The Implementation of Noise Addition Partial Least Squares

by

Jürgen Johann Möller

Assignment presented in partial fulfilment
of the requirements for the degree of Master of Commerce at
Stellenbosch University

Study Leader: Prof. M. Kidd

March 2009

Declaration

By submitting this assignment electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 5 March 2009

Abstract

When determining the chemical composition of a specimen, traditional laboratory techniques are often both expensive and time consuming. It is therefore preferable to employ more cost effective spectroscopic techniques such as near infrared (NIR). Traditionally, the calibration problem has been solved by means of multiple linear regression to specify the model between **X** and **Y**. Traditional regression techniques, however, quickly fail when using spectroscopic data, as the number of wavelengths can easily be several hundred, often exceeding the number of chemical samples. This scenario, together with the high level of collinearity between wavelengths, will necessarily lead to singularity problems when calculating the regression coefficients.

Ways of dealing with the collinearity problem include principal component regression (PCR), ridge regression (RR) and PLS regression. Both PCR and RR require a significant amount of computation when the number of variables is large. PLS overcomes the collinearity problem in a similar way as PCR, by modelling both the chemical and spectral data as functions of common latent variables.

The quality of the employed reference method greatly impacts the coefficients of the regression model and therefore, the quality of its predictions. With both **X** and **Y** subject to random error, the quality the predictions of **Y** will be reduced with an increase in the level of noise. Previously conducted research focussed mainly on the effects of noise in **X**. This paper focuses on a method proposed by Dardenne and Fernández Pierna, called Noise Addition Partial Least Squares (NAPLS) that attempts to deal with the problem of poor reference values.

Some aspects of the theory behind PCR, PLS and model selection is discussed. This is then followed by a discussion of the NAPLS algorithm. Both PLS and NAPLS are implemented on various datasets that arise in practice, in order to determine cases where NAPLS will be beneficial over conventional PLS. For each dataset, specific attention is given to the analysis of outliers, influential values and the linearity between **X** and **Y**, using graphical techniques.

Lastly, the performance of the NAPLS algorithm is evaluated for various settings of the input parameters, in order to determine if it is possible to fine-tune the results obtained with the algorithm.

Opsomming

Met die bepaling van die chemiese samestelling van ‘n monster, is die gebruik van tradisionele laboratorium tegniekse dikwels duur en tydrowend. Dit is vir dié rede dat die gebruik van meer koste effektiewe spektroskopiese tegnieke soos naby infrarooi skandering (NIR) verkies word. Tradisioneel is die kalibrasie problem opgelos met behulp van ‘n veelvuldige lineêre regressie om die verwantskap tussen **X** en **Y** te modelleer. Tradisionele regressie tegnieke is ongelukkig nie geskik vir spektroskopiese data nie, aangesien die aantal golflengtes maklik meer as die aantal waarnemings kan wees. Laasgenoemde scenario, tesame met die hoë vlakke van kollineariteit tussen golflengtes, sal noodwendig lei tot singulariteit probleme met die bepaling van die regressie koëffisiënte.

Metodes soos hoofkomponent regressie (HKR), rif regressie (RR) en parsiële kleinste kwadrate (PKK) regressie kan gebruik word om die probleem van kollineariteit aan te spreek. Beide HKR en RR vereis ‘n wesenlike hoeveelheid berekeninge indien die aantal veranderlikes groot is. PKK regressie spreek die probleem van kollineariteit aan op ‘n wyse soortgelyk aan HKR, deur die chemiese en spektroskopiese data te modelleer as funksies van onderliggende latente veranderlikes.

Die model se koëffisiënte, asook die gehalte van die model se voorspellings, word grootliks beïnvloed deur die gehalte van die verwysings metode. Aangesien beide **X** and **Y** onderhewig is aan lukrake foute, sal die gehalte van die voorspellings van **Y** daal met ‘n toename in ruis. Vorige navorsing het hoofsaaklik gefokus op die effek van ruis in **X**. Hierdie werkstuk fokus op ‘n metode van Dardenne and Fernández Pierna, genoemd “Noise Addition Partial Least Squares” (NAPLS), wat poog om die probleem van swak verwysings waardes aan te spreek.

Die onderliggende teorie agter HKR, PKK en model seleksie word oorsigtelik bespreek. Beide PKK en NAPLS word geïmplimenteer op verskeie praktiese datastelle, sodat daar vasgestel kan word vir watter gevalle die gebruik van NAPLS in plaas van PKK voordelig sal wees. Vir elke datastel, word aandag geskenk aan die grafiese ontleding van uitskieters, invloedryke waarnemings en die lineêre verwantskap tussen **X** en **Y**.

Laastens word die NAPLS algoritme gevalueer vir verskillende waardes van die invoer parameters. Hierdie word gedoen in ‘n poging om te bepaal of dit moontlik is om die resultate van die algoritme te verfyn.

Acknowledgements

I would like to express my sincere gratitude to the following people:

- My parents for all their love and support.
- Prof. Martin Kidd, my study leader, for his patience and valuable guidance throughout the writing of this document.
- Ivona Contardo, who provided much appreciated moral support and for tending to the grammar of this paper.
- My good friend, Corné Nagel, who was always more than willing to help me with complex mathematical problems.

Table of content

Declaration ii

Abstract iii

Opsomming iv

Acknowledgements v

Table of content vi

1 Introduction 1

2 Overview of PCR and PLS 3

2.1 The need for dimension reduction 3

2.2 Principal Component Analysis 4

2.3 Principal Component Regression 7

2.4 Partial Least Squares (Projection to Latent Structures) 8

3 Model Selection 12

3.1 Bias, Variance and Model Complexity 12

4 Noise addition PLS 17

5 Application of PLS to Practical Datasets 20

5.1 Dataset 1 20

5.2 Dataset 2 24

5.3 Dataset 3 28

5.4 Dataset 4 37

5.5 Dataset 5 39

5.6 Dataset 6 42

5.7 Dataset 7 44

5.8 Dataset 8 47

5.9 Dataset 9 49

6 Practical Implementation of Noise Addition PLS 53

6.1 Introduction 53

6.2 Dataset 1 53

6.3 Dataset 2 55
6.4 Dataset 3A 57
6.5 Dataset 3B 59
6.6 Dataset 3C 60
6.7 Dataset 4 62
6.8 Dataset 5 63
6.9 Dataset 6 64
6.10 Dataset 7 66
6.11 Dataset 8 67
6.12 Dataset 9 69
6.13 The effect of the size of the calibration set 70

7 The effect of changing the number of iterations 73
7.1 The effect of the number of iterations on the stability of the model 73

8 Conclusion and Recommendations for Future Research. 77

9 Addendum A: NAPLS increasing the size of the calibration set. 78

10 Addendum B: Detailed simulation output 82
10.1 Summary of NAPLS test results 82
10.2 NAPLS Results: 5 Outer Loops; 100 Inner Loops; 5% Noise 83
10.3 NAPLS Results: 5 Outer Loops; 100 Inner Loops; 10% Noise 86
10.4 NAPLS Results: 15 Outer Loops; 100 Inner Loops; 10% Noise 90
10.5 NAPLS Results: 5 Outer Loops; 300 Inner Loops; 10% Noise 94
10.6 NAPLS Results: 5 Outer Loops; 500 Inner Loops; 10% Noise 97

11 Addendum C: Code examples 102

12 References 108

1 Introduction

When determining the chemical composition of a specimen, traditional laboratory techniques are often both expensive and time consuming. It is therefore preferable to employ more cost effective spectroscopic techniques such as near infrared (NIR). A disadvantage of this technique is the difficulty in finding specific frequency regions (wavelengths) where the constituents of interest selectively absorb or emit light. The problem can be solved by measuring the near infrared reflectance for P distinct frequencies for specimens with known chemical composition. A method is then needed to relate the P measurements of **X** to **Y**. (Here, **X** is a N x P matrix containing P NIR reflectance measurements for N different specimens and **Y** is a N x P' matrix containing P' known chemical compositions for the same N specimens.)

Traditionally, the calibration problem has been solved by means of multiple linear regression to specify the model between **X** and **Y**. (Usually by means of a separate model for each variable / column of **Y**.) Traditional regression techniques, however, quickly fail when using spectroscopic data, as the number of wavelengths can easily be several hundred, often exceeding the number of chemical samples. This scenario, together with the high level of collinearity between wavelengths, will necessarily lead to singularity problems when calculating the regression coefficients. Variable selection can be used as a way of limiting the number of wavelengths, but is not recommended as eliminating some variables will most probably lead to a loss of information.

Other ways of dealing with the collinearity problem include principal component regression (PCR), ridge regression (RR) and PLS regression. Both PCR and RR require a significant amount of computation when the number of variables is large. Ridge regression has the added problem of estimating the ridge parameter. Principal component regression also has the problem of deciding which principal components to delete (Helland, 1988).

In contrast, PLS regression is claimed to overcome most of these problems to some extend. PLS overcomes the collinearity problem in a similar way as PCR, by modelling both the chemical and spectral data as functions of common latent variables. Simulations studies have also shown that PLS minimizes MSE with a smaller number of factors than PCR (Helland, 1988). Furthermore, PLS provides a unique way of choosing the number of factors and requires less computation than PCR and RR.

Intuitively, the quality of the reference method used, greatly impacts the coefficients of the regression model and therefore, the quality of its predictions. With both **X** and **Y** subject to random error, the quality of the predictions of **Y** will be reduced with an increase in the level of noise. A few studies have been done (Dardenne and Fernández Pierna, 2006) to address the affect of noise on calibrations,

with most of them being done on noise addition on **X**. Dardenne and Fernández Pierna propose a method (called Noise Addition Partial Least Squares / NAPLS) to build more robust PLS models in order to deal with the problem of poor reference values.

The goal of this research is to implement the NAPLS algorithm and investigate its performance on various datasets. This paper is organised as follows: Firstly, a brief overview of principal component regression and PLS regression is given. This is then followed by a short discussion on methods of model selection. Finally, in the main section, the application of the Noise Addition PLS (NAPLS) algorithm is explored. Emphasis is placed on the application of NAPLS to datasets that arise in practice, in order to determine cases where NAPLS will be beneficial over conventional PLS. Lastly, the performance of the NAPLS algorithm is evaluated for various settings of the input parameters, in order to determine if it is possible to fine-tune the results obtained with the algorithm.

2 Overview of PCR and PLS

2.1 The need for dimension reduction

As the number of dimensions of a space increases, an exponentially increased number of observations are needed to adequately describe the space. This problem is known as the curse of dimensionality.

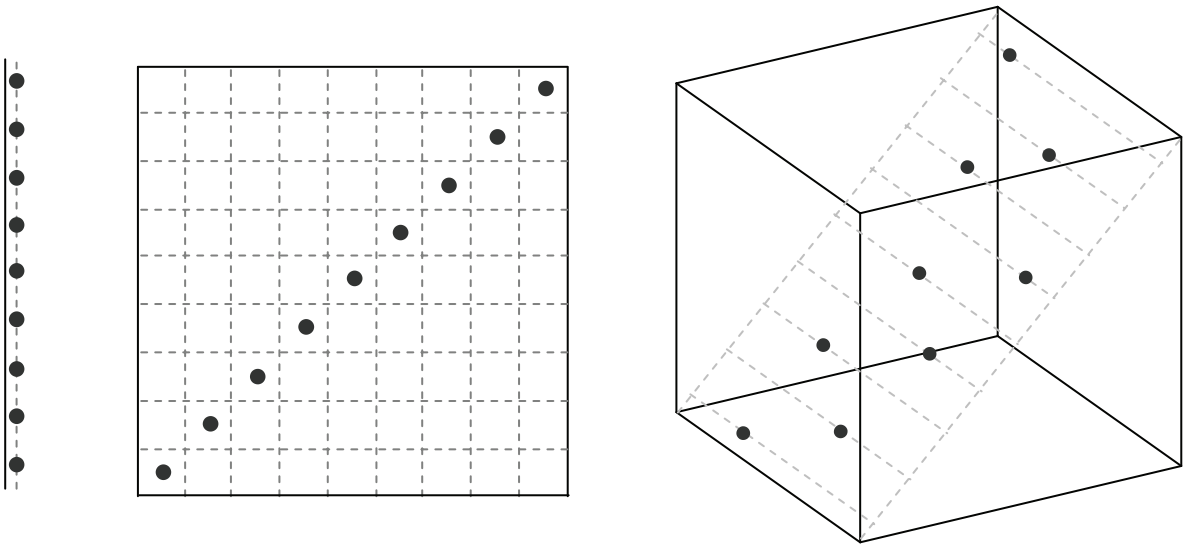

FIGURE 2.1 *10 points represented in One, Two and Three dimensions respectively. As the number of dimensions are increased, the space is increasingly more sparsely populated, and therefore less adequately described.*

This problem is illustrated in **FIGURE 2.1**: In the first illustration, the ten points are adequate at sufficiently describing a one-dimensional space. As the number of dimensions are increased (as shown in the second and third illustrations), the space becomes increasingly sparsely populated. Also, as the number of dimensions increase, the distance between points increases.

Traditional multiple linear regression can yield undesirable results when the number of independent variables is large relative to the number of observations, or when the independent variables are highly correlated. The goal is therefore is to find a lower dimensional space that is better described by the points.

Dimension reduction can be achieved by eliminating redundant and irrelevant dimensions. When two variables are highly correlated, using both instead of one, adds little or no additional knowledge. The additional variable is therefore redundant. If an independent variable has little or no relation with the dependent variable, and it has no interactions with any of the other input variables, it is irrelevant in the modelling context and can therefore be disregarded.

Variable selection can be employed in order to reduce the number of input variables. With variable selection, only the variables that add significantly towards describing the dependent variable are retained. Unfortunately, variable selection is a highly discreet process and may increase the variance of predictions as an unwanted side effect.

If variables are highly correlated, it can be assumed that there are a few underlying latent factors that produced the observed data. The observed variables are just attempts to measure these underlying factors. Dimension reduction can be achieved by identifying these latent factors and disregarding unwanted noise. Two methods of achieving this in a regression sense will be discussed next.

2.2 Principal Component Analysis

As mentioned, ordinary multiple linear regression can not be applied when the number of variables is large relative to the number of observations, or when the independent variables are highly correlated. Principal component regression (PCR) tries to overcome these problems by first applying principal component analysis (PCA) on the matrix of independent variables, **X**, and then applying multiple linear regression on the scores of the significant principal components.

PCA starts of by letting the n observations of the p independent variables, x_1, x_2, …, x_p, form a swarm of points in p-dimensional space. Although PCA can be applied to any distribution of **X**, it is easier to visualise it if the swarm of points is ellipsoidal.

If the variables x_1, x_2, …, x_p are correlated, then the swarm of points is not oriented parallel to any of the axes. PCA determines the natural (principal) axes of the swarm of points, by first translating the swarm to its origin, $\bar{\mathbf{x}}$, and then rotating the axes in such a way that the new variables (called principal components) are uncorrelated. This is illustrated in **FIGURE 2.2**.

The axes are rotated by multiplying each of the $\mathbf{x}_i$ with an orthogonal matrix **A**:

$$\mathbf{z}_i = \mathbf{A}\mathbf{x}_i ,$$

or simply as:

$$\mathbf{Z} = \mathbf{X}\mathbf{A}$$

Since **A** is orthogonal, $\mathbf{A}'\mathbf{A} = \mathbf{A}\mathbf{A}' = \mathbf{I}$. The distance to the origin is therefore unchanged:

$$\mathbf{z}_i'\mathbf{z}_i = (\mathbf{A}\mathbf{x}_i)'(\mathbf{A}\mathbf{x}_i) = \mathbf{x}_i'\mathbf{A}'\mathbf{A}\mathbf{x}_i = \mathbf{x}_i'\mathbf{x}_i$$

The orthogonal matrix, **A**, is found by finding the rotation matrix that rotates the axes to line up with the natural axes of the swarm of points, thereby producing a new set of uncorrelated variables z_1, z_2, …, z_p called the principal components.

It should be noted that the maximum number of principal components is the minimum of p (the number of independent variables), or n-1 (the number of observations - 1). In order to simplify

notation, the maximum number of principal components are also denoted by p, but an upper bound of n-1 applies if the number of independent variables is larger than the number of observations.

The sample covariance matrix of **Z**, $\mathbf{S}_z$, is diagonal and is given by:

$$\mathbf{S}_z = \mathbf{A}\mathbf{S}_x\mathbf{A}' = \begin{pmatrix} s_{z_1}^2 & 0 & \cdots & 0 \\ 0 & s_{z_2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & s_{z_p}^2 \end{pmatrix}$$

As the newly acquired covariance matrix is diagonal, all the covariance terms are zero, and therefore, all the principal components are uncorrelated. A further property of the principal components, as shown in **FIGURE 2.2**, is that the first principal component captures the most of the variance, the second principal component the second most variance, etc. The last principal component thus captures the minimum amount of variance of the data.

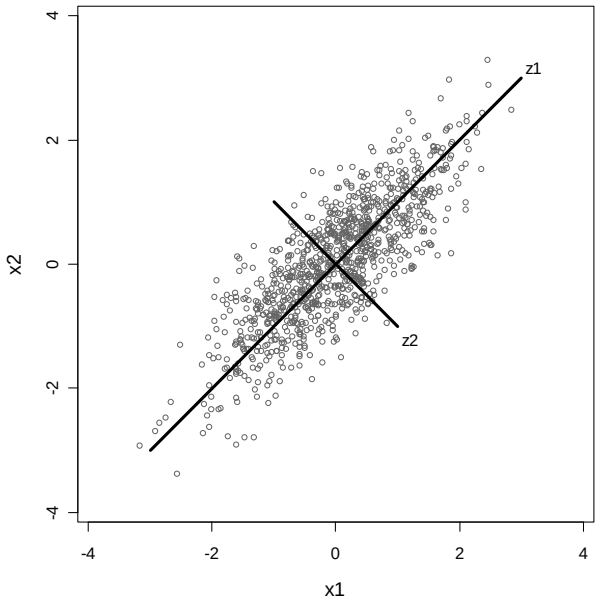

FIGURE 2.2. *Principal components of some input data points. The first principal component is the largest and is the direction that maximizes the variance of the projected data. The last principal component is the smallest and is the direction that minimizes the variance of the projected data.*

By noting that the eigen values of $\mathbf{S}_z$ are equal to the diagonal elements of $\mathbf{S}_z$, we have:

$$\lambda_i = s_{z_i}^2$$

and therefore:

$$\lambda_1 > \lambda_2 > \ldots > \lambda_p$$

since each principal component accounts for the maximum amount of variance, given the variance already accounted for by the larger (preceding) principal components.

The proportion of the variance that is explained by the first k principal components can now be expressed as:

$$\text{Proportion of variance} = \frac{\lambda_1 + \lambda_2 + ... + \lambda_k}{\lambda_1 + \lambda_2 + ... + \lambda_p}$$

If the original variables are highly correlated, the effective dimensionality is much smaller than p, in which case, the proportion of variance explained by the first k (for k small) principal components will be close to 1. On the other hand, if the variables are fairly uncorrelated, the intrinsic dimensionality is close to p and the principal components do little more than to merely reproduce the original variables. In this case no useful dimension reduction can be achieved.

The final step in principal component analysis is to decide how many components to retain. There is however no clear cut rule, but the following guidelines have been proposed:

1. Retain sufficient components to account for a specified proportion of the total variance, say 90%. This however leaves the risk of setting the threshold too high, in which case components that are sample or variable specific might be retained.
2. Retain the components whose eigenvalues are greater than the average of the eigenvalues, $\overline{\lambda} = \sum_{i=1}^{p} \lambda_i$. The major drawback of this approach is that the average of the eigenvalues might be distorted by extreme values for the largest or smallest eigenvalues.
3. Make use of a scree plot, a plot of the λ_i versus i . A natural break should separate the large and small eigenvalues. Unfortunately it is not always clear where the break should be.

Previously, the rotation of **X** onto principal axes was given as

$$\mathbf{Z} = \mathbf{XA}$$

Since $\mathbf{A}$ is orthogonal, multiplying both sides by $\mathbf{A}'$ will yield the original input data:

$$\mathbf{ZA}' = \mathbf{XAA}'$$
$$\therefore \mathbf{X} = \mathbf{ZA}'$$

$\mathbf{Z}$ is also called the matrix of principal component scores and each column of $\mathbf{Z}$ is known as a score vector. $\mathbf{A}$ is commonly known as the loading matrix.

After the number of significant principal components has been determined, the original data can be approximated by the reduced versions of **Z** and **A**, denoted by $\mathbf{Z_k}$ and $\mathbf{A_k}$, **X** can now be written as:

$$\mathbf{X} = \mathbf{Z}_k \mathbf{A}'_k + \mathbf{E}$$

or, otherwise stated: **X** = Structure + Error.

2.3 Principal Component Regression

The multiple regression model is given by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

With **y** and **X** the matrix of dependent and independent variables respectively, $\boldsymbol{\beta}$, the matrix of regression coefficient and $\boldsymbol{\varepsilon}$ the matrix of residuals.

As mentioned, multiple linear regression (MLR) yields unstable results when the independent variables are significantly correlated or when the number of variables is large relative to the number of observations. A way is therefore needed to first deal with these problems, before MLR can be continued.

Variable selection can be employed as a way to deal with highly correlated variables. This is however not always feasible for spectroscopic data, as variable selection is a very discrete process and will most possibly lead to a loss in information.

On the other hand, the score vectors produced by PCA are completely uncorrelated and have the added advantage that the first few principal components account for most of the variance in the data, for cases where the original input variables are highly correlated. It is therefore feasible to substitute the original input data **X**, with the principal component scores **Z**. The MLR model is now given by:

$$\mathbf{y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

and is now called principal component regression or PCR.

If all principal components are to be kept, PCR essentially is the same as MLR, as **Z** accounts for all the information in **X**. As shown earlier, it is expected that the insignificant principal components contain little more than random noise, and should be disregarded. The final issue to be addressed is the number of principal components to retain. With PCA, there is unfortunately no clear cut rule, with different methods sometimes suggesting different solutions. Fortunately, with PCR, there is a better alternative – use the number of components that minimise the error of prediction.

