593	16.204	87.872	23.573	-1	-1	…	-1
F008	R007	R086	46.184	14.283	121.185	26.463	0	0	…	1
F009	R182	R213	34.741	10.857	128.433	24.581	0	0	…	-1
F010	R009	R106	37.907	11.607	127.972	26.605	-1	-1	…	1
F012	R021	R172	52.659	13.852	155.903	24.437	0	0	…	-1
F014	R012	R191	44.668	11.952	149.28	25.276	0	0	…	0

The code to read the data and perform PLS analysis are given below.

```
filename imf2 "imf2.csv";
proc import datafile=imf2 out=imf2 dbms=csv replace;
run;

proc pls data=imf2 method=pls cv=split(278) details;
    model kgw = bin1-bin1619/solution;
    ods output CVResults=CV ParameterEstimates=Parm
               ResidualSummary=PRESS  XLoadings=Loading
               XWeights=Weight;
run;
```

The sample size is 278 and thus `cv=split(278)` instructs PROC PLS to perform the leave-one-out cross validation. The output is presented in all the tables and figures shown in next page.

Table 9
Output of the IMF2 rice PLS analysis

Data Set	WORK.IMF2
Factor Extraction Method	Partial Least Squares
PLS Algorithm	NIPALS
Number of Response Variables	1
Number of Predictor Parameters	1619
Missing Value Handling	Exclude
Maximum Number of Factors	15
Validation Method	278-fold Split-sample Validation

Split-sample Validation for the Number of Extracted Factors

Number of Extracted Factors	Root Mean PRESS
0	1.00361
1	0.726101
2	0.6585
3	0.629983
4	0.617455
5	0.615447
6	0.614721
7	0.609407
8	0.611222
9	0.611907
10	0.615009
11	0.617133
12	0.62567
13	0.626958
14	0.630506
15	0.638382

The minimum root mean PRESS value is 0.609407 that occurs for 7 extracted factors. This translates into a predictability of

$$R^2 = 1 - \frac{0.609407^2}{1.00361^2} = 1 - \frac{0.3713769}{1.007233} = 0.63129 \qquad (16)$$

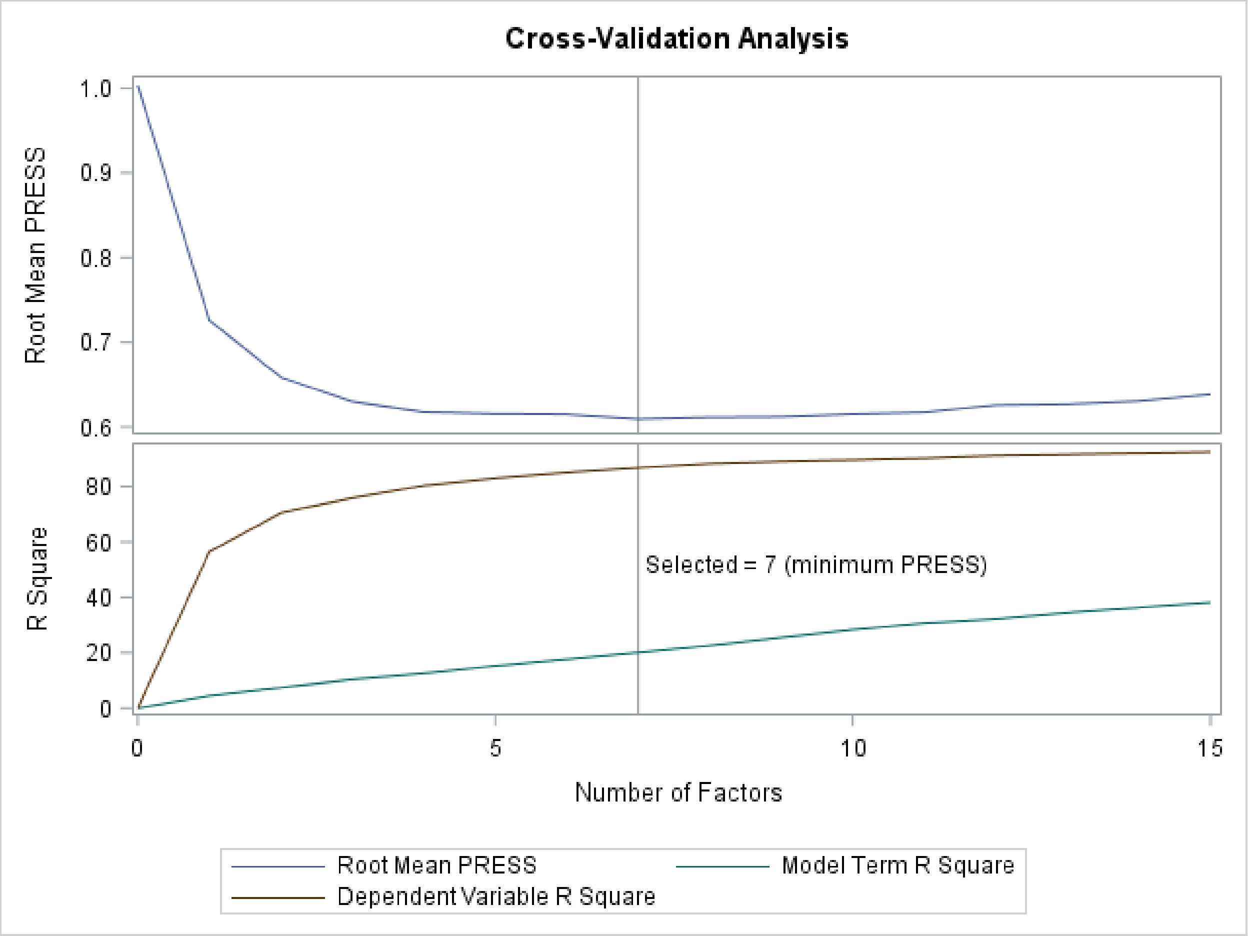
This predictability is relatively high and it reflects the high heritability of the KGW trait in rice.