This is accomplished as follows: Starting with only the first principal, predict a completely independent set of data. Continue adding sequential components, until the prediction error starts to increase. The components that minimise the prediction error (and therefore maximised the predictive power of the model) should be retained.

Although PCR is of much use in dealing with collinearity and reducing dimensionality, it suffers from the fact that it only maximises the amount of variance explained by the first few principal components. There is no guarantee that it will produce a structure that is correlated with the dependent variable. It

is quite possible that the most significant components contain variance that that is not correlated with **y**. Even worse, it may very well be that some of the variance captured by the components deemed insignificant might capture some of the variance correlated with **y**. Unfortunately these components get discarded due to the magnitudes of factors that are irrelevant in a (**X**,**Y**)-regression sense. As this might very well lead to a loss in valuable information, a way is needed to simultaneously maximise both the variance in the projected components, as well the covariance between the dependent variable and the projected components.

2.4 Partial Least Squares (Projection to Latent Structures)

PLS is able to obtain the same prediction results as PCR, but based on a smaller number of components. This is accomplished by allowing the structure of the y-variable to directly intervene with the decomposition of **X**.

Initially, the decomposition of the **X**- and **Y** spaces can be viewed as separate two PCA decompositions. The score and loading vectors are given by **T** and **P**, respectively, for **X**, and **U** and **Q**, respectively, for **Y**. **X** also has an additional loading matrix **W**.

The main difference between PCA and PLS however, is that $\mathbf{u_1}$ (of the **Y** space) acts as a starting point for the proxy $\mathbf{t_1}$ vector, thereby letting the **Y** space guide the, otherwise PCA-like, decomposition of **X**. This leads to the calculation of the X-loadings which is now called the loading weights vector, **w**. In turn, **w** is again used to calculate the **t**-vectors of the **X** space in a PCA like fashion. Subsequently, $\mathbf{t_1}$ is used as proxy for $\mathbf{u_1}$ with the decomposition of the **Y** space. In this way, the **Y** space guides the decomposition of the **X** space and vice versa.

The core of the PLS algorithm is therefore based around the interdependent $\mathbf{u}_1 \rightarrow \mathbf{t}_1$ and $\mathbf{t}_1 \rightarrow \mathbf{u}_1$ substitutions in an iterative way until convergence is reached. At convergence, a final set of (**t,w**) and corresponding (**u,q**) vectors are calculated for the current PLS component, for the X and Y spaces respectively. The decomposition of both spaces, based on interchanged score vectors thus achieves the goal of modelling the X and Y space interdependently. In this way, the X and Y information is balanced and large variations in X that are not correlated with Y are effectively reduced. This then solves the weakness of the two stage PCR approach.

2.4.1 The PLS-2 NIPALS Algorithm (Multivariate Y)

0. Centre and scale both the **X** and **Y**-matrices.
   Initialize the indexes: i = 1; $\mathbf{X}_i = \mathbf{X}$; $\mathbf{Y}_i = \mathbf{Y}$
1. For $\mathbf{u}_i$, choose any column of **Y** as the initial proxy **u**-vector.

2. $\mathbf{w}_i = \frac{\mathbf{X}_i^T \mathbf{u}_i}{|\mathbf{X}_i \mathbf{u}_i|}$ (**w** is normalized)
3. $\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i$
4. $\mathbf{q}_i = \frac{\mathbf{Y}_i^T \mathbf{t}_i}{|\mathbf{Y}_i^T \mathbf{t}_i|}$ (**q** is normalized)
5. $\mathbf{u}_i = \mathbf{Y}_i \mathbf{q}_i$
6. If $|\mathbf{t}_{i.new} - \mathbf{t}_{i.old}|$ < convergence limit, stop; else go to step 2.
   [equivalent to $|\mathbf{u}_{i.new} - \mathbf{u}_{i.old}|$ < convergence limit]
7. $\mathbf{p}_i = \frac{\mathbf{X}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}$
8. $b_i = \frac{\mathbf{u}_i^T \mathbf{t}_i}{\mathbf{t}_i^T \mathbf{t}_i}$
9. $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i \mathbf{p}_i^T$
   $\mathbf{Y}_{i+1} = \mathbf{Y}_i - b_i \mathbf{t}_i \mathbf{q}_i^T$
10. If f = the desired number of components, stop; else set i = i + 1 and go to step 1

Explanation of the NIPALS PLS-2 algorithm

0. Scaling and centering: Various methods exist, but most commonly, this is done be first subtracting the mean and scaling the variables to unit variance. (This method is known as auto scaling.)
1. The algorithm should be started with a proxy for the **u**-vector. Although any column of **Y** will do, it is advantageous to choose the largest column of **Y** namely max |$\mathbf{Y}_i$|
2. Calculation of the loading weight vector $\mathbf{w}_i$, for iteration i.
3. Calculation of the score vector $\mathbf{t}_i$, for iteration i.
4. Calculation of the loading vector $\mathbf{q}_i$, for iteration i.
5. Calculation of the score vector $\mathbf{u}_i$, for iteration i.

Steps 3 and 5 represent the projection of the object vectors onto the ith PLS-component in the **X** and **Y** variable spaces respectively. Equivalently, steps 2 and 4 are the projection of the **w** and **q** variable vectors onto the *i*th PLS-component in the corresponding object spaces. These steps all resemble a form of regression. In this sense, the PLS-NIPALS algorithm can be viewed as a set four independent "criss-cross" X-/Y-space regressions. The loading weights vector, **w**, also indicates the direction which simultaneously maximizes both the X-variance and the Y-variance.

6. Convergence. The PLS-NIPALS algorithm normally converges to a stable solution in fewer iterations than the equivalent PCA case. At convergence, the PLS optimization criterion is

composed of the product of both a modelling optimization term and a prediction error minimization term.

7. Calculation of the p-loadings for the X-space. These are mainly needed for subsequent updating.
8. Calculation of the regression coefficients for the inner X-Y space regression by means of a univariate regression of **u** upon **t**. The inner relation is the operative link of the PLS-model and is estimated one dimension at a time; hence the acronym PLS – Partial Least Squares.
9. Updating or deflation: $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i\mathbf{p}_i^T$ and $\mathbf{Y}_{i+1} = \mathbf{Y}_i - b_i\mathbf{t}_i\mathbf{p}_i^T$

   In this step the i^{th} component is subtracted from both spaces. The **p**-vectors are used to update **X**, instead of the **w**-vectors, in order to ensure that the **t**-vectors are orthogonal.
10. The PLS model. $\mathbf{TP}^T$ and $\mathbf{UQ}^T$ is calculated and deflated for one component dimension at a time. Finally, **X** and **Y** are given by: $\mathbf{X} = \sum_A \mathbf{TP}^T + \mathbf{E}$ and $\mathbf{Y} = \sum_A \mathbf{UQ}^T + \mathbf{F}$, where A is the optimal number of PLS components to retain.

2.4.2 Loadings (p) and Loading weights (w)

All PLS calibrations result in two sets of X-loadings for the same model, namely the loadings, **P**, and the loading weights, **W**.

The P-loadings are similar to the PCA loadings, as they express the relationship between **X** and its scores, **T**. As such, the P-loadings can be interpreted in the same way as PCA loadings. Quite often, **P** and **W** will be quite similar. This implies that the dominating structures in **X** are correlated with **Y**. On the other hand the duality between **P** and **W** will be important when the **P** and **W** directions differ significantly.

The loading weights, **W**, represent the effective loadings resembling the regression relationship between **X** and **Y**. The vector $\mathbf{w_1}$ therefore gives the direction in which all objects in the **X** space is projected, to form the first PLS component. The size of the difference in $\mathbf{w_1}$ and $\mathbf{p_1}$ is therefore indicative of the amount of influence **Y** had the decomposition of **X**.

Finally, there is also a set of loadings for the Y-space, namely **Q**. These are the regression coefficients of the Y-variables onto the scores, **U**. Together, **Q** and **W** may be used to interpret relationships between the X- and Y-variables, as well as interpret patterns in their related score plots.

After the PLS model is constructed, new values can be predicted using the normal regression equation, **Y** = **XB**. For normal multiple linear regression, the regression coefficients matrix, **B** is given by:

$$\mathbf{B} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{Y}^T$$

For a PLS solution, it becomes clear that both **P** and **W** are important, as the **B** matrix is calculated as follows:

$$\mathbf{B} = \mathbf{W}(\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{Q}^{\mathrm{T}}$$

The univariate version of the PLS algorithm will now be given as a special case of the multivariate PLS algorithm.

2.4.3 The PLS1 NIPALS Algorithm (Univariate y)

0. Centre and scale both the **X** and **y**.
   Initialize the indexes: i = 1; $\mathbf{X}_i = \mathbf{X}$; $\mathbf{y}_i = \mathbf{y}$
   As **y** has only one column, **y** is the proxy **u**-vector.

1. $\mathbf{w}_i = \dfrac{\mathbf{X}_i^{\mathrm{T}}\mathbf{y}_i}{|\mathbf{X}_i\mathbf{y}_i|}$ (**w** is normalized)

2. $\mathbf{t}_i = \mathbf{X}_i\mathbf{w}_i$

3. $q_i = \dfrac{\mathbf{t}_i^{\mathrm{T}}\mathbf{y}_i}{|\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i|}$

4. $\mathbf{p}_i = \dfrac{\mathbf{X}_i^{\mathrm{T}}\mathbf{t}_i}{\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i}$

5. $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{t}_i\mathbf{p}_i^{\mathrm{T}}$
   $\mathbf{y}_{i+1} = y_i - q_i\mathbf{t}_i$

3 Model Selection

3.1 Bias, Variance and Model Complexity

Suppose we have a data set containing a target variable, y, and a set of input variables X_i. This dataset is now randomly divided into two datasets, with n and m observations each, to form a training and validation dataset. Further, let $\hat{f}(X)$ be the prediction model that is estimated using the training sample.

As the complexity of $\hat{f}$ increases, it is able to adapt to more complicated underlying structures (a decrease in bias). As shown in **FIGURE 3.1**, the training error continues to drop as the model complexity increases. Therefore, training error is not a good estimate of the goodness of fit, as it is possible to train a model complex enough to predict the training data with zero error. Overly complicating a model will necessarily lead to overfitting, thereby compromising the model's ability to generalise. As a consequence, the model will perform poorly when predicting new data, as shown in **FIGURE 3.1**.

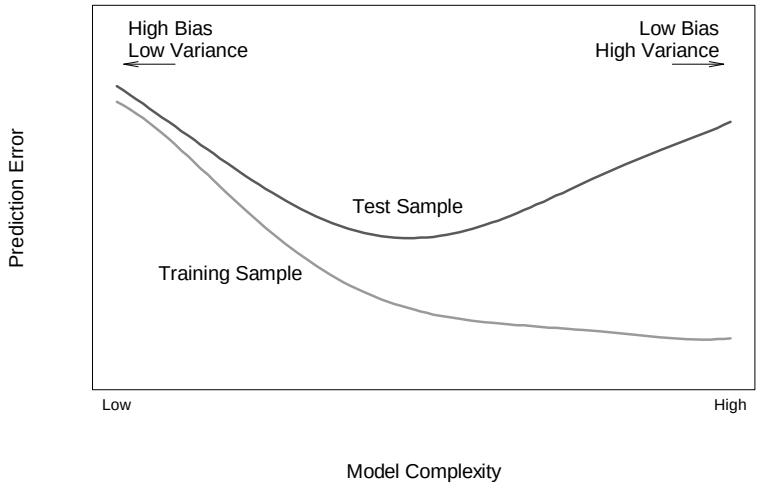

FIGURE 3.1. *The behaviour of prediction error for the training- and test sample as the model complexity is varied. An increase in model complexity reduces the prediction error for the training sample, but also decreases the model's ability to generalize.*

It is therefore important to estimate the error curve for two reasons:

1. Model selection – This is done in order to estimate the performance of different models in order to choose the (approximate) best one.
2. Model assessment – After choosing the final model, its prediction (generalisation) error is measured using a new set of data.

If data is available in abundance, the best approach is to randomly divide to dataset into three parts: a training set, a validation set and a test set. The training set is used to fit various models, the validation

set is used to estimate prediction error (to aid model selection) and the test set is used to assess the generalisation error of the final model. It is important to note that the test set should at all cost be kept separate until the end of the analysis. Failure to do so, for example repeatedly using the testing set to choose the model with the smallest test error, will cause the true test error to be underestimated.

A typical allocation of observations to these three datasets might be 50% for training, for 25% validation and 25% for testing. It is however difficult to specify a general rule, as the allocation is dependent on the signal-to-noise ratio in the data, the complexity of the model to be fitted, as well as the availability of data. Typically, data is not available in abundance, in which case one has to resort to analytical or resampling techniques in order to approximate the validation step. Two resampling techniques, namely cross-validation and the bootstrap, will be discussed.

3.1.1 Cross-Validation

Cross-validation is most probably the simplest and most wide used method for estimating the prediction error. In the absence of enough data to allow for a training, validation and test data set to be constructed, cross-validation provides a way of directly estimating the generalization error when $\hat{f}(X)$ is applied to an independent test sample.

To overcome the problem of data scarcity, K-fold cross-validation uses part of the available data to fit the model and a different part to test it. This is achieved by dividing the data into K roughly equal sized parts. For the k^{th} part, the model is fitted to the remaining K-1 parts. This model then is used to calculate the prediction error for the k^{th} part. Repeating this for k = 1, 2, …, K, the cross-validation estimate of prediction error is given by the average of the K individual estimates:

Let $\hat{f}^{-k}(x)$ be the model fitted with the k^{th} part removed. Now, let $\kappa(i)$ be the block of the i^{th} observation. Assuming quadratic loss, the cross-validation estimate of prediction error is given by:

$$\mathrm{CV} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f^{-\kappa(i)}(x_i))^2$$

Typical values for K are 5 or 10. The case where K = N is known as leave-one-out cross-validation. This implies that $\kappa(i) = i$, meaning that for each observation, the model is trained on all the data excluding the i^{th} observation.

What value should be chosen for K? For K = N, CV is approximately unbiased for the true prediction error, but can have a high variance due to the high level of similarity between the N different training sets. Setting K = N can also be computationally expensive, as the learning method needs to be applied N times.

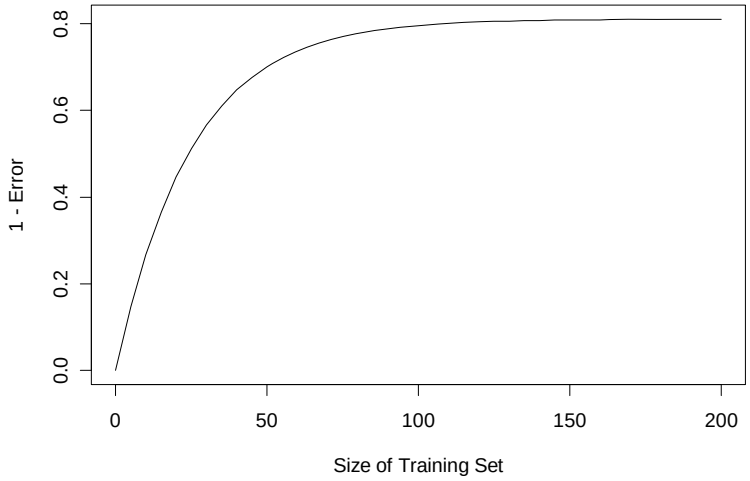

FIGURE 3.2. *A hypothetical learning curve. With a dataset of 200 observations, fivefold cross-validation would use training sets of 160, which would behave much like the full set. For a dataset with only 50 observations, fivefold cross-validation would use training sets of size 40, resulting in a considerable overestimate of prediction error.*

For K larger, say 5, CV has lower variance. However, depending on how the performance of the learning method varies with the size of the training set, bias might be a problem. **FIGURE 3.2** gives a hypothetical "learning curve" for a learning method. The method's performance increases as the size of the training set increases to 100 observations. Any further increase thereafter only provides a marginal improvement in performance. If the training set has 200 observations, fivefold cross-validation will use training sets of 160 which would yield almost the same performance as a training set of 200 observations. Cross-validation would thus not suffer from much bias. On the other hand, if the original training set has only 50 observations, cross-validation will use training sets of 40 observations. From **FIGURE 3.2**, this will severely underestimate "1 – Error", implying that the cross-validation estimate of prediction error will be upward biased. It can therefore be concluded that, if the learning curve has a considerable slope for the given training set size, five- and tenfold cross-validation will overestimate the true prediction error.

3.1.2 Bootstrap

The bootstrap is a general tool for assessing statistical accuracy. The basic idea is to randomly draw samples, with replacement, from the training data. All of these samples are of same size the original training set. This is repeated B times, for B large, thereby producing B datasets. The bootstrap estimate of prediction error is then computed by fitting the model to each of the bootstrap datasets and recording how well it predicts the original training set. By letting $\hat{f}^{*b}(x_i)$ be the predicted value at x_i, from the model fitted to bth bootstrap sample, the bootstrap estimate of prediction error is given by:

$$\widehat{\mathrm{Err}}_{\mathrm{boot}} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N}(y_i - f^{*b}(x_i))$$

It is not difficult to see why this approach does not provide a good estimate of prediction error in general: The bootstrap datasets (that are used for training) and the original training set (used for testing) have observations in common. This overlap can make overfit predictions look unrealistically good. It is for this very exact reason that cross-validation uses non-overlapping datasets for the training and test samples.

The bootstrap estimate can be improved by employing this non-overlapping requirement: For each observation, only keep track of the predictions from the bootstrap samples that do not contain that observation. By letting C^{-i} be the indices of the bootstrap samples that do not contain observation i, and $|C^{-i}|$ the number of such samples, the leave-one-out bootstrap estimate of prediction error is given by:

$$\widehat{\mathrm{Err}}^{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}}^{N}(y_i - f^{*b}(x_i))$$

In computing $\widehat{\mathrm{Err}}^{(1)}$, care should be taken to ensure that the number of bootstrap repetitions, B, is large enough to ensure that all of the $|C^{-i}|$ are greater than zero. Alternatively, all the terms for cases where $|C^{-i}| = 0$ can be left out.

The leave-one-out bootstrap solves the problem of overfitting suffered by $\widehat{\mathrm{Err}}_{\mathrm{boot}}$, but still suffers from the training-set-size bias, as discussed in the section on cross-validation. The probability that observation i is in bootstrap sample, b, is approximated as follows:

$$\begin{aligned}\mathrm{P}(\text{Observation } i \in \text{ bootstrapsample } b) &= 1-(1-\frac{1}{N})^N \\ &\approx 1-e^{-1} \\ &= 0.632\end{aligned}$$

Therefore, the average number of distinct observations in each bootstrap sample will be about 0.632*N. It can therefore be expected that leave-one-out bootstrap will behave roughly like two-fold cross-validation. Thus, if the learning curve exhibits considerable slope at sample size N/2, the leave-one-out bootstrap estimate will be upward biased.

In order to alleviate this problem, the ".632 estimator" can be used:

$$\widehat{\mathrm{Err}}^{(.632)} = .368\cdot\overline{\mathrm{err}} + .632\cdot\widehat{\mathrm{Err}}^{(1)}$$

Intuitively, the .632 estimator reduces the upward bias of $\widehat{\text{Err}}^{(1)}$, by pulling it down towards the training error rate, $\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(x_i))$. $\widehat{\text{Err}}^{(.632)}$ works well in "light fitting" situations, but can break down in overfit ones. (Hastie, Tibshirani & Friedman, 2001)

As example: Suppose we have two equal-sized classes, with the targets independent of the class labels (i.e. class labels are assigned randomly to the objects). Now, if a one-nearest neighbour classification rule is applied, then $\overline{\text{Err}}^{(1)} = 0.5$, $\widehat{\text{Err}}^{(1)} = 0.5$ and therefore, $\widehat{\text{Err}}^{(.632)} = 0.368 * 0 + 0.632 * 0.5 = 0.316$. The true error rate is, however, 0.5.

The .632 estimator can be improved by taking the amount of overfitting into consideration. Firstly, define the no-information error rate, γ. This is the error rate if there was no relation between the dependent and independent variables. An estimate of gamma is obtained by evaluating the fitted model on all possible combination of targets y_i and predictors x_j:

$$\hat{\gamma} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(y_i - \hat{f}(x_j))^2$$

The relative overfitting rate is given by

$$\hat{R} = \frac{\widehat{\text{Err}}^{(1)} - \overline{\text{err}}}{\hat{\gamma} - \overline{\text{err}}}$$

and ranges from 0 if there is no overfitting ($\widehat{\text{Err}}^{(1)} - \overline{\text{err}}$), to 1 if the overfitting equals the no-information value $\hat{\gamma} - \overline{\text{err}}$. Finally, the ".632+" estimator is defined by:

$$\widehat{Err}^{(.632+)} = (1-\hat{\omega})\cdot\overline{err} + \hat{\omega}\cdot\widehat{Err}^{(1)}$$

$$\text{with } \hat{\omega} = \frac{.632}{1-.368\hat{R}}$$

The weight ω ranges from 0.632 (if $\hat{R}$ = 0) to 1 (if $\hat{R}$ = 1). $\widehat{\text{Err}}^{(.632+)}$ therefore ranges from $\widehat{\text{Err}}^{(.632)}$ to $\widehat{\text{Err}}^{(1)}$ and gives a compromise between the leave-one-out bootstrap and the training error rate, depending on the amount of overfitting.

4 Noise addition PLS

With PLS, both the **y** and **X** values are subject to random error. This noise will affect the quality of the model, and the quality of predictions of **y** is expected to decrease with an increase in noise. The robustness of the model can be improved by including stabilisation spectra into the data. This can include taking more samples over time to reflect changing conditions, such as spectra at different temperatures, or by creating artificial variation by using, for example, two grindings for each sample. Dardenne and Fernández Pierna (2006) also quote some work that has been done on noise addition on **X**.

A crucial assumption in multivariate calibration is that the reference values are sufficiently precise and that the accuracy of the NIR model will never be more accurate than the reference method. It was however shown that cases do exist where the predicted value is often more accurate than the reference value. (Dardenne and Fernández Pierna, 2006)

While the various ways described above can help improve the robustness of the calibration model, the accuracy of the model ultimately depends on the quality of the reference values. Dardenne and Fernández Pierna (2006) explore a technique whereby noise is iteratively added to the reference values in order to improve the robustness of PLS calibration models.

A slightly differently stated version of their algorithm is given in **ALGORITHM 4.1**. The algorithm starts by creating three subsets, CAL, VAL and TEST, of the original input data. CAL is used to construct the various PLS models and VAL is used for internal optimisation in aid of model selection. TEST should be kept separate at all cost. It is only used to assess the improvement of the Noise Added PLS model over the conventional PLS model.

The core of the algorithm, Step 7, is repeated a large number of times. Dardenne and Fernández Pierna (2006) suggest using R = 500 iterations, but give no reasoning for their choice. Empirical findings in this paper suggest that, by increasing this to as much as 1000 to 3000 repetitions, better results can be obtained in some situations. It is reasonable to expect that the number of repetitions might depend on the sizes of the different subsets, the level of random error present in the data, as well as the magnitude of the noise added.

ALGORITHM 4.1 *The noise addition PLS algorithm*.

1. *Split the input data into 3 subsets: CAL = [$\mathbf{X}_{cal}$, y_{cal}], VAL = [$\mathbf{X}_{val}$, $\mathbf{y}_{val}$] and TEST = [$\mathbf{X}_{test}$, $\mathbf{y}_{test}$].*
2. *Let $f_{Cal}^{(0)}$ be the PLS model constructed using the CAL subset.*
3. *Use $f_{Cal}^{(0)}$ to predict the VAL and TEST subsets. Use these predictions to calculate $RMSE_{Val}^{(0)}$, $R_{Val}^{2\ (0)}$, $RMSE_{Test}^{(0)}$, $R_{Test}^{2\ (0)}$*
4. *Repeat for r_outer = 1..R_outer repetitions:*
5. *Noise_Level = 0*
   $\tilde{\mathbf{y}}_{\text{cal}} = \mathbf{y}_{\text{cal}}$
   $RMSE_{Val}^{(Begin)} = 100$
6. *Repeat for r_inner = 1..R_inner repetitions*
7. *Add 10% noise to CAL($\mathbf{y}_{\text{cal}}$):*
   $\tilde{\mathbf{y}}^{*}_{\text{cal}} = \tilde{\mathbf{y}}_{\text{cal}}$ + RandNorm(mu=0, sigma = 0.1* $\bar{y}_{\text{cal}}$, k) * 0.9^ Noise_Level,
   with k the number of observations in $\mathbf{y}_{\text{cal}}$.
8. *Let $f_{Cal}^{(New)}$ be the PLS model constructed using the (noisy) CAL subset, $\tilde{\mathbf{y}}^{*}_{\text{cal}}$*
   Use $f_{Cal}^{(New)}$ to predict the VAL subset. Use these predictions to calculate $RMSE_{Val}^{(New)}$
9. *If $RMSE_{Val}^{(New)} < RMSE_{Val}^{(Begin)}$ (I.e. noise addition improved the model), store the error and the noisy calibration set; reduce the level of noise:*
   $\tilde{\mathbf{y}}_{\text{cal}} = \tilde{\mathbf{y}}^{*}_{\text{cal}}$
   $RMSE_{Val}^{(Begin)} = RMSE_{Val}^{(New)}$
   $Noise\_Level = Noise\_Level + 1$
10. *Store model coefficients for iteration r (irrespective of effect of Step 9): $f_{Cal}^{(r)} = f_{Cal}^{(New)}$ where r = (r_outer-1)*R_outer + r_inner*
11. *Goto Step 6*
12. *Goto Step 4*
13. *Compute the median of coefficient vectors of the R = R_outer*R_inner models.*
14. *Select f_{Cal}^{*}, the best model, as the model with the most correlated coefficients with the median of the coefficients.*
15. *Use f_{Cal}^{*} to predict the TEST subset ($\mathbf{y}_{test}$). Using these predictions, compute $RMSE_{Test}^{*}$ / $R_{Test}^{2\ *}$.*

The algorithm works by means of a nested loop structure. The first (outer) loop (Step 4) is a repetition of the entire noise addition process. Each repetition of this loop starts out with a copy of the original calibration dataset and level of noise.

The actual noise addition process is done in the second (inner) loop (Step 6). For each repetition of this loop, noise is added to the calibration dataset. Thereafter, a PLS model is trained and the goodness of fit of the model is assessed by prediction the validation set. If noise addition improved the quality of the model, the “noisy” calibration data is saved and the level of noise is decreased.

Lastly, the coefficients of the current PLS model is saved. It should be noted that the coefficients of the current model is saved, regardless of whether or not noise addition managed to improve the quality of the model.

After R repetitions of noise addition, the optimal model is defined as the model that has the coefficients that are the highest correlated with the median of all the coefficients. Other statistics, such as the mean or trimmed mean, can also be used, but the median is simple to compute and is known not to be influenced by outliers. Dardenne and Fernández Pierna (2006) suggest doing R = 500 iterations of noise addition. This is accomplished by doing R_{outer} = 5 outer loops and R_{inner} = 100 inner loops.

Finally, the optimal model is used to predict the TEST subset. This subset is able to give an unbiased estimate of the true test error as it was never used to build and select the final model. This test error is then compared to the test error of the conventional PLS model. If noise was present in the original reference values, it is expected that the test error obtained with Noise Addition PLS should be lower than the test error of the conventional PLS model.

5 Application of PLS to Practical Datasets

5.1 Dataset 1

Dataset 1 consists of 129 observations, 46 independent variables and one dependent variable. For modelling purposes, the input dataset was split into two different subsets - a training set to train the PLS model and a testing set to assess the model's predictive ability. A total of 104 observations were allocated to the training set, while the remaining 25 observations were used for the test set.

FIGURE 5.1 provides a visual presentation of the spectra for Dataset 1. The spectra vary within a fairly small range and "flattens off" just after the second half.

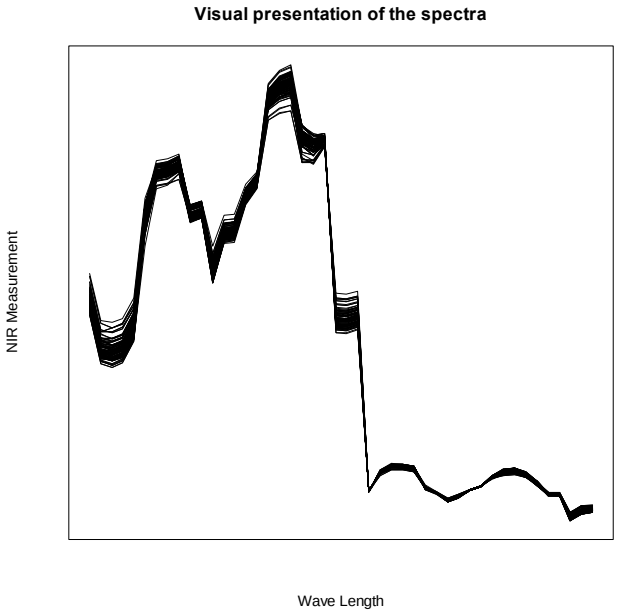

FIGURE 5.1 *Visual presentation of the spectra of Dataset 1.*

In order to detect the presence of outliers and possible subgroups in the data, the T-score vectors (representing the X space) can be plotted against each other in a scatter plot. The TT plot is similar to normal PCA plot, but uses the scores of the PLS components, in the place of the scores of the principal components.

The TT plot for Dataset 1 is given in **FIGURE 5.2**. The first thing that can be noticed from this plot is the presence of replicates – almost each observation has an almost identical "twin". (It should be noted that only the training data were used to construct this plot, so there will be a few cases that do not have "twins".) Replicates are usually used in order to determine or reduce the effect of external or unwanted factors. In this case, the replicates tended to lie close to each other, indicating that there was little disturbance in the NIR measurement process. This could explain why the spectra in **FIGURE 5.1** vary in such a narrow band.

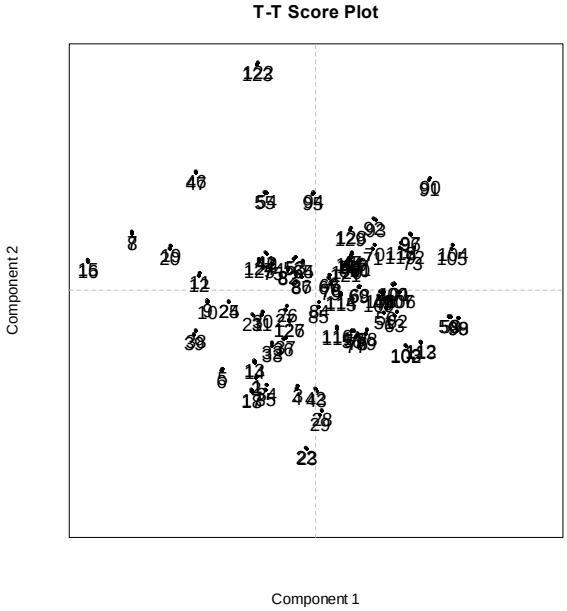

FIGURE 5.2. *The T-scores for component 1 and component 2.*

Observations 122 and 123 (top of **FIGURE 5.2**) appear to be removed from the bulk of the observations and might influence the fit of the final PLS model. The amount of influence can be quantified by calculating the leverage for each point. This is discussed in more detail shortly.

FIGURE 5.3 gives the cross-validation error as well as the percentage of the total variance explained by each component (only the first ten components are shown). The cross-validation (left panel) error tapers off after seven components. However, the improvement in error is negligible after four components. According to the scree plot (right panel), about 90% of the total variance is explained by the first three components. It is interesting to note that the scree plot displays a spike between components 4 and 6. This could be caused by components, with significant variances in the X-space, which are not too closely correlated with the Y-space. PLS modelling was performed by retaining four components.

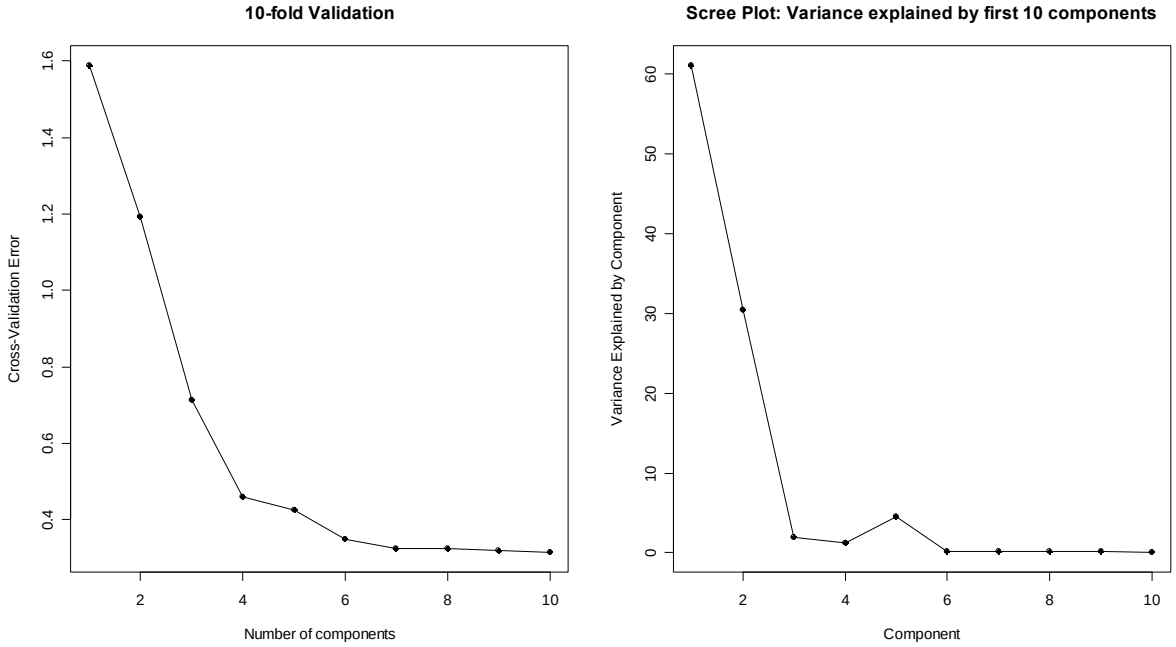

FIGURE 5.3 *The cross-validation error and a scree plot for Dataset 1.*

The relationship between the input spectra and the target variable (known chemical composition) can be displayed graphically by plotting the T-scores, of the X-space, against the U-scores of the Y-space, one component at a time. This plot is known as the T vs. U plot and is shown in **FIGURE 5.4**.

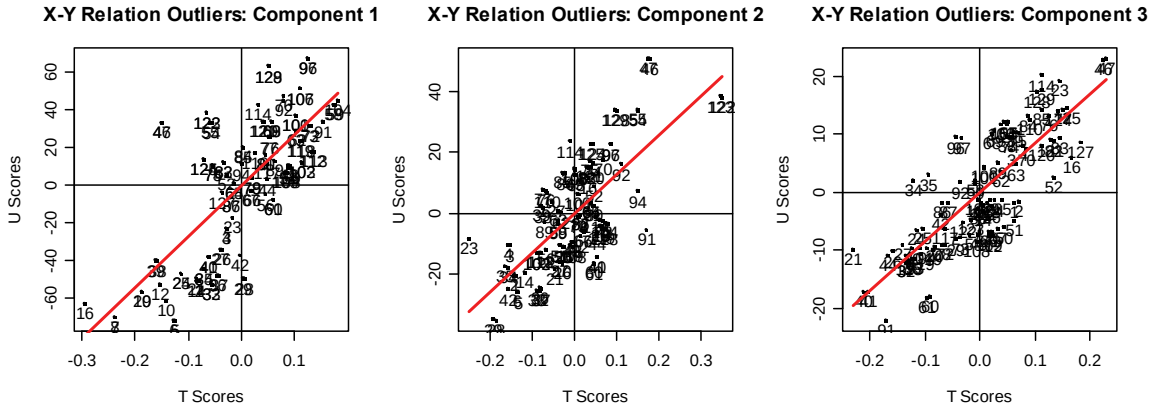

FIGURE 5.4 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components.*

From **FIGURE 5.4**, it seems as if there is a fairly linear relationship between the T- and U-scores.

The leverage of a point measures the effect of that point on the final fit of the model, and is, to a large extent, related to the point's distance from the model centre. The leverage is always scaled so that it will have a value between 0 and 1. An extreme value will have a high leverage (close to 1); while a normal value will have a small leverage (closer to 0).

The leverage is calculated as follows:

$$h_i = \frac{1}{n} + \sum_{a=1}^{A} \frac{t_{ia}^2}{\mathbf{t}_a^T \mathbf{t}_a}$$

where

A = number of components calculated

t_{ia} = the score value for each object i in component a

h_i = leverage for object i

The leverage for the 5 most influential values for Dataset 1 is shown in the following table:

Observation	91	16	122	123	28
Leverage	0.2117	0.1603	0.1418	0.1389	0.1384

It is reasonable to assume that the leverage for each observation will increase as the size of the training dataset decreases. By looking at the above mentioned table, as well as **FIGURE 5.4**, it does not seem as if there are any significant influential values present in Dataset 1. A formal discussion on leverage detection is however, beyond the scope of this paper.

The final test for Dataset 1 is to assess how well it can predict new data. This was done by using the 104 values of the training set to construct a model consisting of four retained components. This model was then used to predict the 25 values of the test set. The following fit statistics were calculated: Root Mean Square Error (RMSE), the squared correlation coefficient (R^2) and the Coefficient of determination for predicted values (R^2_{pred}).

Statistic	**Value**
RMSE	0.4994
R^2	0.9605
R^2_{pred}	0.9429

From the above statistics, it seems as if the model fits reasonably well – the predicted values are highly correlated with the actual values. Also, by looking at the R^2_{pred} statistic, the model is able to explain 94.29 % of the variation of the test data. This confirms the results of **FIGURE 5.4** (T- vs U plot) that clearly show the existence of a linear relationship between the dependent variable and the NIR measurements.

5.2 Dataset 2

Dataset 2 consists of 197 observations, 1557 independent variables and one dependent variable. One hundred and fifty observations were allocated to the training set and the remaining 47 observations were kept separate for testing purposes.

The TT plot for Dataset 2 is given in **FIGURE 5.5**. Observations 67, 163 and 164 (top and right of **FIGURE 5.5**) appear to be removed from the bulk of the observations and might influence the fit of the final PLS model.

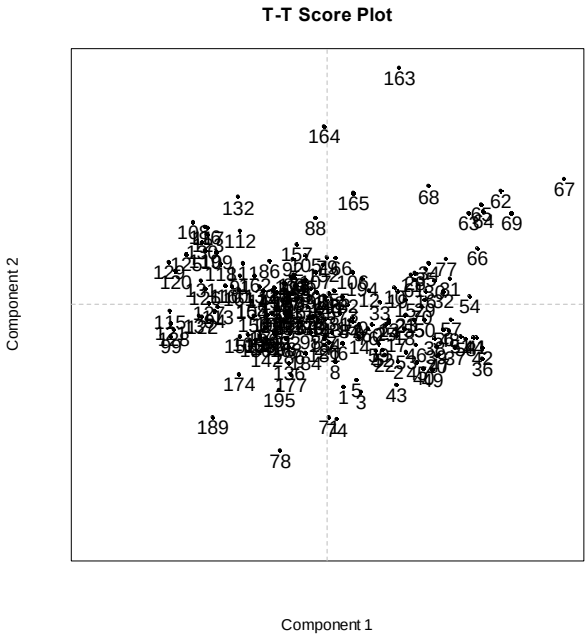

FIGURE 5.5 *The T-scores for component 1 and component 2.*

FIGURE 5.6 gives the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error tapers off after 21 components, but exhibits irregular behaviour. As shown in **FIGURE 3.1**, the cross validation error is expected to decrease monotonically until the optimal level of model complexity is reached. However, the cross validation error for Dataset 2 shows the presence of a few spikes.

According to the scree plot (right panel), almost all the variance is explained by the first three components. A total of 21 components were retained for PLS modelling.

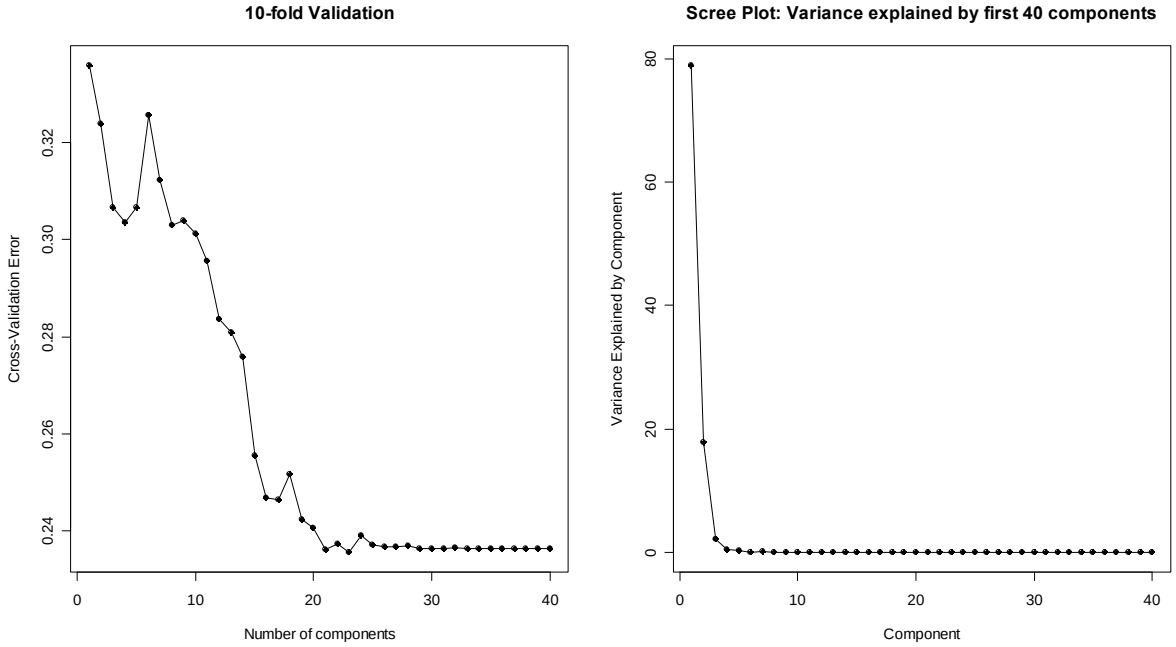

FIGURE 5.6 *The cross-validation error and a scree plot for Dataset 2.*

The relationship between the input spectra and the target variable (known chemical composition) is shown in **FIGURE 5.7**.