Table 10
The percent variation accounted for by PLS factors

Percent Variation Accounted for by Partial Least Squares Factors

Number of Extracted Factors	Model Effects		Dependent Variables	
	Current	Total	Current	Total
1	4.5247	4.5247	56.4112	56.4112
2	2.8816	7.4063	14.0099	70.4211
3	3.0423	10.4486	5.4900	75.9111
4	2.1568	12.6054	4.3923	80.3034
5	2.6754	15.2809	2.6402	82.9436
6	2.4370	17.7179	2.0794	85.0230
7	2.3275	20.0454	1.7384	86.7614

Figure 4
Cross validation plot of the IMF2 rice data



Using the following code, we write the results to external files.

```
filename two "kgw-pls-effect.csv";
filename three "kgw-pls-r2.csv";
filename four "kgw-pls-press.csv";
filename five "kgw-pls-loading.csv";
filename six "kgw-pls-weight.csv";

proc export data=Parm outfile=two dbms=csv replace;
proc export data=CV outfile=three dbms=csv replace;
proc export data=Press outfile=four dbms=csv replace;
proc export data=Loading outfile=five dbms=csv replace;
proc export data=Weight outfile=six dbms=csv replace;

run;
```

For example, the file named “kgw-pls-effect.csv” stores the intercept and the 1619 regression coefficients. Part of this file is shown below,

Table 11

Part of the estimated regression coefficients of the IMF2 rice data

Parameter	KGW
Intercept	24.66282714
bin1	0.02721497
bin2	0.02327881
bin3	0.01548871
bin4	0.00621312
bin5	0.00496259
bin6	0.00402357
bin7	0.01279444
bin8	0.00991898
bin9	0.02692098
bin10	0.02682722
bin11	0.01767195
bin12	0.01496909
bin13	0.01112993
bin14	0.00270475
bin15	-0.01485373

6. Genomic prediction in rice via PLS in R

For the same rice data, we now perform partial least squares regression analysis using an R package called “pls” (Mevik and Wehrens 2007). The pls package implements principal component regression (PCR) and partial least squares regression (PLSR) in R (R Development Core Team 2006b), and is freely available from the Comprehensive R Archive Network (CRAN), licensed under the GNU General Public License (GPL).

```
dir<-"C:\\Users\\Lecture Notes\\PLS"
setwd(dir)

imf2<-read.csv(file="imf2.csv",header=T)
library(pls)

foldid<-imf2$foldid
y<-imf2$kgw
x<-as.matrix(imf2[,-c(1:9)])
n<-length(y)

fit <- plsr(y~x,ncomp=30,validation="LOO")
plot(RMSEP(fit), legendpos = "topright")
nn <-as.numeric( which.min(tt <-RMSEP(fit)$val[1,,]))-1
yhat<-predict(fit, newdata=x,ncomp=nn)
yp<-yhat
yo<-y
r2.fit<-cor(yo,yp)^2
r2.fit
a<-fit$Ymeans
b<-c(a,coef(fit,ncomp=nn))

rmsep0<-RMSEP(fit)$val[1,,][1]
rmsep<-RMSEP(fit)$val[1,,][nn]
R2<-1-rmsep^2/rmsep0^2
R2

write.csv(x=b,file="coefficients.csv")
```

Table 12

Root means squared error of prediction from LOOCV using the “pls” package of R

Factor	RMSEP
(Intercept)	1.925827
1 comps	1.395971
2 comps	1.264721
3 comps	1.208096
4 comps	1.184395
5 comps	1.180225
6 comps	1.181509
7 comps	1.166801
8 comps	1.179698
9 comps	1.177795
10 comps	1.181397
11 comps	1.190408
12 comps	1.201981
13 comps	1.203785
14 comps	1.213624
15 comps	1.225493

The predictability is

$$R^2 = 1 - \frac{1.166801^2}{1.925827^2} = 1 - \frac{1.361425}{3.70881} = 0.6329214 \tag{17}$$

which is slightly different from the result of PROC PLS in SAS (0.63129). Part of the estimated regression coefficients (including the intercept) are shown in Table 13 below (next page). Table 14 shows comparison of the regression coefficients between SAS and R. The two software packages are very much the same for the estimated PLS regression coefficients.