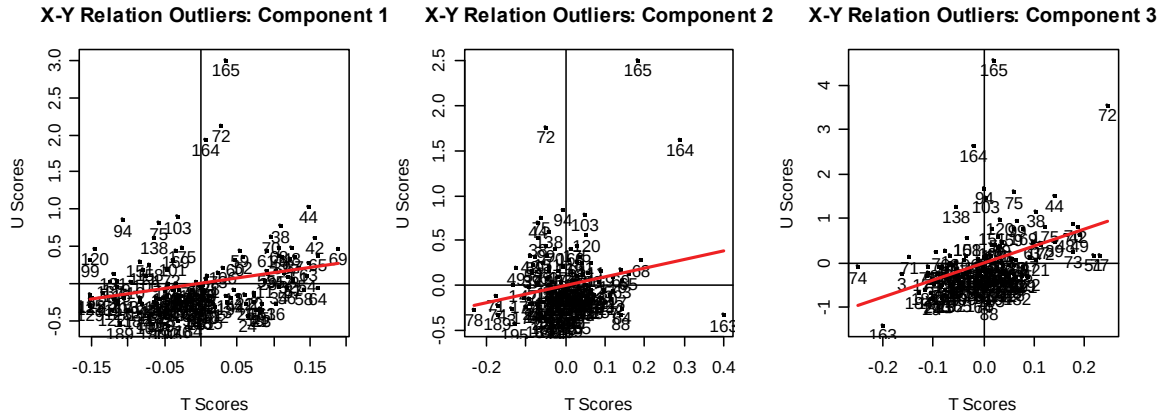

FIGURE 5.7 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components.*

From **FIGURE 5.7** there is no clear evidence of a strong linear relationship between the T- and U scores. Also, it is quite evident that there are some severe outliers. Observations 72, 163, 164 and 165 consistently plot far from the swarm of points. This is also confirmed by looking at the leverage for the five most influential values:

Observation	163	165	27	72	77
Leverage	0.6497	0.6292	0.6259	0.4687	0.3742

The above mentioned table clearly shows the presence of severe outliers that significantly impacts the model's fit.

Finally the model's predictive ability is assessed by predicting the 47 observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	Value
RMSE	0.2050
R^2	0.6201
R^2_{pred}	0.5939

The fit statistics confirm the conclusions drawn from **FIGURE 5.7** – the model does not appear to fit well, and as a result, does not perform well at predicting new values.

It may be possible to improve the model's fit by removing some of the outliers that have a severe negative influence on the fit of the model. Leverage correction should be done by removing one observation at a time and recalculating the leverage of the remaining observations. This iterative process needs to be followed because the removal a single observation can influence the fit of a model to such an extent that observations that were previously deemed non-influential are now influential. Likewise, observations that were previously influential might become non-influential.

TABLE 5.1 *6 Iterations of leverage correction.*

Iteration		Observations in order of influence									
1	Observation	163	165	27	72	77	74	164	70	111	78
	Leverage	0.6497	0.6292	0.6259	0.4687	0.3742	0.3679	0.3470	0.3279	0.3096	0.2782
2	Observation	27	165	164	72	74	77	70	111	78	189
	Leverage	0.6380	0.6162	0.4335	0.4174	0.4093	0.3632	0.3200	0.2898	0.2810	0.2459
3	Observation	165	72	164	74	77	70	111	78	142	189
	Leverage	0.6087	0.4498	0.4493	0.4012	0.3661	0.3198	0.2980	0.2751	0.2606	0.2571
4	Observation	164	72	74	77	70	111	142	189	78	100
	Leverage	0.6730	0.4872	0.4083	0.3741	0.3358	0.3038	0.2512	0.2493	0.2387	0.2385
5	Observation	72	74	70	77	111	142	78	189	51	71
	Leverage	0.5223	0.4227	0.3688	0.3263	0.3057	0.2687	0.2682	0.2645	0.2623	0.2443
6	Observation	74	77	70	142	78	111	51	79	189	103
	Leverage	0.4415	0.4111	0.3823	0.2943	0.2739	0.2712	0.2639	0.2629	0.2589	0.2557

This can be observed from the results of iterative leverage correction in **TABLE 5.1**. The five most influential values are (in order of leverage) 163, 165, 27, 72 and 77. However, when the five most influential observations were removed using an iterative approach, observations 163, 27, 165, 164 and 72 were removed (in order of appearance). This clearly shows that the observations that were

removed, as well as the order in which they were removed, differed significantly from the original five most influential values.

FIGURE 5.8 shows the cross validation error and scree plot after the five most influential values were removed. The cross validation error is still a bit erratic, but is better behaved than previously. It reaches is minimum at 21 components.

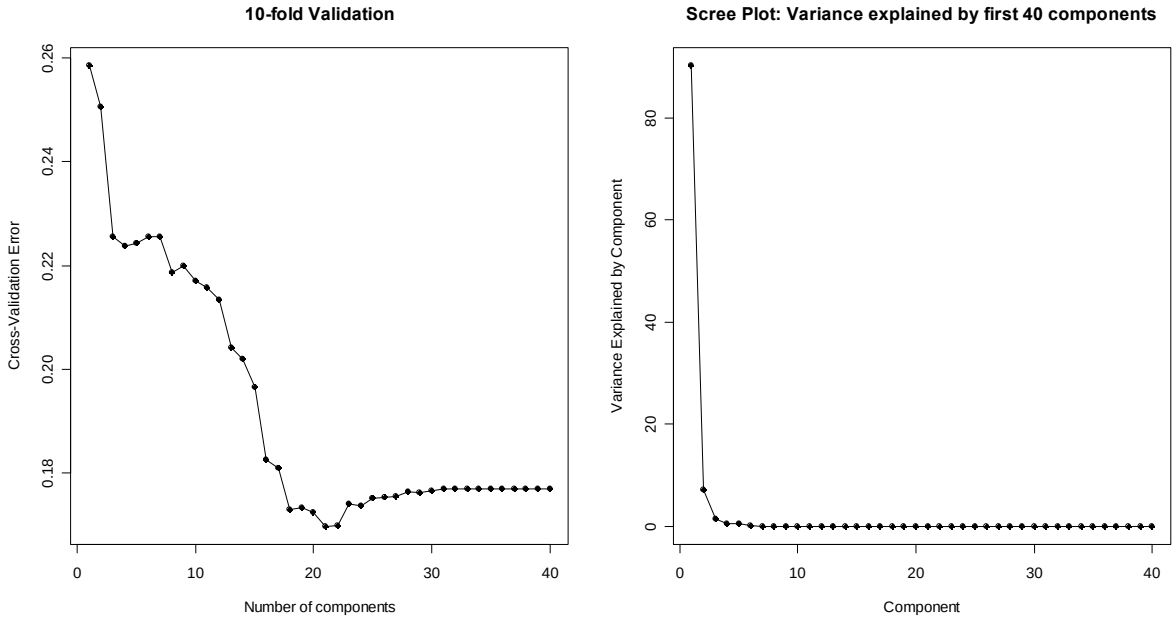

FIGURE 5.8 *The cross-validation error and a scree plot for the leverage corrected version of Dataset 2.*

Finally, the impact of removing the influential values is assessed. **TABLE 5.2** gives the results of successively removing the most influential value, refitting the model and re-assessing the model's fit by predicting the test set.

Initially, the model's fit improves marginally, as shown by all three statistics in **TABLE 5.2**. After the removal of observation(s) 164 (and 72), the quality of the model deteriorates. This shows that information can be lost if the data is over-corrected.

TABLE 5.2 *The effect of leverage correction on the model's fit, measured by predicting the independent test set.*

	Observation Removed					
Statistic	**None**	**163**	**27**	**165**	**164**	**72**
RMSE	0.2050	0.2031	0.2007	0.1971	0.2171	0.2157
R^2	0.6201	0.6273	0.6425	0.6409	0.5643	0.5666
R^2_{pred}	0.5939	0.6015	0.6109	0.6247	0.5444	0.5504

5.3 Dataset 3

Dataset 3 consists of 102 (103 for Dataset 3A) observations, 1557 independent variables and three dependent variables. The training set was allocated 90 observations and twelve (thirteen for Dataset 3A) observations were kept separate for testing purposes. PLS modelling was done separately for each of the dependent variables.

The TT plot, for Dataset 3A is shown in **FIGURE 5.9** below. The prominent formation of two distinct subgroups of observations can be noticed clearly. This could be due to the fact that NIR scanning was performed at two distinct levels of the constituent of interest, or that the researchers tried to quantify the effect of an external factor (e.g. temperature) by measuring NIR reflectance at two (or possibly more) different levels of the external factor.

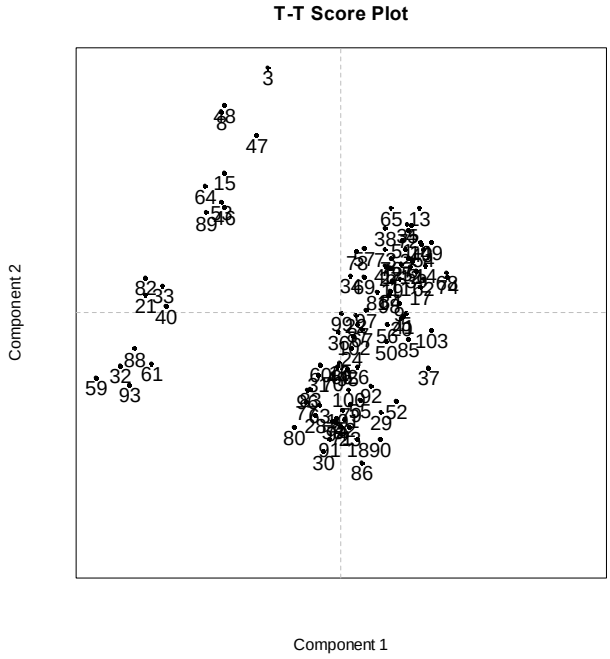

FIGURE 5.9 *The T-scores for component 1 and component 2 of Dataset 3A.*

The TT plot, for Dataset 3B is shown in **FIGURE 5.10** below. Similar to Dataset 3A, the presence of two distinct subgroups is evident. For future studies, it might be worthwhile to investigate why these subgroups exist, as well as their effect on the fit of the model.

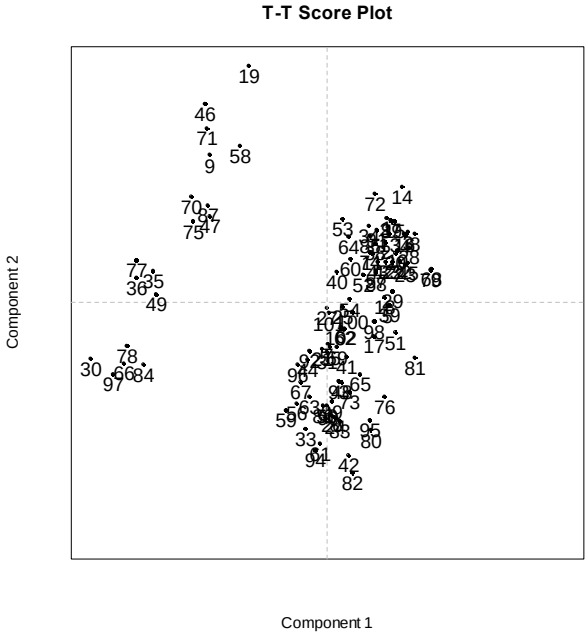

FIGURE 5.10 *The T-scores for component 1 and component 2 of Dataset 3B.*

The TT plot, for Dataset 3C is given in **FIGURE 5.11** below. Similar to Datasets 3A and 3B, two distinct subgroups can be observed.

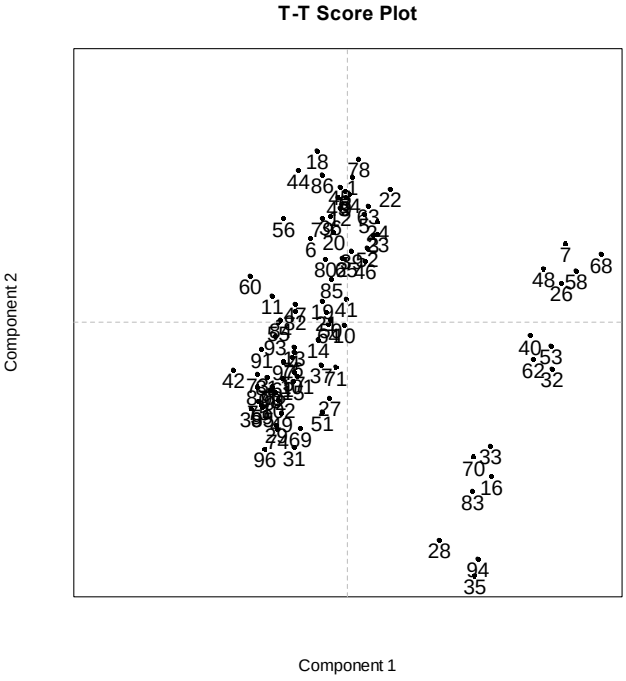

FIGURE 5.11 *The T-scores for component 1 and component 2 of Dataset 3C*

The modelling process will now be discussed separately for each of the three dependent variables.

5.3.1 Dataset 3A

FIGURE 5.12 gives the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a minimum at eleven components. The general shape of the error curve is the same as the theoretical training error curve that is shown in **FIGURE 3.1**.

According to the scree plot (right panel), almost all the variance is explained by the first component. Little variance is explained by the second component, while the remaining components are insignificant in terms of the amount of variance that they explain. PLS modelling was done by retaining eleven components.

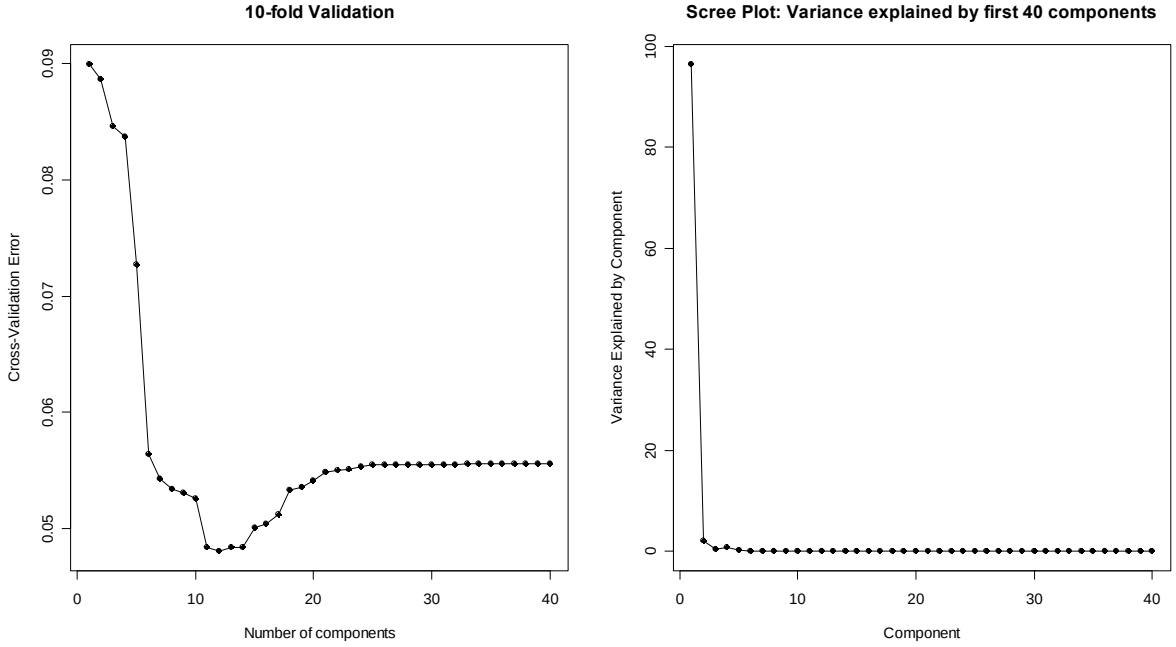

FIGURE 5.12 *The cross-validation error and a scree plot for Dataset 3A.*

FIGURE 5.13 shows the relationship between the input spectra and the target variable (known chemical composition).

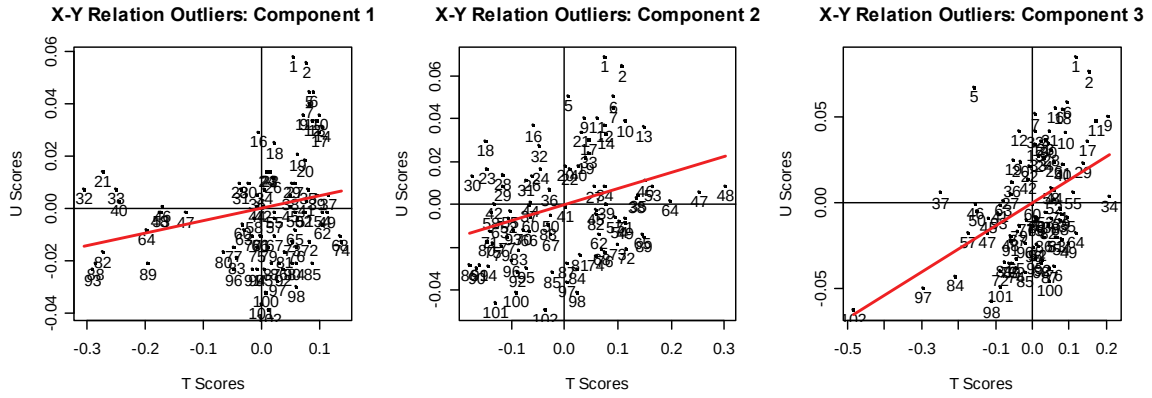

FIGURE 5.13 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components.*

There is no clear evidence of a strong linear relationship between the T- and U scores. In the first pane of **FIGURE 5.13**, there appears to be a small group of points that severely distorts the relationship between the T- and U scores. This group appears to be similar to the subgroup that was observed in **FIGURE 5.9**. For easy comparison, **FIGURE 5.14** shows the T-T score plot next to the T- vs. U plot for the first PLS component:

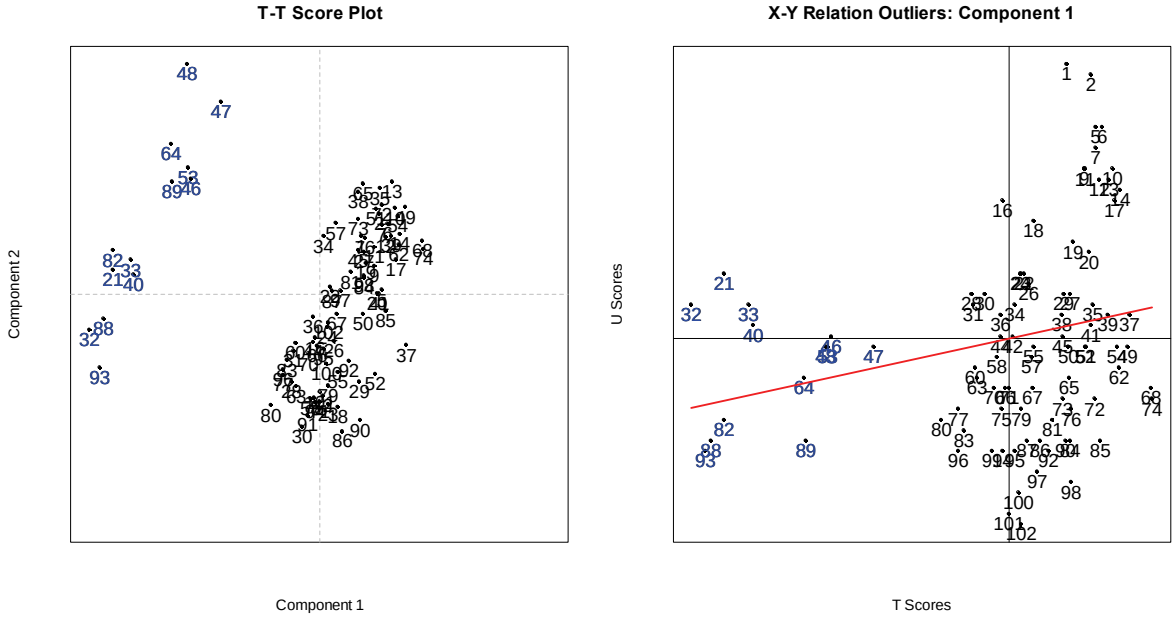

FIGURE 5.14 *A side by side comparison of the subgroups observed in the T-T score plot and the T- vs. U plot.*

The left pane of **FIGURE 5.14** shows the small subgroup that was first observed in **FIGURE 5.9**. These observations were colour coded in order to assist comparison. The right panel of **FIGURE 5.14** highlights these same points in the T-vs.-U plot. It is evident that there are two distinct groups present in Dataset 3A. It appears as if there is a reasonable linear relationship between X and Y for the larger of the two groups (right hand side of T- vs. U plot in **FIGURE 5.14**). The overall relationship between X and Y is

however severely distorted due to the presence of the smaller group on the left hand side of the T – vs. U plot.

The five most influential observations are given below. It does seem as if observation 102 has a reasonably high leverage. This should however, be seen in context with the fact the training dataset is fairly small and that there are two subgroups present in the data.

Observation	102	51	13	32	46
Leverage	0.4118	0.3245	0.3187	0.2735	0.2721

Finally the model's predictive ability is assessed by predicting the thirteen observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	**Value**
RMSE	0.0452
R^2	0.9243
R^2_{pred}	0.8828

The fit statistics show that the eleven components of the fitted model perform reasonably well at predicting new values. The RMSE is quite small, but could be misleading due to the measurement scale of the dependent variable. However, both the R^2 and R^2_{pred} statistics are scale invariant and show that the fitted model performs reasonably well.

5.3.2 Dataset 3B

FIGURE 5.15 gives the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a minimum at nine components

According to the scree plot (right panel), almost all the variance is explained by the first component. Little variance is explained by the second component, while the remaining components are insignificant in terms of the amount of variance that they explain. PLS modelling was done by retaining nine components.

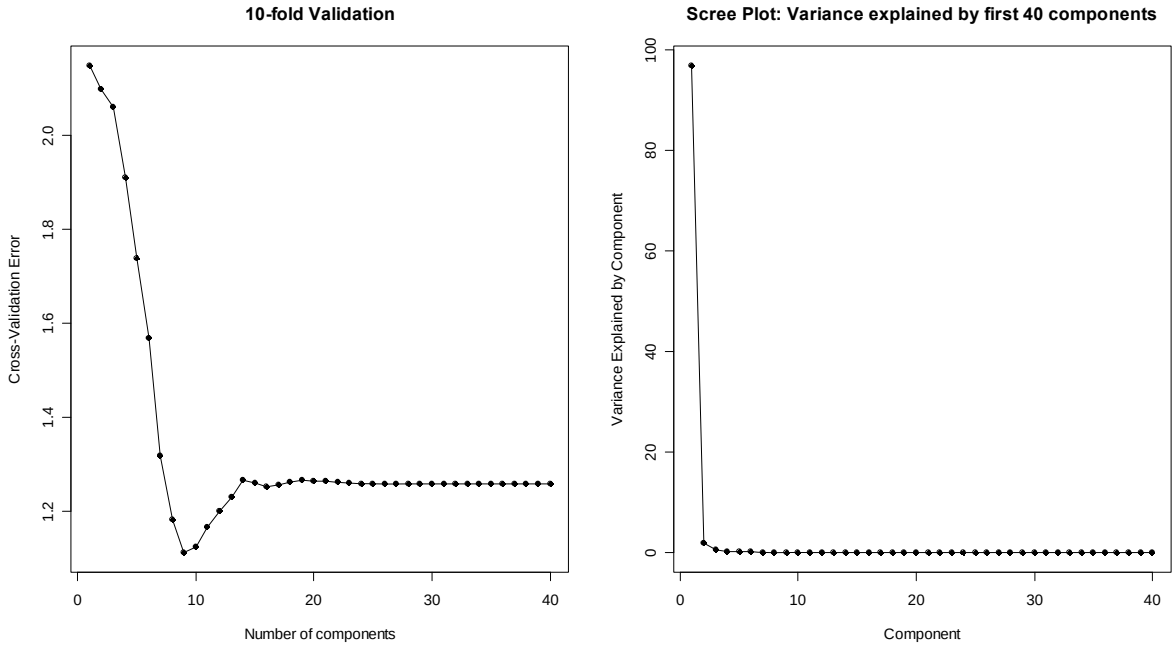

FIGURE 5.15 *The cross-validation error and a scree plot for Dataset 3B.*

The relationship between the input spectra and the target variable (known chemical composition) is shown in **FIGURE 5.16**.

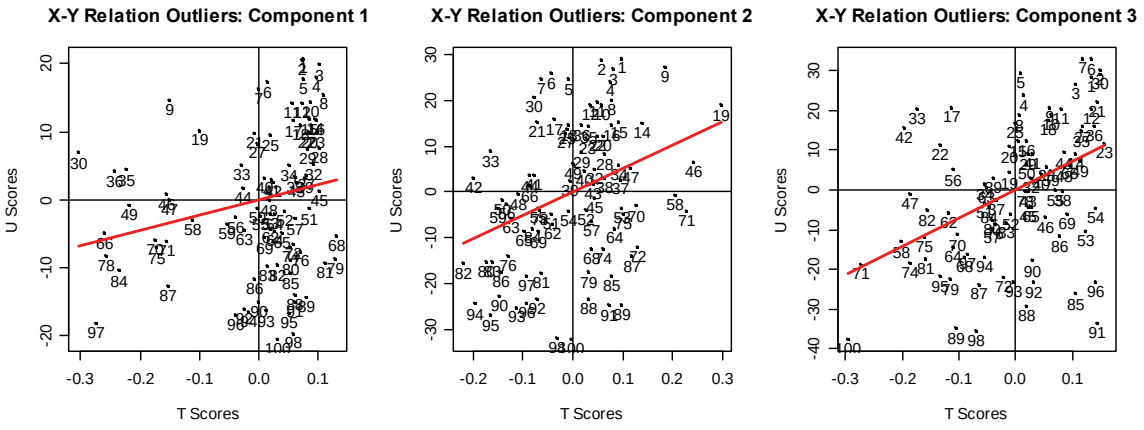

FIGURE 5.16 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components.*

There is no clear evidence of a strong linear relationship between the T- and U scores in any of the three panes of **FIGURE 5.16**. In order to investigate the subgroup that was observed in **FIGURE 5.10**, the T-T plot is graphed next to the T- vs. U plot in **FIGURE 5.17**.

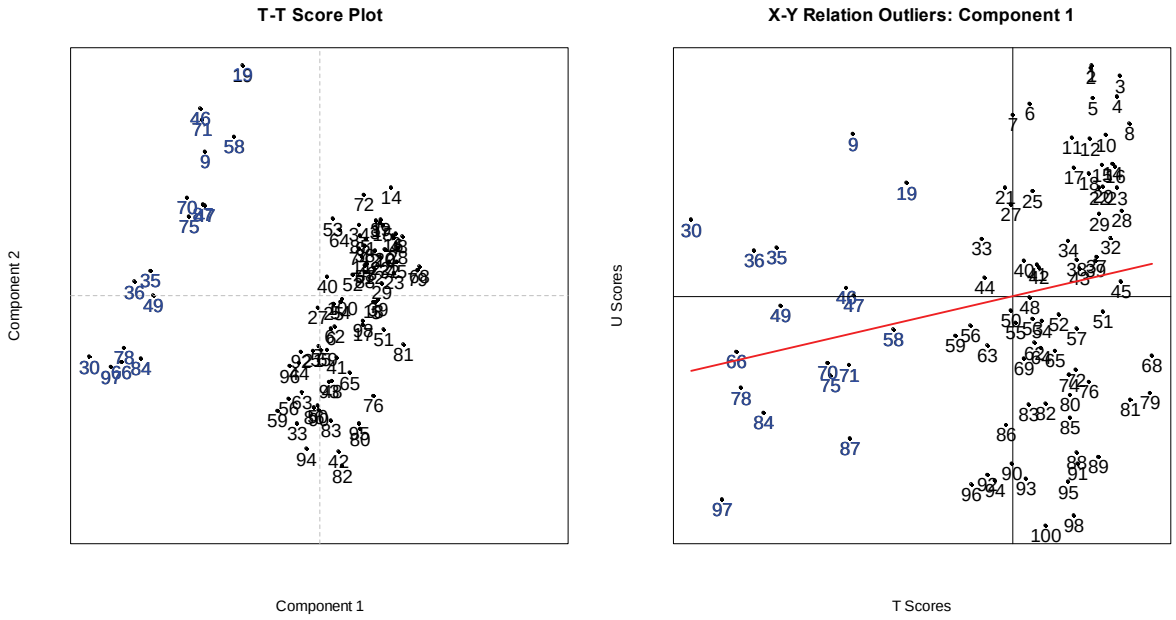

FIGURE 5.17 *A side by side comparison of the subgroups observed in the T-T score plot and the T- vs. U plot.*

The subgroup that was observed in the T-T score plot in **FIGURE 5.10**, is colour coded and shown in the T- vs. U plot in the right pane of **FIGURE 5.17**. As was noted previously with Dataset 3A, in **FIGURE 5.14**, the observations of the subgroup severely distorts the linear relationship between X and Y.

The five most influential observations are given below. The leverage of the five most influential points is stable and does not seem extraordinary large. Leverage does not seem to pose a problem here.

Observation	100	30	81	70	9
Leverage	0.2539	0.2453	0.2369	0.2229	0.2189

Finally the model's predictive ability is assessed by predicting the twelve observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	**Value**
RMSE	0.9037
R^2	0.8861
R^2_{pred}	0.8672

The fit statistics, specifically R^2 and R^2_{pred} show that the nine components of the fitted PLS model performs reasonably well at predicting new values.

5.3.3 Dataset 3C

FIGURE 5.18 shows the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error is slightly jagged between the third and fourth component and reaches a minimum at eleven components.

According to the scree plot (right panel), almost all the variance is explained by the first component. Little variance is explained by the second component, while the remaining components are insignificant in terms of the amount of variance that they explain. Eleven components were retained for modelling purposes.

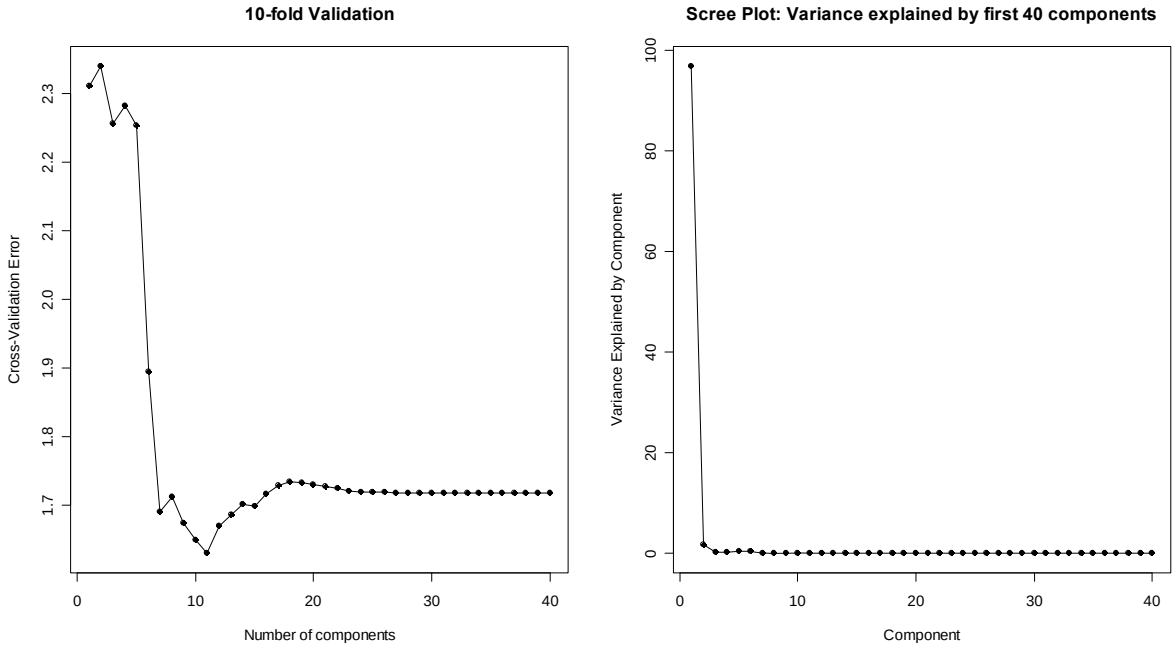

FIGURE 5.18 *The cross-validation error and a scree plot for Dataset 3C.*

FIGURE 5.19 shows the relationship between the input spectra and the target variable (known chemical composition).

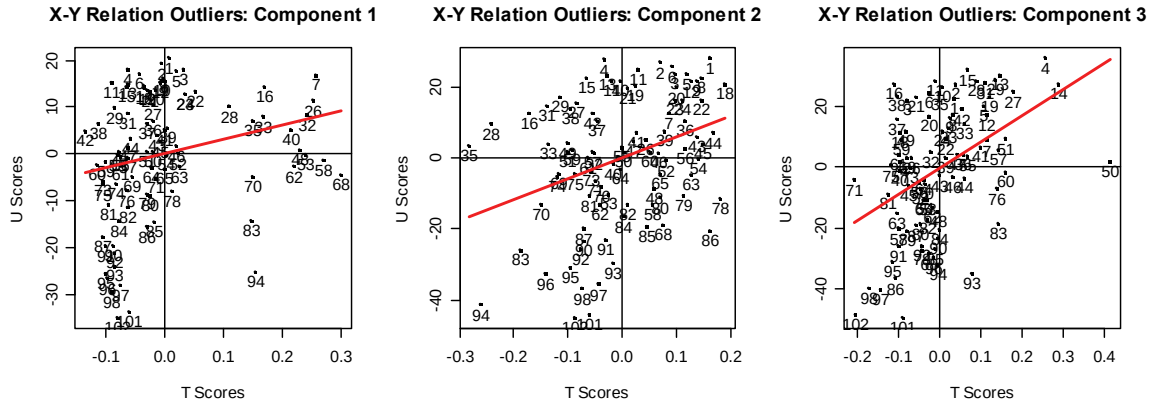

FIGURE 5.19 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components*

There is no clear evidence of a strong linear relationship between the T- and U scores. On the right hand side of the first pane of **FIGURE 5.19**, there seems to be a subgroup of points that distort the relationship between the T- and U scores. As was noted previously with Dataset 3A and 3B, this subgroup appears to be similar to the one found in the T-T score plot (**FIGURE 5.11**). For effortless comparison, **FIGURE 5.20** shows the T-T score plot next to the T- vs. U plot for the first PLS component:

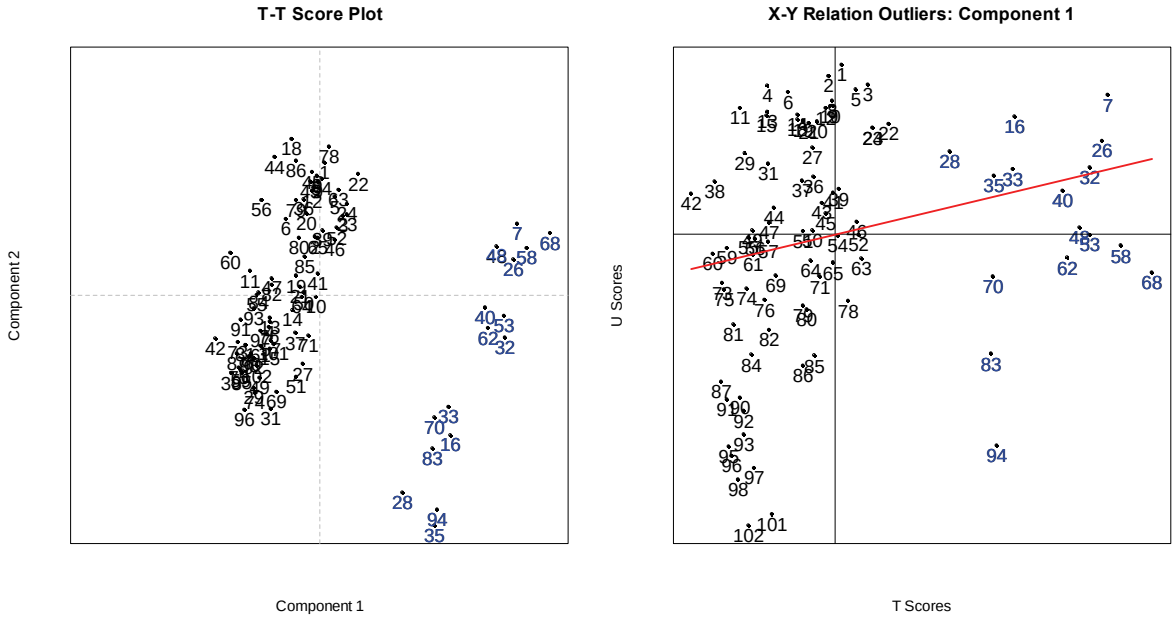

FIGURE 5.20 *A side by side comparison of the subgroups observed in the T-T score plot and the T- vs. U plot.*

The subgroup that was observed in the T-T score plot in **FIGURE 5.11**, is colour coded and shown in the T- vs. U plot in the right pane of **FIGURE 5.20**. As noted earlier with Dataset 3A and Dataset 3B in **FIGURE 5.14** and **FIGURE 5.17**, respectively, the observations of the subgroup severely distorts the linear relationship between X and Y.

The five most influential observations are given below. The leverage of the five most influential points is stable and does not seem extraordinary large. Leverage does not appear to pose a problem here.

Observation	50	83	96	94	35
Leverage	0.4126	0.3050	0.2904	0.2650	0.2644

Finally the model's predictive ability is assessed by predicting the twelve observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	**Value**
RMSE	0.9687
R^2	0.9021
R^2_{pred}	0.8733

The fit statistics, specifically R^2 and R^2_{pred}, show that the eleven components of the fitted PLS model performs reasonably well at predicting new values.

5.4 Dataset 4

Dataset 4 consists of 237 observations, 279 independent variables and one dependent variable. One hundred and ninety observations were allocated to the training set and the remaining 47 observations were kept separate for testing purposes.

The TT plot for Dataset 4 is given in **FIGURE 5.21**. A few of the observations are removed from the bulk of the observations and are scattered to the right of **FIGURE 5.21**. These observations have larger scores for component 1.

FIGURE 5.22 gives the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a minimum at 17 components.

According to the scree plot (right panel), almost all the variance is explained by the first two components. Little variance is explained by the third and fourth components, while the remaining components are insignificant in terms of the amount of variance that they explain. PLS modelling was performed by retaining 17 components.

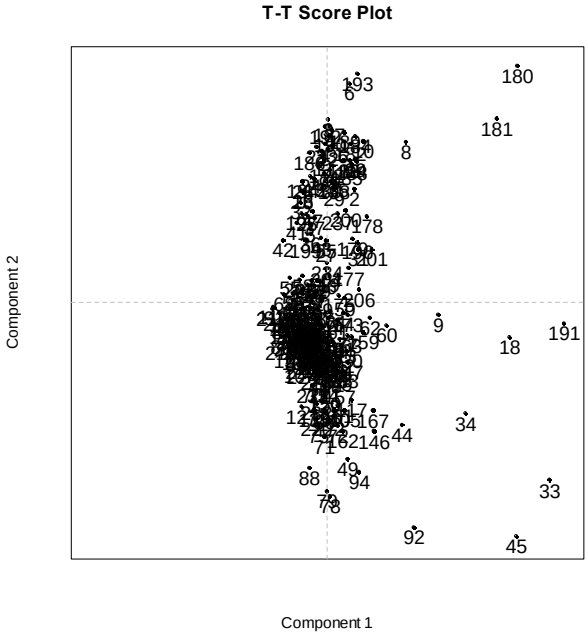

FIGURE 5.21 *The T-scores for component 1 and component 2.*

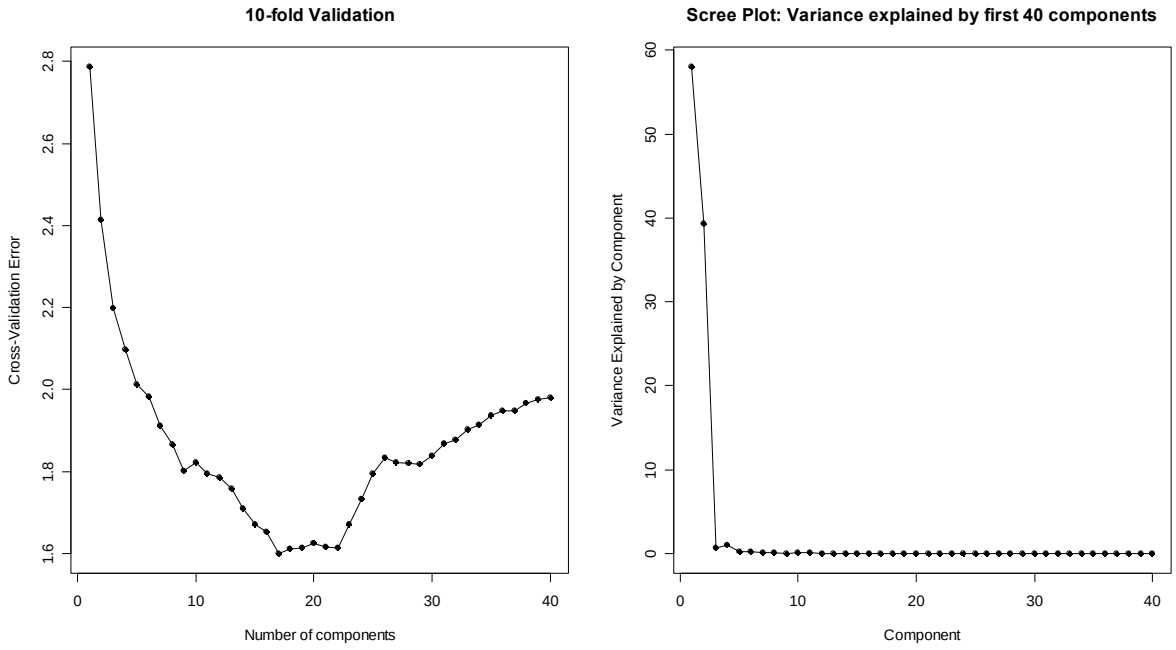

FIGURE 5.22 *The cross-validation error and a scree plot for Dataset 4.*

The relationship between the input spectra and the target variable (known chemical composition) is shown in **FIGURE 5.23**.

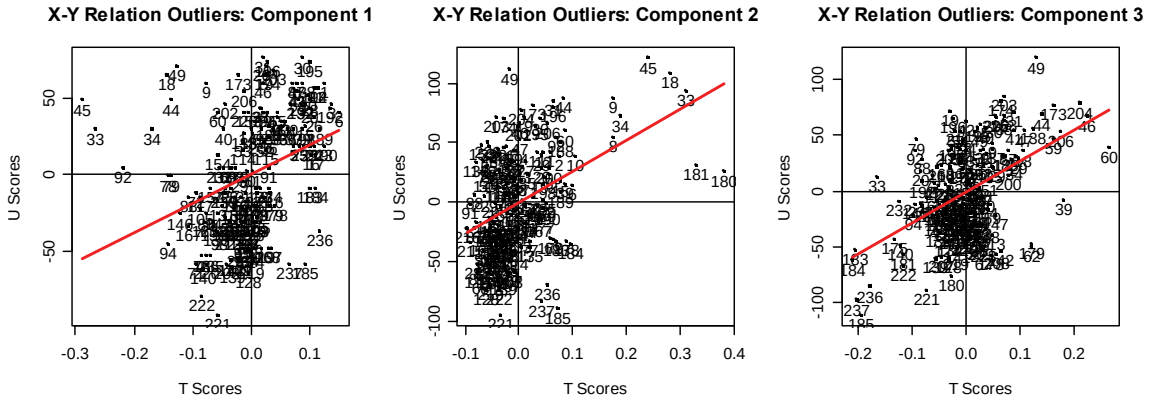

FIGURE 5.23 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components*

There appears to be evidence of a linear relationship between the T- and U scores. However, due to the presence of some outliers, this relationship appears to be distorted to some degree.

The five most influential observations are given below. The leverage for the five most influential points is quite large when compared to the average leverage of 0.0947. Leverage seems to be of concern here.