Table 13
Part of the estimated regression coefficients (including the intercept) of the rice data from “pls” of R

Bin	Parameter
bin1	24.42311
bin2	0.02594
bin3	0.0227
bin4	0.01496
bin5	0.006512
bin6	0.005355
bin7	0.004531
bin8	0.011917
bin9	0.009273
bin10	0.025042
bin11	0.025323
bin12	0.017414
bin13	0.014648
bin14	0.011454
bin15	0.002868

Table 14

Comparison of PLS regression coefficients between SAS and R

BIN	SAS	PLS
bin1	0.027215	0.02594
bin2	0.023279	0.0227
bin3	0.015489	0.01496
bin4	0.006213	0.006512
bin5	0.004963	0.005355
bin6	0.004024	0.004531
bin7	0.012794	0.011917
bin8	0.009919	0.009273
bin9	0.026921	0.025042
bin10	0.026827	0.025323
bin11	0.017672	0.017414
bin12	0.014969	0.014648
bin13	0.01113	0.011454
bin14	0.002705	0.002868
bin15	-0.01485	-0.01511
bin16	-0.0086	-0.00883
bin17	-0.00214	-0.00237
bin18	-0.01058	-0.0102
bin19	-0.01379	-0.01346
bin20	-0.0324	-0.03221
bin21	-0.02711	-0.02722
bin22	-0.02451	-0.02413
bin23	-0.0331	-0.03108
bin24	-0.02939	-0.02839
bin25	-0.02693	-0.0273
bin26	0.011831	0.010836
bin27	0.022311	0.021763
bin28	0.020002	0.019392
bin29	0.029855	0.029845
bin30	0.001884	0.001114

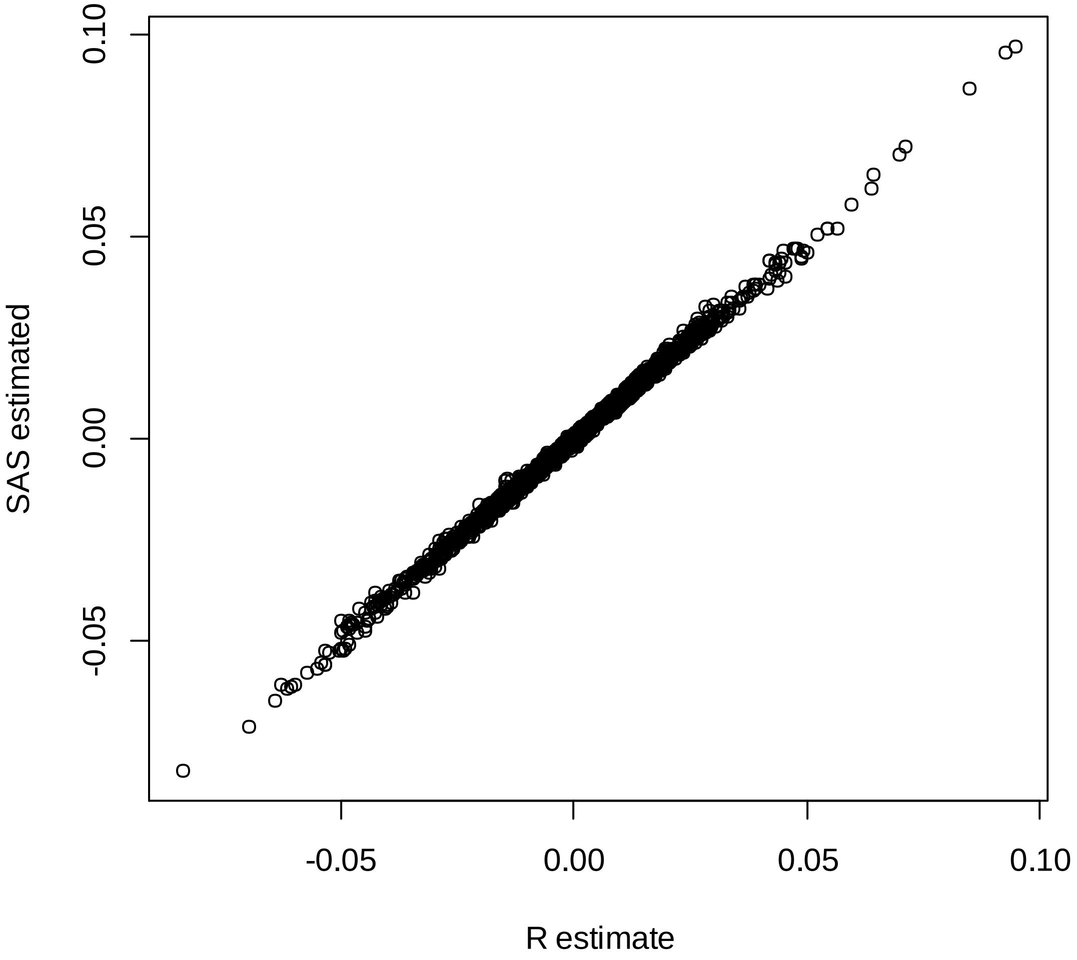


Figure 5. Comparison of PLS regression coefficients between SAS and R.