Observation	18	181	180	33	45
Leverage	0.5568	0.5396	0.5330	0.5001	0.4329

Finally the model's predictive ability is assessed by predicting the 47 observations that were kept separate. The RMSE, R^2 and PRESS statistics were calculated:

Statistic	Value
RMSE	1.3114
R^2	0.7752
R^2_{pred}	0.7546

The fit statistics show that the 17 components of the fitted PLS model performs reasonably well at predicting new values.

5.5 Dataset 5

Dataset 5 consists of 2445 observations, 279 independent variables and one dependent variable. The training set was allocated 400 observations, while testing was performed using 2245 observations.

The TT plot for Dataset 5 is given in **FIGURE 5.24**. The majority of the points plot in a swarm around the centre of the graph. A few of the observations are removed from the bulk of the observations.

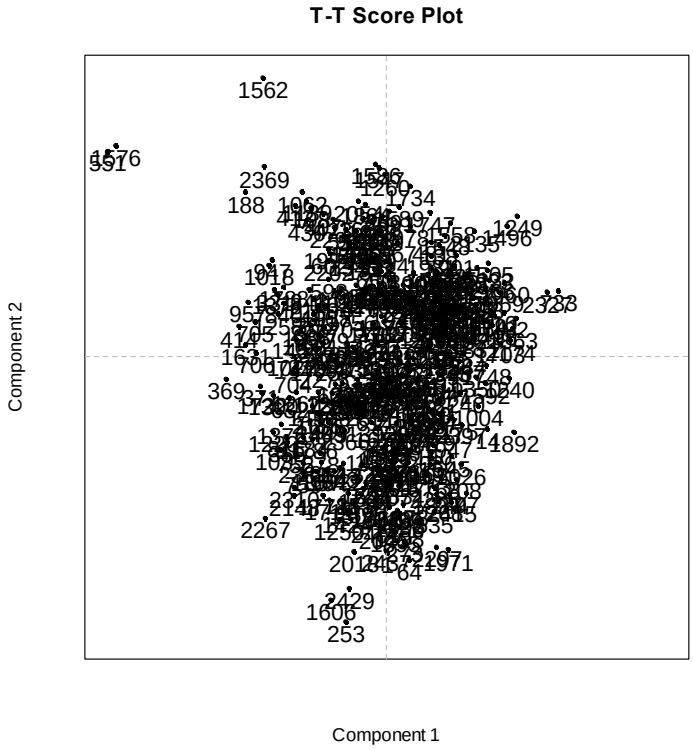

FIGURE 5.24 *The T-scores for component 1 and component 2.*

FIGURE 5.25 provides the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a minimum at thirteen components.

According to the scree plot (right panel), most of the variance is explained by the first two components. A small proportion of the variance is explained by components 3 to 7, while the remaining components are insignificant in terms of the amount of variance that they explain. Thirteen components were retained for modelling purposes.

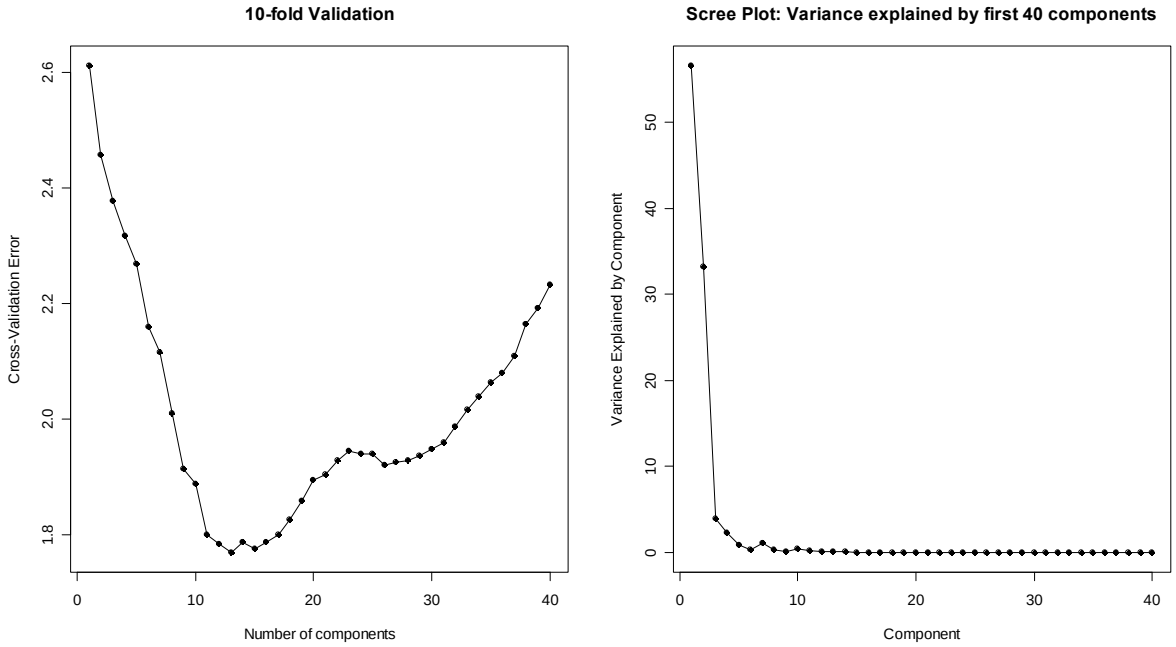

FIGURE 5.25 *The cross-validation error and a scree plot for Dataset 5.*

FIGURE 5.26 shows the relationship between the input spectra and the target variable (known chemical composition).

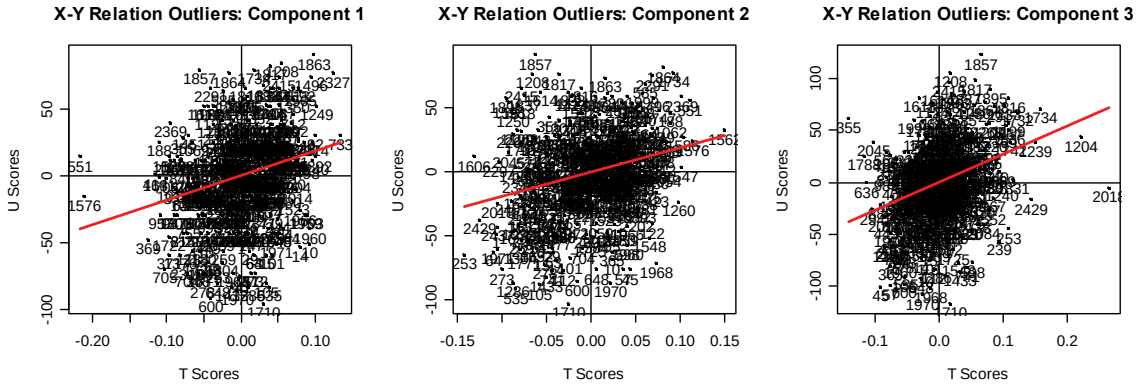

FIGURE 5.26 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components*

There does not appear to be strong evidence of a linear relationship between the T- and U scores. This relationship is further distorted by the presence of a few outliers.

The five most influential observations are given below. The leverage for the five most influential points is quite large when compared to the average leverage of 0.035. Given the size of the training dataset, leverage appears to be of concern here.

Observation	1725	2018	2237	551	1732
Leverage	0.3931	0.2692	0.2474	0.2311	0.2096

Finally the model's predictive ability is assessed by predicting the 2245 observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	Value
RMSE	1.6894
R^2	0.5495
R^2_{pred}	0.5308

The fit statistics show that the thirteen components of the fitted PLS model does not perform too well at predicting new values.

5.6 Dataset 6

Dataset 6 consists of 219 observations, 2557 independent variables and one dependent variable. One hundred and forty observations were allocated to the training set and the remaining 79 observations were kept separate for testing purposes.

The TT plot for Dataset 6 is given in **FIGURE 5.27**. The majority of the points plot in a swarm around the centre of the graph, while a few of the observations appear to be removed from the bulk of the observations.

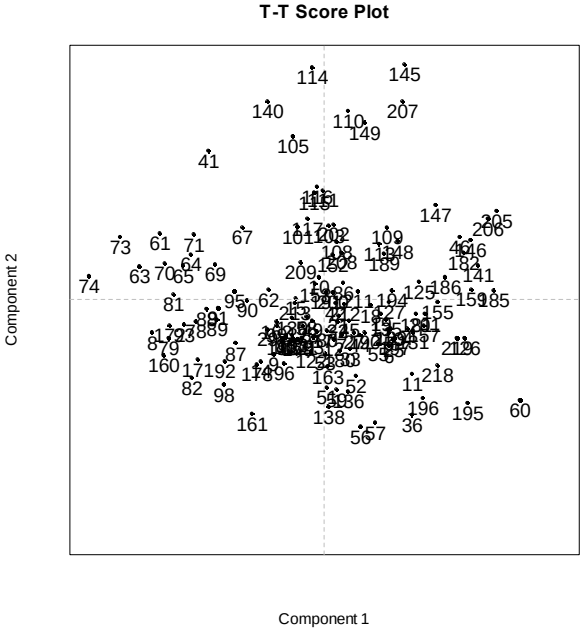

FIGURE 5.27 *The T-scores for component 1 and component 2.*

FIGURE 5.28 shows the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a minimum at ten components.

According to the scree plot (right panel), most of the variance is explained by the first component. A small proportion of the variance is explained by component 2, while the remaining components are insignificant in terms of the amount of variance that they explain. PLS modelling was done by retaining ten components.

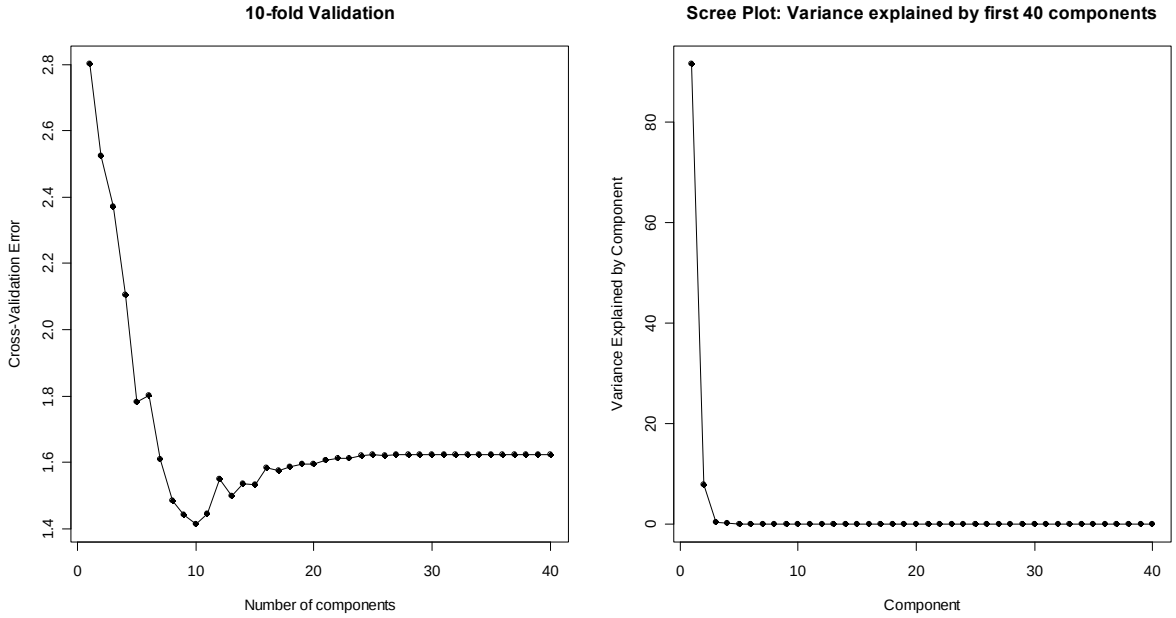

FIGURE 5.28 *The cross-validation error and a scree plot for Dataset 6.*

The relationship between the input spectra and the target variable (known chemical composition) is shown in **FIGURE 5.29**.

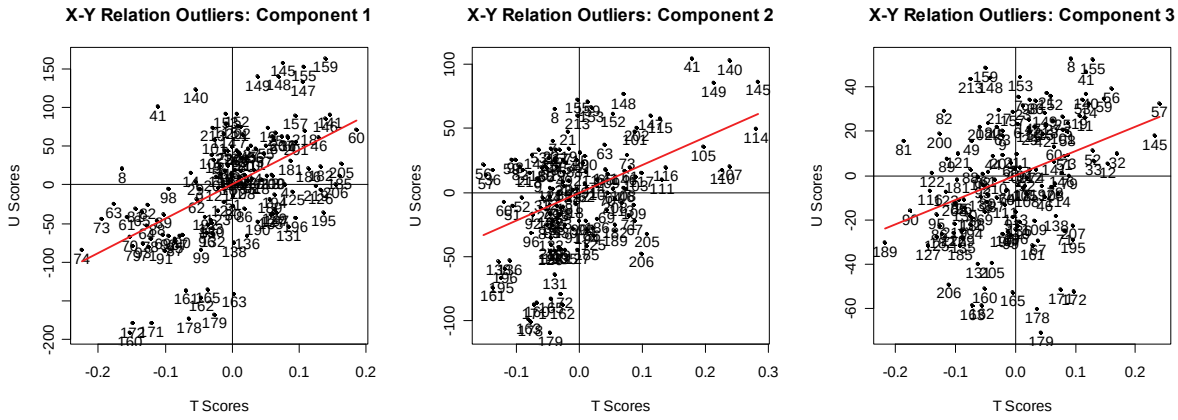

FIGURE 5.29 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components*

The observations, as shown in **FIGURE 5.29**, exhibit a significant amount of random scatter. The amount of random scatter is even more noticeable for the third component. There does not appear to be strong evidence of a linear relationship between the T- and U scores.

The five most influential observations are given below. Except for observation 135, leverage for the five most influential observations is reasonably constant. Leverage does not appear to be of concern here.

Observation	145	140	179	114	172
Leverage	0.3647	0.2069	0.2019	0.1983	0.1859

Finally the model's predictive ability is assessed by predicting the 79 observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	Value
RMSE	1.4532
R^2	0.8780
R^2_{pred}	0.8725

The fit statistics show that the ten components of the fitted PLS model performs reasonably well at predicting new values.

5.7 Dataset 7

Dataset 7 consists of 219 observations, 2557 independent variables and one dependent variable. One hundred and seventy two observations were allocated to the training set and the remaining 47 observations were kept separate for testing purposes.

The TT plot for Dataset 7 is given in **FIGURE 5.30**. The majority of the points plot in a swarm around the centre of the graph. A few of the observations appear to be removed from the bulk of the observations.

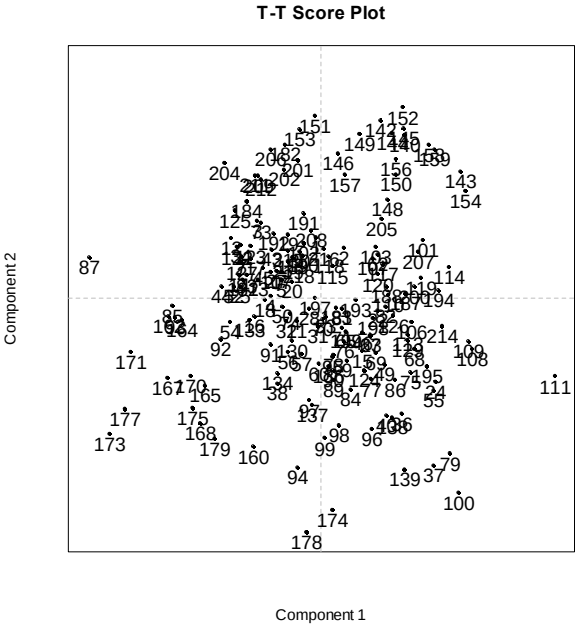

FIGURE 5.30 *The T-scores for component 1 and component 2.*

FIGURE 5.31 shows the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a minimum at twelve components.

According to the scree plot (right panel), most of the variance is explained by the first component. A small proportion of the variance is explained by components 2 to 4, while the remaining components are insignificant in terms of the amount of variance that they explain. PLS modelling was performed by retaining twelve components.

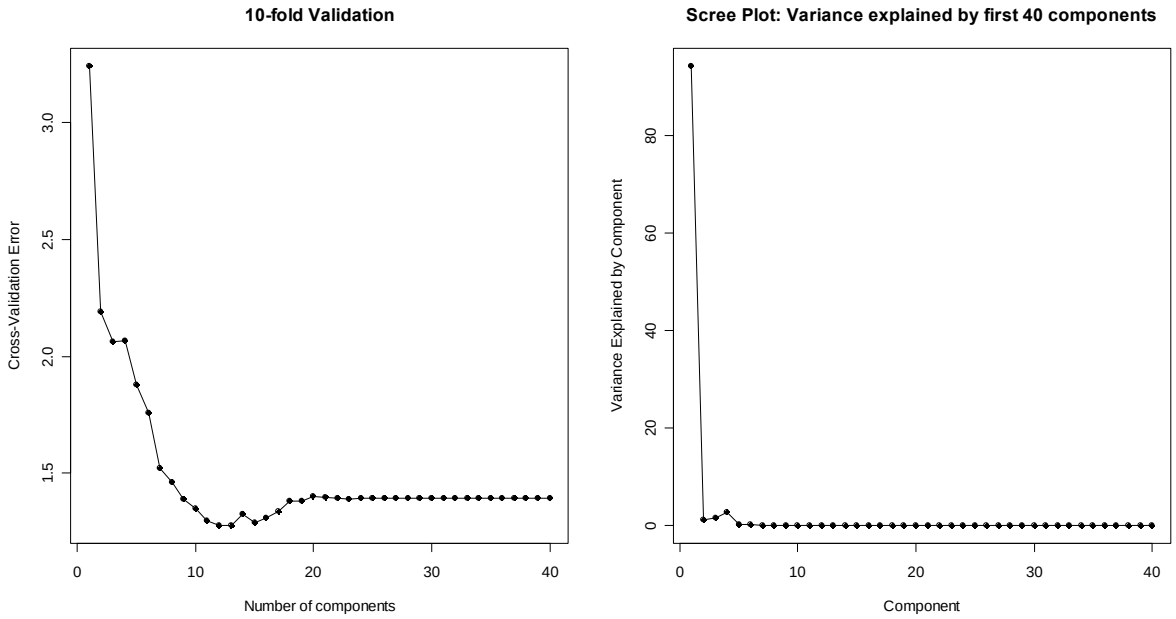

FIGURE 5.31 *The cross-validation error and a scree plot for Dataset 7.*

FIGURE 5.32 shows the relationship between the input spectra and the target variable (known chemical composition).

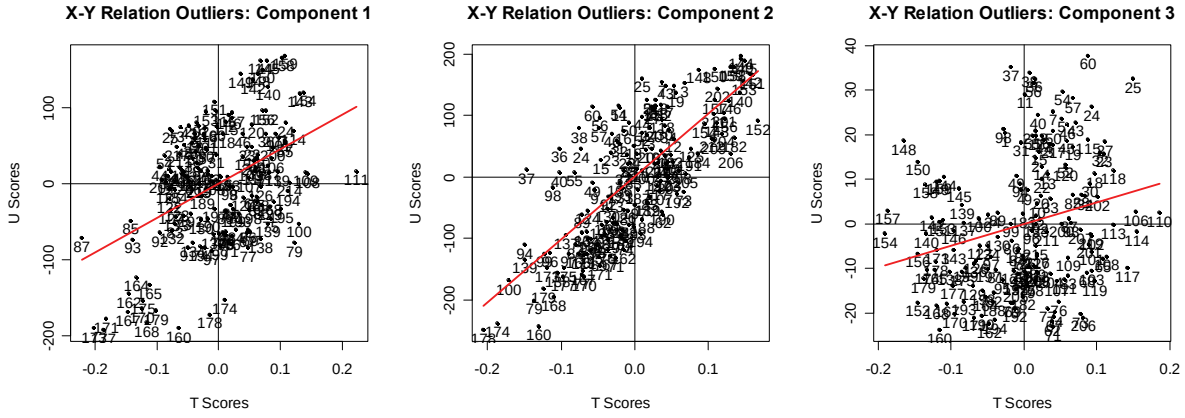

FIGURE 5.32 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components*

Considering components 1 and 3, there does not appear to be strong evidence of a linear relationship between the T- and U scores. There is however, a clear linear relationship between the T- and U scores for component 2.

The five most influential observations are given below. Except for observation 170, leverage for the five most influential observations was reasonably constant. Leverage does not appear to be of concern here.

Observation	170	28	207	139	159
Leverage	0.2903	0.1814	0.1785	0.1729	0.1708

Finally the model's predictive ability is assessed by predicting the 47 observations of the test set. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	Value
RMSE	1.6407
R^2	0.8000
R^2_{pred}	0.7939

The fit statistics show that the twelve components of the fitted PLS model performs reasonably well at predicting new values.

5.8 Dataset 8

Dataset 8 consists of 220 observations, 2557 independent variables and one dependent variable. The training set was allocated 156 observations the remaining 64 observations were used for testing.

The TT plot for Dataset 8 is given in **FIGURE 5.33**. The majority of the points plot in a swarm around the centre of the graph. Some of the observations appear to be removed from the bulk of the observations.

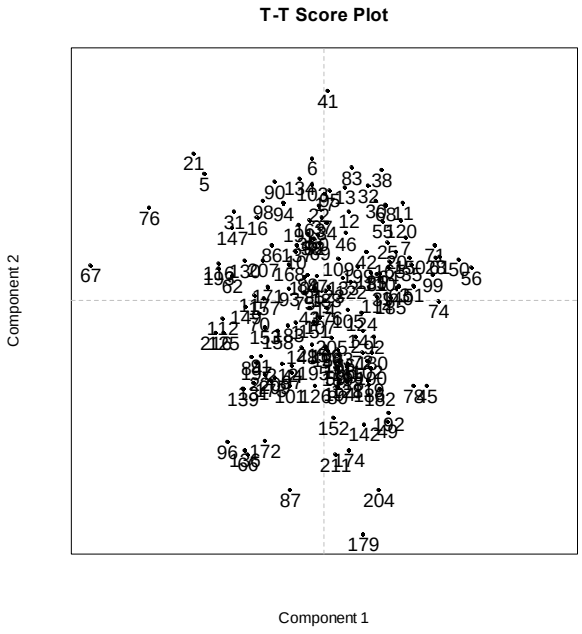

FIGURE 5.33 *The T-scores for component 1 and component 2.*

FIGURE 5.34 shows the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a minimum at twelve components.

According to the scree plot (right panel), most of the variance is explained by the first two components. The remaining components are insignificant in terms of the amount of variance that they explain. Twelve components were retained for modelling purposes.

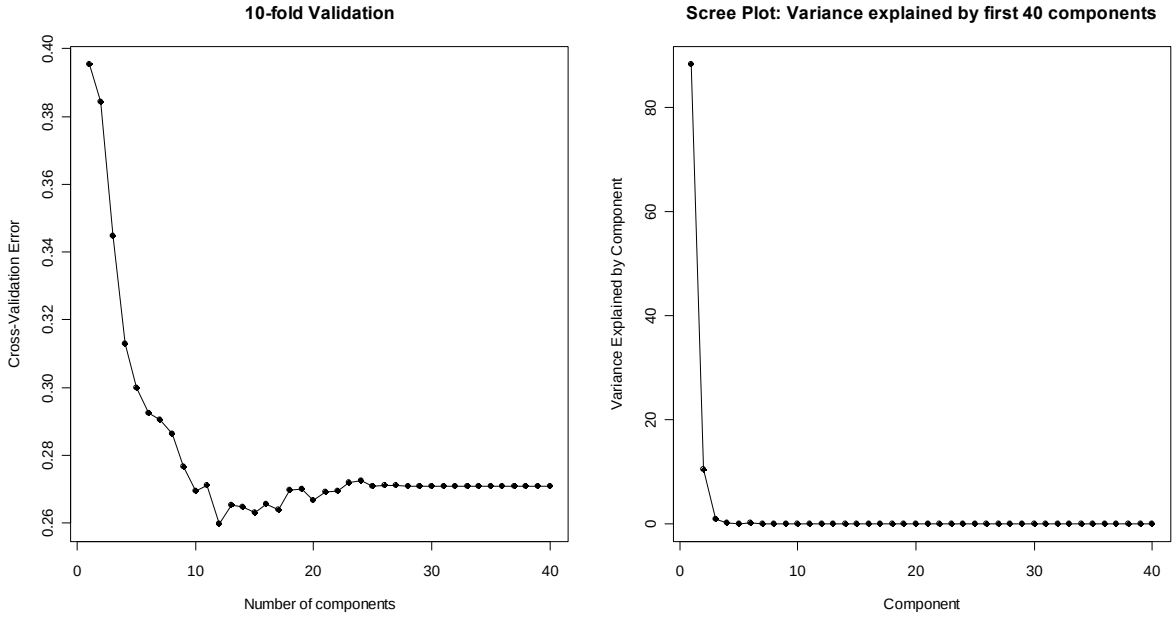

FIGURE 5.34 *The cross-validation error and a scree plot for Dataset 8.*

The relationship between the input spectra and the target variable (known chemical composition) is shown in **FIGURE 5.35**. A fair amount of random scatter can be observed for all three components. There does not appear to be strong evidence of a linear relationship between the T- and U scores.

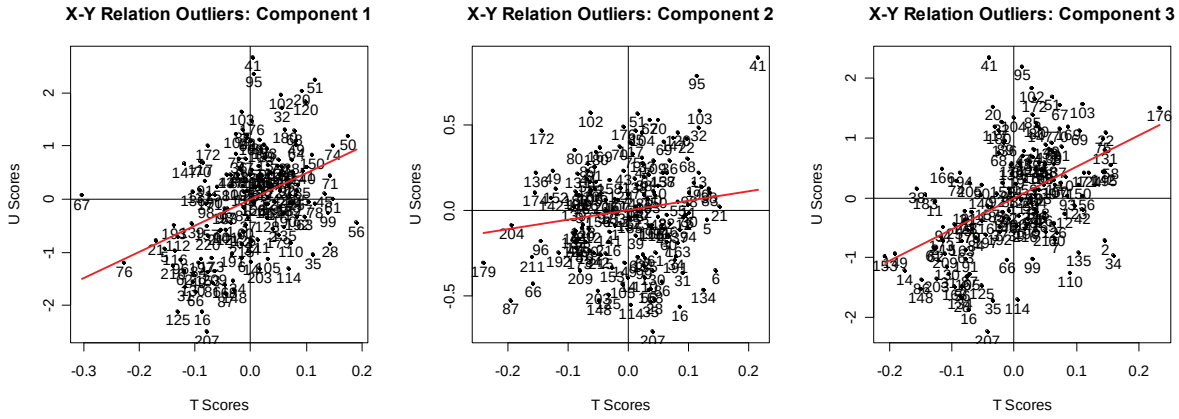

FIGURE 5.35 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components*

The five most influential observations are given below. Except for observation 67, leverage for the five most influential observations was reasonably constant. Leverage does not appear to be of concern here.

Observation	67	131	14	41	75
Leverage	0.2859	0.1898	0.1838	0.1615	0.1440

Finally the model's predictive ability is assessed by predicting the 64 observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	Value
RMSE	0.2523
R^2	0.7629
R^2_{pred}	0.7531

The fit statistics show that the twelve components of the fitted PLS model performs reasonably well at predicting new values.

5.9 Dataset 9

Dataset 9 consists of 200 observations, 2557 independent variables and one dependent variable. One hundred and thirty nine observations were allocated to the training set and 61 observations were used for testing purposes.

The TT plot for Dataset 9 is given in **FIGURE 5.36**. The majority of the points plot in a swarm around the centre of the graph. There appears to be a small group of observations that are scattered more towards the left of the graph.

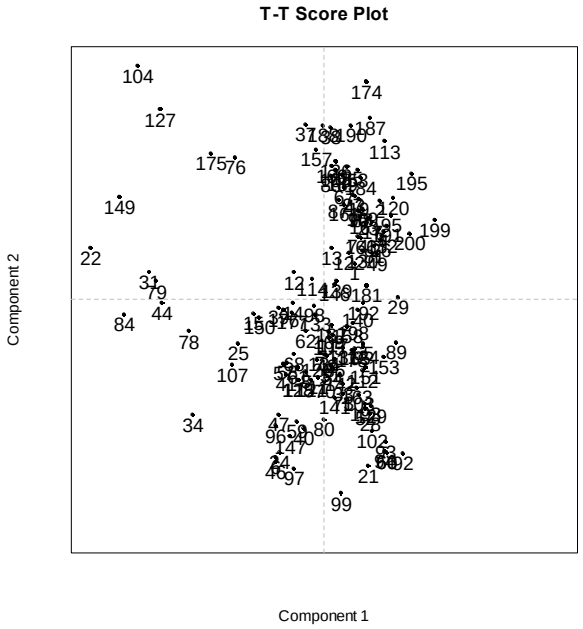

FIGURE 5.36 *The T-scores for component 1 and component 2.*

FIGURE 5.37 shows the cross-validation error as well as the percentage of the total variance explained by each component (only the first 40 components are shown). The cross-validation (left panel) error reaches a clear minimum at four components.

According to the scree plot (right panel), almost all of the variance is explained by the first two components. The remaining components are insignificant in terms of the amount of variance that they explain. PLS modelling was performed by retaining four components.

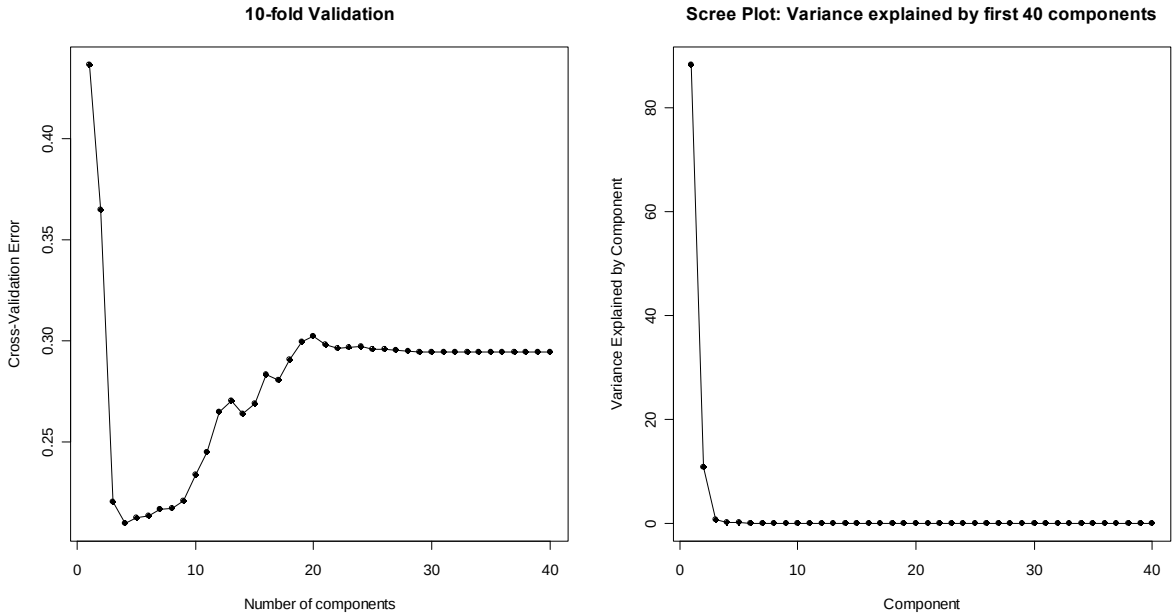

FIGURE 5.37 *The cross-validation error and a scree plot for Dataset 9.*

FIGURE 5.38 shows the relationship between the input spectra and the target variable (known chemical composition).

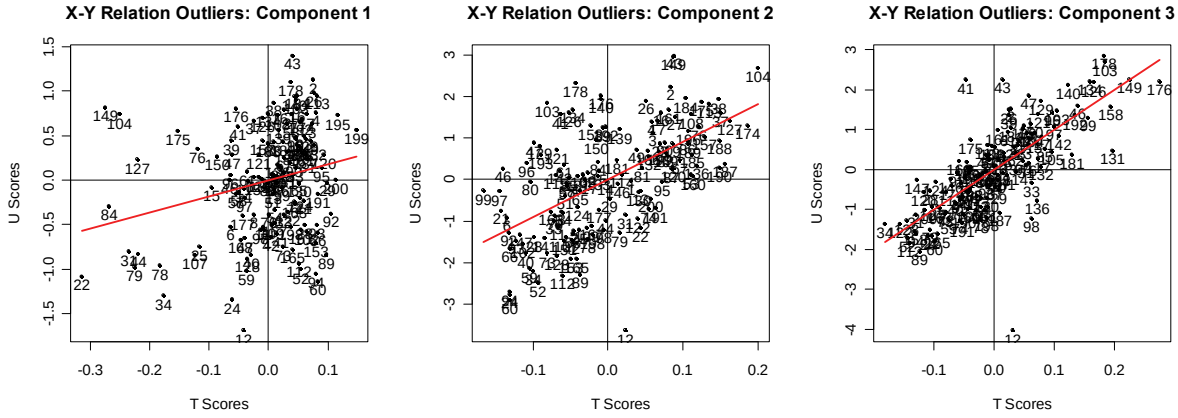

FIGURE 5.38 *The linear relationship between the X-space and Y-space, as given by the scores of the first 3 PLS components*

When investigating the first component, there does not appear to be strong evidence of a linear relationship between the T- and U scores. However, there is a clear linear relationship between the T- and U scores when looking at the second and third components. It appears if observation 12 is an outlier, as it continuously plots far from the bulk of the observations.

The five most influential observations are given below. The leverage for the five most influential observations is reasonably constant. It appears that observation 12 does not have a significant effect on the model's fit. Leverage does not appear to be of concern here.

Observation	22	37	41	149	167
Leverage	0.2701	0.2468	0.2146	0.2127	0.2108

Finally the model's predictive ability is assessed by predicting the 61 observations that were kept separate. The RMSE, R^2 and R^2_{pred} statistics were calculated:

Statistic	Value
RMSE	0.2076
R^2	0.7931
R^2_{pred}	0.7817

The fit statistics show that the four components of the fitted PLS model performs reasonably well at predicting new values.

6 Practical Implementation of Noise Addition PLS

6.1 Introduction

In the previous section, the input dataset was split into two subsets, namely a training set and a test set. The training set was used to train the model and the test set was used to assess the fit of the before mentioned model. For the purpose of NAPLS, three datasets are needed: a calibration set, a validation set and a test set. The calibration set is used to construct a PLS model, the validation set is used for internal optimisation (within the NAPLS algorithm) and the test set is kept separate in order to assess the final model's fit.

In order to be able to draw a direct comparison between PLS and NAPLS, the exact same training and test sets of the previous section were used. Contrary to "normal" PLS, which only requires one dataset for training, NAPLS requires two sets for training. These two datasets were obtained by dividing the original training set into a calibration- and a validation subset. The test set was therefore kept constant at all times, thereby allowing a direct comparison between PLS and NAPLS.

All the test results that are given in this section were obtained by executing the NAPLS algorithm with default parameters. Dardenne and Fernández Pierna (2006) suggest using a starting noise of 10% of the dependent variable's mean. The "optimal" model is then chosen as the model with the highest correlation to the median of all the models produced by 500 iterations of the NAPLS algorithm. The 500 iterations were obtained by doing 5 trials (outer loops) of 100 noise additions (inner loops) each.

In order to assess the stability and repeatability of the NAPLS algorithm, NAPLS was run ten times for each dataset. For each run, the calibration, validation and test sets were kept the same in order to eliminate all non-simulation variation.

6.2 Dataset 1

For normal PLS, a total of 104 observations were allocated to the training set, while the remaining 25 observations were used for the test set. A total of four components were retained. For NAPLS, the training set was split into a calibration- and a validation set of 52 observations each.

The results of ten runs of NAPLS are given in **TABLE 6.1**. These statistics were all calculated by using the final model to predict the independent test set. The first line gives the "PLS baseline" to which all the NAPLS models are compared. Only the calibration set was used to train the baseline model. NAPLS is compared to the baseline model in order to assess the algorithm's ability to capitalise on the information contained in the validation set.

TABLE 6.1: *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSEP		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.6684**	**0.0%**	**0.9400**	**0.0%**	**0.8977**	**0.0%**	
1	0.5442	18.6%	0.9459	0.6%	0.9322	3.8%	0.9781
2	0.5982	10.5%	0.9419	0.2%	0.9180	2.3%	0.9680
3	0.8157	-22.0%	0.9448	0.5%	0.8476	-5.6%	0.9807
4	0.5105	23.6%	0.9529	1.4%	0.9403	4.8%	0.9728
5	0.5588	16.4%	0.9286	-1.2%	0.9285	3.4%	0.9866
6	0.4585	31.4%	0.9534	1.4%	0.9518	6.0%	0.9641
7	0.5838	12.7%	0.9491	1.0%	0.9219	2.7%	0.9903
8	0.5281	21.0%	0.9411	0.1%	0.9361	4.3%	0.9853
9	0.4813	28.0%	0.9585	2.0%	0.9469	5.5%	0.9851
10	0.4518	32.4%	0.9614	2.3%	0.9532	6.2%	0.9807
Average	**0.5531**	**17.2%**	**0.9478**	**0.8%**	**0.9277**	**3.3%**	**0.9792**

The improvement in RMSE is comparable to the results of Dardenne and Fernández Pierna (2006), which had an average improvement of 25%. On average, NAPLS was able to reduce (improve) RMSE by 17.2% over the baseline PLS model. Both the squared correlation (R^2) and the proportion of explained variance (R^2_{pred}) showed small improvements. R^2_{median} is the squared correlation between the coefficients of the selected "optimal" model and the coefficients of the median model.

Comparing the results of NAPLS to the results of the baseline PLS model, that was trained using only the calibration set, will most probably be misleading, as the information of the validation set is built into to NAPLS model, but not into the baseline PLS model. It is therefore more plausible to compare the final results of NAPLS with the results of running PLS on the full training (calibration and validation) set. The results of both the full PLS model and NAPLS model are compared to the results of the baseline PLS model. This gives an indication of both algorithms' ability to extract additional information from the validation set.

TABLE 6.2 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	0.4994	0.6684	0.5531
R^2	0.9605	0.9400	0.9478
R^2_{pred}	0.9429	0.8977	0.9277

The effect of increasing the size of the training set is shown in **TABLE 6.2**. It is evident that the increase in sample size increases the quality of the fit of the model significantly, improving RMSE by about 25%. A true comparison between PLS and NAPLS can now be made: RMSE for NAPLS is about 5% worse than for PLS. Also, the proportion of explained variance for the predicted data, R^2_{pred}, is marginally lower for NAPLS. This shows that NAPLS did manage to improve the goodness of fit of the

model, but not sufficiently to outperform a PLS model based on the full (calibration and validation) data.

Additional tests in Chapter 7 show that RMSE can be improved by increasing the number of repetitions. In **TABLE 7.3** it is shown that, by increasing the number of inner loops to 500, RMSE improved to 0.4530. This is lower than RMSE for the full PLS model (0.4994), which shows that the NAPLS algorithm is able to improve upon the results of traditional PLS.

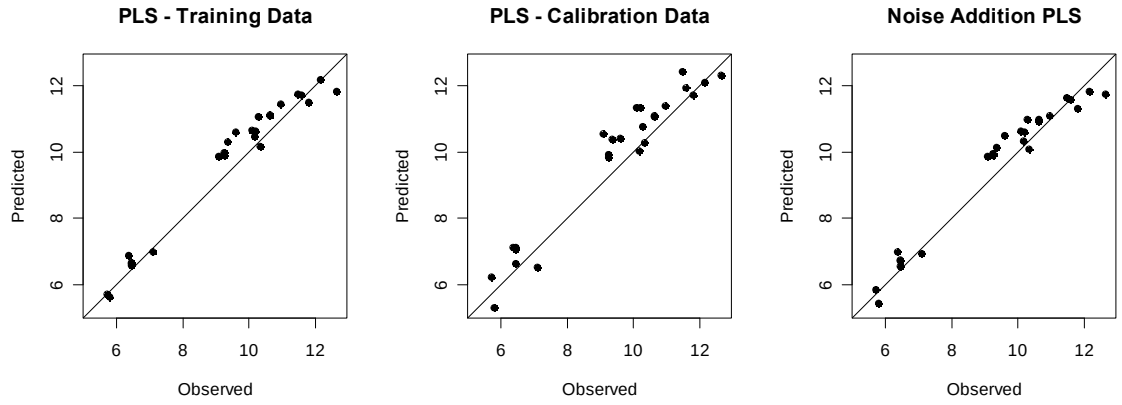

FIGURE 6.1 *Observed vs. Predicted values for Dataset 1.*

The observed versus predicted values for the full PLS model (trained using the full training set) is given in the left pane of **FIGURE 6.1**. The straight line indicates the position where the predicted values are expected to plot (i.e. the expected predicted value equals the observed value). The points lie in a fairly narrow band along the straight line, however, most of them plot above it. The right pane of **FIGURE 6.1** represents the NAPLS equivalent. The distribution of the points seems to be reasonably consistent with the distribution of the points in the left pane. These points, however, exhibit marginally more scatter than the predicted values for the full PLS model.

6.3 Dataset 2

For Dataset 2, 75 observations were allocated to each of the calibration and validation sets and the remaining 47 observations were kept separate for testing purposes. PLS modelling was performed by retaining 21 components.

TABLE 6.3 provides the results of ten runs of NAPLS. The fit statistics for the baseline PLS model is given in the first line.

Applying NAPLS to Dataset 2 yielded mixed results. In four cases, NAPLS yielded a marginally improved RMSE and R^2_{pred}. For the remaining six cases, however, noise addition decreased the model's accuracy when predicting data. On average, RMSE for Dataset 2 deteriorated with 2%. The

other fit statistics, R^2 and R^2_{pred}, also experienced a marginal decline. The results are however, not consistent, and more research needs to be done in order to stabilise the results of NAPLS.

TABLE 6.3 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2387**	**0.0%**	**0.4514**	**0.0%**	**0.4496**	**0.0%**	
1	0.2581	-8.1%	0.3820	-15.4%	0.3563	-20.8%	0.8516
2	0.2417	-1.3%	0.4409	-2.3%	0.4353	-3.2%	0.8633
3	0.2674	-12.0%	0.3417	-24.3%	0.3091	-31.3%	0.8790
4	0.2415	-1.2%	0.4384	-2.9%	0.4364	-2.9%	0.8388
5	0.2436	-2.1%	0.4325	-4.2%	0.4266	-5.1%	0.8550
6	0.2354	1.4%	0.5172	14.6%	0.4647	3.3%	0.8596
7	0.2373	0.6%	0.4606	2.0%	0.4560	1.4%	0.8680
8	0.2310	3.2%	0.5057	12.0%	0.4846	7.8%	0.8579
9	0.2457	-2.9%	0.4224	-6.4%	0.4169	-7.3%	0.8247
10	0.2329	2.4%	0.5124	13.5%	0.4759	5.8%	0.8734
Average	**0.2435**	**-2.0%**	**0.4454**	**-1.3%**	**0.4262**	**-5.2%**	**0.8571**

TABLE 6.4 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	0.2050	0.2387	0.2435
R^2	0.6201	0.4514	0.4454
R^2_{pred}	0.5939	0.4496	0.4262

The effect of increasing the sample size from the calibration set to the full training set yielded a moderate improvement for the fit statistics for PLS. However, all the fit statistics deteriorated when NAPLS was applied. Due to the inconsistency in results, it is not possible to determine if Dataset 2 is a good candidate for NAPLS. However, of the improvements that were observed, none were significant enough to compete with results of the full PLS model. It is therefore doubtful that an improvement in the consistency of the results would be of any value.

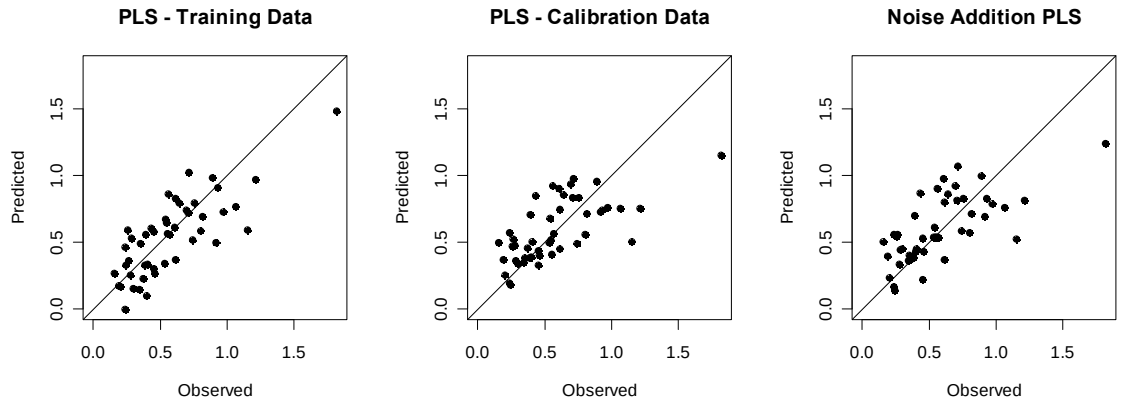

FIGURE 6.2 *Observed vs. Predicted values for Dataset 2.*

The predicted values for the full PLS model (as shown in the left pane of **FIGURE 6.2**) is distributed randomly around the straight line. These points exhibit a fair amount of scatter, which explains why the fit statistics for Dataset 2 were not too good. The NAPLS counterpart is shown in the right pane of **FIGURE 6.2**. There is a reasonable increase in the amount of scatter, as a result of deterioration of the quality of the model.

6.4 Dataset 3A

For Dataset 3A, the calibration and validation sets were each allocated 45 observations, while thirteen observations were kept separate for testing purposes. A total of eleven components were retained for modelling.

The results of ten runs of NAPLS are shown in **TABLE 6.5** below.

TABLE 6.5 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.0739**	**0.0%**	**0.7580**	**0.0%**	**0.6867**	**0.0%**	**0**
1	0.1394	-88.6%	0.2093	-72.4%	-0.1148	NA	0.5618
2	0.2159	-192.0%	0.1127	-85.1%	-1.6721	NA	0.3922
3	0.2105	-184.8%	0.4298	-43.3%	-1.5422	NA	0.4610
4	0.1982	-168.2%	0.6854	-9.6%	-1.2532	NA	0.4882
5	0.1466	-98.4%	0.5381	-29.0%	-0.2330	NA	0.5222
6	0.1910	-158.5%	0.4015	-47.0%	-1.0929	NA	0.4336
7	0.2547	-244.6%	0.2697	-64.4%	-2.7212	NA	0.4297
8	0.2283	-208.8%	0.4485	-40.8%	-1.9887	NA	0.4278
9	0.1298	-75.6%	0.2451	-67.7%	0.0344	-95.0%	0.4257
10	0.1301	-76.1%	0.5232	-31.0%	0.0287	-95.8%	0.6386
Average	**0.1845**	**-149.6%**	**0.3863**	**-49.0%**	**NA**	**NA**	**0.4781**

The application of NAPLS to Dataset 3A caused a significant decline in the quality of the model: On average, RMSE deteriorated (increased) by almost 150% when compared to the PLS baseline. Also, R^2 decreased significantly, showing that the predicted values had little correlation to the actual values. The full negative impact of noise addition can be seen when calculating R^2_{pred}. In eight of the ten runs, R^2_{pred} was negative. Taking into consideration that R^2_{pred} is calculated as $R^2_{pred} = (1 - SS_{prediction\ error} / SS_{total})$, this implies that the sum of squares for prediction error is larger that the total sum of squares of the observed values.

The correlation between the selected model and the median model (as measured by R^2_{median}) does not appear to be significantly large. This could be because the regression coefficients varied significantly between successive noise additions. Also, this could explain the erratic behaviour of the NAPLS model.

TABLE 6.6 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	0.0452	0.0739	0.1845
R^2	0.9243	0.7580	0.3863
R^2_{pred}	0.8828	0.6867	NA

The effect of increasing the sample size from the calibration set to the full training set yielded a significant improvement for the fit statistics for PLS. All the fit statistics deteriorated significantly when NAPLS was applied.

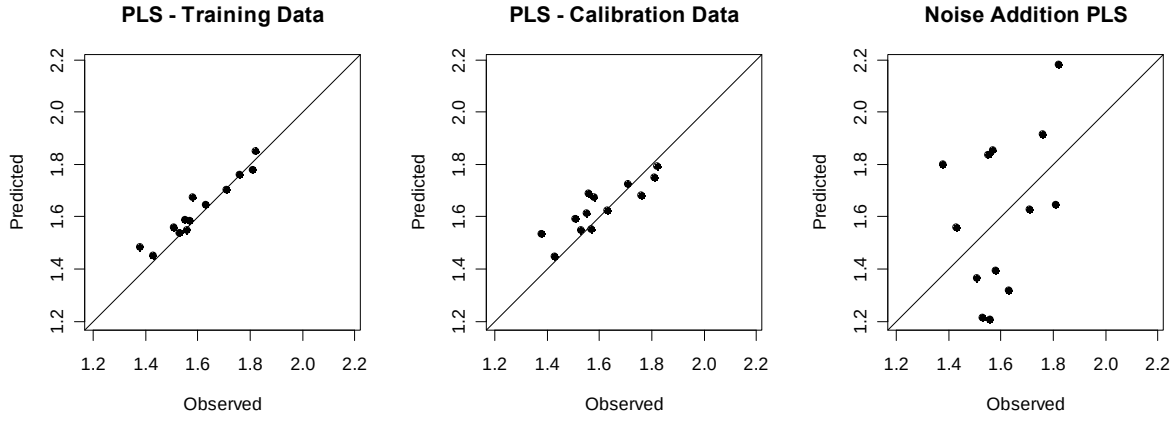

FIGURE 6.3 *Observed vs. Predicted values for Dataset 3A.*

The predicted values of the full PLS model, as shown in the left pane of **FIGURE 6.3**, all lie close to the straight line. In sharp contrast to this, the points of the NAPLS equivalent (right pane of **FIGURE 6.3**), exhibit an extreme amount of scatter. The variation of the predicted values is much higher than the

variation of the actual values. It is for this reason that R^2_{pred} is negative. It is clear that the NAPLS model is unstable as it produced erratic results.

6.5 Dataset 3B

For Dataset 3B, 45 observations were allocated to each of the calibration and validation sets, while twelve observations were kept separate for testing purposes. Nine components were retained for modelling purposes.

TABLE 6.7 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.7135**	**0.0%**	**0.9179**	**0.0%**	**0.9172**	**0.0%**	
1	0.9873	-38.4%	0.8687	-5.4%	0.8415	-8.3%	0.8544
2	1.0211	-43.1%	0.8886	-3.2%	0.8305	-9.5%	0.8522
3	1.8895	-164.8%	0.6966	-24.1%	0.4195	-54.3%	0.8475
4	0.9687	-35.8%	0.8515	-7.2%	0.8474	-7.6%	0.8114
5	1.1908	-66.9%	0.7833	-14.7%	0.7694	-16.1%	0.8712
6	1.0825	-51.7%	0.8828	-3.8%	0.8095	-11.7%	0.8685
7	1.0745	-50.6%	0.8210	-10.6%	0.8122	-11.4%	0.9208
8	0.9260	-29.8%	0.8673	-5.5%	0.8606	-6.2%	0.8297
9	1.0945	-53.4%	0.8228	-10.4%	0.8052	-12.2%	0.8557
10	1.1051	-54.9%	0.8226	-10.4%	0.8014	-12.6%	0.8784
Average	**1.1340**	**-58.9%**	**0.8305**	**-9.5%**	**0.7797**	**-15.0%**	**0.8590**

The results of ten runs of NAPLS are provided in **TABLE 6.7**. The NAPLS algorithm yielded poor results when applied to Dataset 3B. RMSE increased almost 60%, while both R^2 and R^2_{pred} experienced a reasonable decline. Finally, R^2_{median} is better than R^2_{median} for Dataset 3A, but remains sufficiently worse than R^2_{median} for Dataset 1. It appears as if the regression coefficients are sensitive to the addition of noise.

TABLE 6.8 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	0.9687	0.7135	1.1340
R^2	0.9021	0.9179	0.8305
R^2_{pred}	0.8733	0.9172	0.7797

Decreasing the sample size from the full training set to only the calibration set had quite an interesting effect on the model's fit – all three fit statistics improved. As mentioned previously, NAPLS brought upon a decrease in the quality of the fit statistics for Dataset 3B.

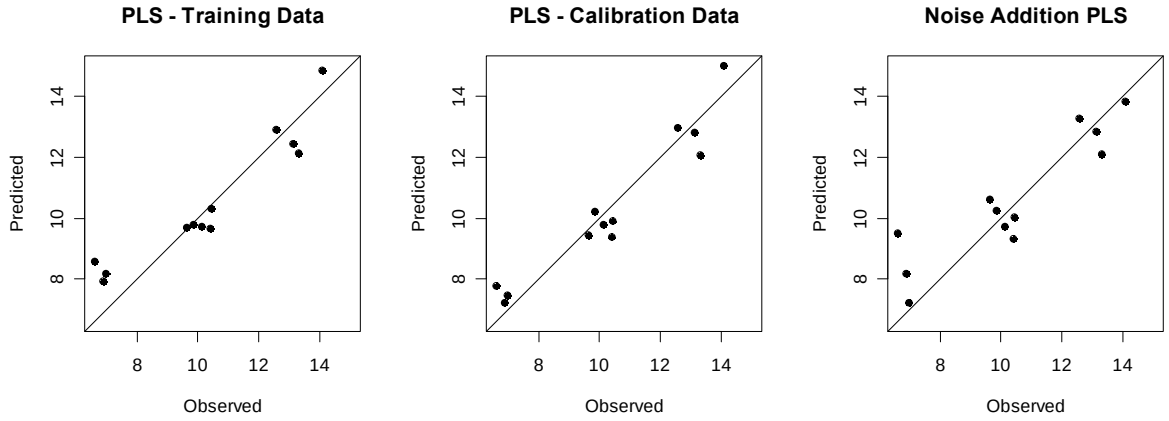

FIGURE 6.4 *Observed vs. Predicted values for Dataset 3B.*

The observed versus predicted values for the full PLS model is shown in the left pane of **FIGURE 6.4**. By observing the left pane of **FIGURE 6.4**, it is evident that there is a formation of three to four small clusters of points. This would imply that NIR was applied at different levels of the constituent of interest. These groups may have some relation to the subgroups that were observed in **FIGURE 5.10**.

Similar, but less clearly defined, groupings can be identified when studying the observed versus predicted values for the model produced by NAPLS (right pane, **FIGURE 6.4**). Also, it is evident that noise addition decreased the quality of the model, as the predicted values from NAPLS exhibit more scatter than the predicted values of the full PLS model.

6.6 Dataset 3C

The calibration and validation sets each comprised 45 observations, while twelve observations were kept separate for model testing. A total of eleven components were retained for modelling.

The results of ten runs of NAPLS are shown in **TABLE 6.7**. Applying NAPLS to Dataset 3C failed to yield an improvement in results. On average, RMSE deteriorated by over 60%. The correlation between the observed and predicted values also decreased from 0.92 to an average of 0.73. Also, the correlation between the selected model and the median model was not significantly high for any of the runs. It appears as if noise addition destabilised the model and is therefore not suitable for Dataset 3C.

TABLE 6.9 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.0804**	**0.0%**	**0.9192**	**0.0%**	**0.8423**	**0.0%**	
1	2.4270	-124.6%	0.5914	-35.7%	0.2043	-75.7%	0.6265
2	1.6247	-50.4%	0.8316	-9.5%	0.6434	-23.6%	0.6654
3	1.7832	-65.0%	0.6412	-30.2%	0.5705	-32.3%	0.6337
4	1.6371	-51.5%	0.8792	-4.3%	0.6380	-24.3%	0.6865
5	1.6168	-49.6%	0.6484	-29.5%	0.6469	-23.2%	0.6149
6	1.2475	-15.5%	0.8666	-5.7%	0.7898	-6.2%	0.8380
7	1.7334	-60.4%	0.6850	-25.5%	0.5941	-29.5%	0.7405
8	1.8260	-69.0%	0.6622	-28.0%	0.5496	-34.8%	0.6554
9	1.7907	-65.7%	0.6547	-28.8%	0.5669	-32.7%	0.6568
10	1.8865	-74.6%	0.7999	-13.0%	0.5193	-38.4%	0.6742
Average	**1.7573**	**-62.6%**	**0.7260**	**-21.0%**	**0.5723**	**-32.1%**	**0.6792**

TABLE 6.10 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	0.9687	1.0804	1.7573
R^2	0.9021	0.9192	0.7260
R^2_{pred}	0.8733	0.8423	0.5723

Decreasing the sample size from the full training set to only the calibration set had a marginally negative influence on the model's fit. NAPLS, however, failed to utilise the information in the validation set of Dataset 3C.

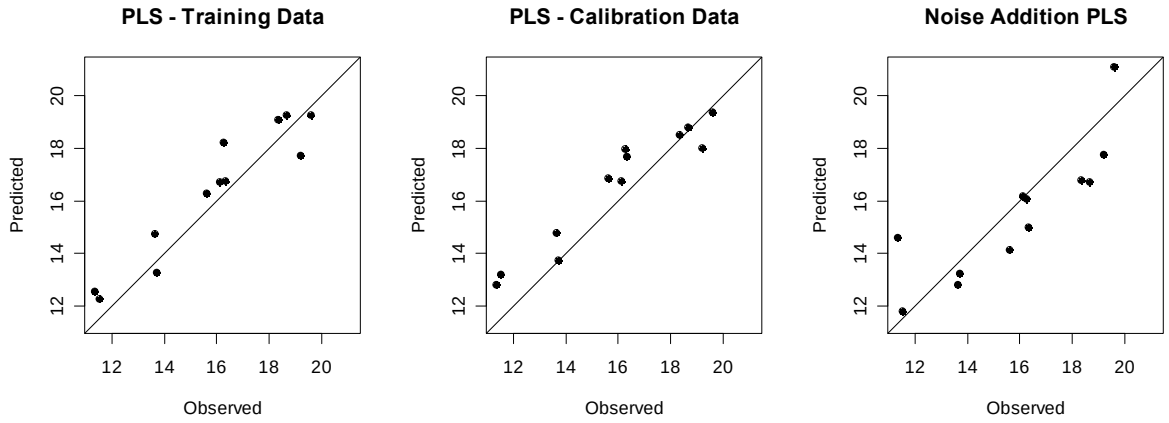

FIGURE 6.5 *Observed vs. Predicted values for Dataset 3C.*

The predicted values for the full PLS model are closely scattered around the straight line in the left pane of **FIGURE 6.5**. For the calibration PLS model, the amount of scatter is marginally increased, while most of the predicted values are above their expected values. The right pane of **FIGURE 6.5** shows the

results for NAPLS: There is a fair increase in the amount of scatter of the predicted values. Also, the majority of the predicted values are now below their expected value. This would explain the larger deterioration of RMSE and R^2_{pred} when compared to R^2.

6.7 Dataset 4

The calibration and validation sets of Dataset 4 were each allocated 95 observations. The remaining 47 observations were kept separate for testing purposes. The final PLS model was trained using 17 retained components.

TABLE 6.11 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5088**	**0.0%**	**0.6874**	**0.0%**	**0.6752**	**0.0%**	
1	2.6426	-75.1%	0.4392	-36.1%	0.0037	-99.5%	0.6071
2	2.2990	-52.4%	0.5404	-21.4%	0.2459	-63.6%	0.6944
3	3.2694	-116.7%	0.2578	-62.5%	-0.5250	NA	0.6088
4	2.4800	-64.4%	0.5621	-18.2%	0.1225	-81.9%	0.7006
5	1.7216	-14.1%	0.6012	-12.5%	0.5771	-14.5%	0.7334
6	1.7926	-18.8%	0.6317	-8.1%	0.5415	-19.8%	0.5959
7	2.5698	-70.3%	0.3737	-45.6%	0.0578	-91.4%	0.5958
8	2.5995	-72.3%	0.3767	-45.2%	0.0359	-94.7%	0.5951
9	2.6041	-72.6%	0.4018	-41.5%	0.0325	-95.2%	0.6376
10	3.0417	-101.6%	0.3117	-54.7%	-0.3200	NA	0.6030
Average*	**2.5020**	**-65.8%**	**0.4496**	**-34.6%**	**0.2021**	**-70.1%**	**0.6372**

** Negative values of R^2_{pred} excluded from average*

TABLE 6.11 shows the results of ten runs of NAPLS. For Dataset 4, unsatisfactory results were produced by the NAPLS algorithm. RMSE increased over 60%, while both R^2 and R^2_{pred} experienced a significant decline. In two of the cases, R^2_{pred} is negative. The results are however, not consistent between different runs. Finally, R^2_{median} is not significantly high for any of the runs. It appears as if the regression coefficients are quite sensitive to the presence of noise.

TABLE 6.12 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	1.3114	1.5088	2.5020
R^2	0.7752	0.6874	0.4496
R^2_{pred}	0.7546	0.6752	0.2021

A comparison between the results of PLS and NAPLS is given in **TABLE 6.12**. A small deterioration in the fit statistics was observed when the calibration set was excluded from the full training data. The results obtained with NAPLS were noticeably worse than for the baseline and full PLS models.

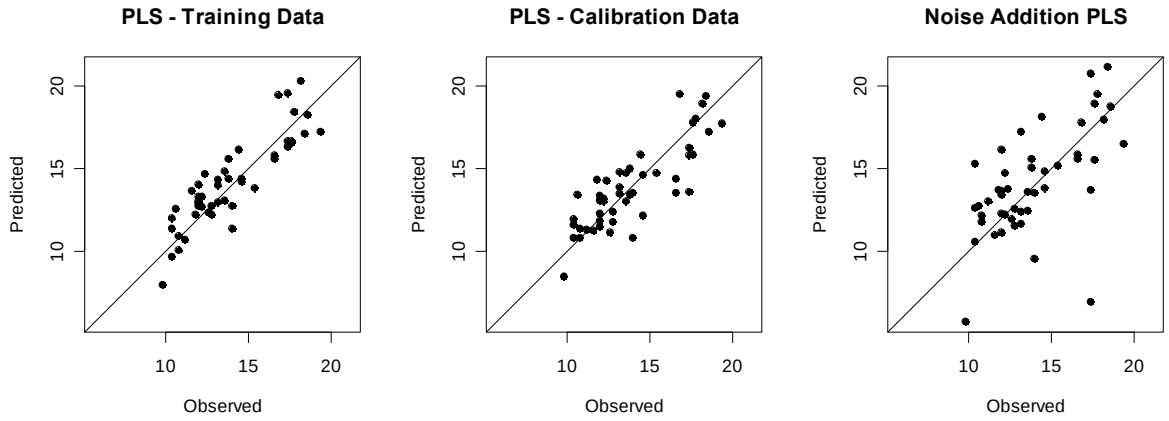

FIGURE 6.6 *Observed vs. Predicted values for Dataset 4.*

The predicted values for the full PLS model (as shown in the left pane of **FIGURE 6.6**) are distributed randomly around the straight line. These points exhibit a reasonable amount of scatter, which explains why the fit statistics for Dataset 4 are less than desirable. The NAPLS counterpart is shown in the right pane of **FIGURE 6.6**. It is clear that the quality of the model deteriorated, as there is a fair amount of increase in the size of the scatter.

6.8 Dataset 5

The calibration and validation sets were each allocated 200 observations, while 2245 observations were allocated to the test set. Thirteen components were retained for PLS modelling.

TABLE 6.13 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.7958**	**0.0%**	**0.5048**	**0.0%**	**0.4698**	**0.0%**	
1	1.8401	-2.5%	0.4830	-4.3%	0.4434	-5.6%	0.8847
2	1.9093	-6.3%	0.4573	-9.4%	0.4007	-14.7%	0.9125
3	1.8424	-2.6%	0.4942	-2.1%	0.4419	-5.9%	0.9051
4	1.8957	-5.6%	0.4609	-8.7%	0.4092	-12.9%	0.8994
5	1.9338	-7.7%	0.4609	-8.7%	0.3852	-18.0%	0.8921
6	1.9857	-10.6%	0.4487	-11.1%	0.3518	-25.1%	0.9025
7	1.8654	-3.9%	0.4880	-3.3%	0.4279	-8.9%	0.9254
8	1.8323	-2.0%	0.5038	-0.2%	0.4480	-4.6%	0.9104
9	1.9014	-5.9%	0.4724	-6.4%	0.4056	-13.7%	0.9131
10	1.9102	-6.4%	0.4808	-4.8%	0.4001	-14.8%	0.9111
Average	**1.8916**	**-5.3%**	**0.4750**	**-5.9%**	**0.4114**	**-12.4%**	**0.9056**

Noise addition was not able to improve the fit of the model. As shown in **TABLE 6.13**, RMSE decreased marginally over the baseline, while the correlation between the predicted and observed values decreased by almost 6%. Overall, R^2_{pred} deteriorated slightly, declining from 0.47 to 0.41. R^2_{median} is not significantly high. The regression coefficients appear to be sensitive to the addition of noise.

TABLE 6.14 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	1.6894	1.7958	1.8916
R^2	0.5495	0.5048	0.4750
R^2_{pred}	0.5308	0.4698	0.4114

Decreasing the sample size from the full training set to only the calibration set had a negative influence on the fit of the model. The application of NAPLS to Dataset 5 failed to deliver satisfactory results, with R^2_{pred} deteriorating significantly from 0.53 to 0.41.

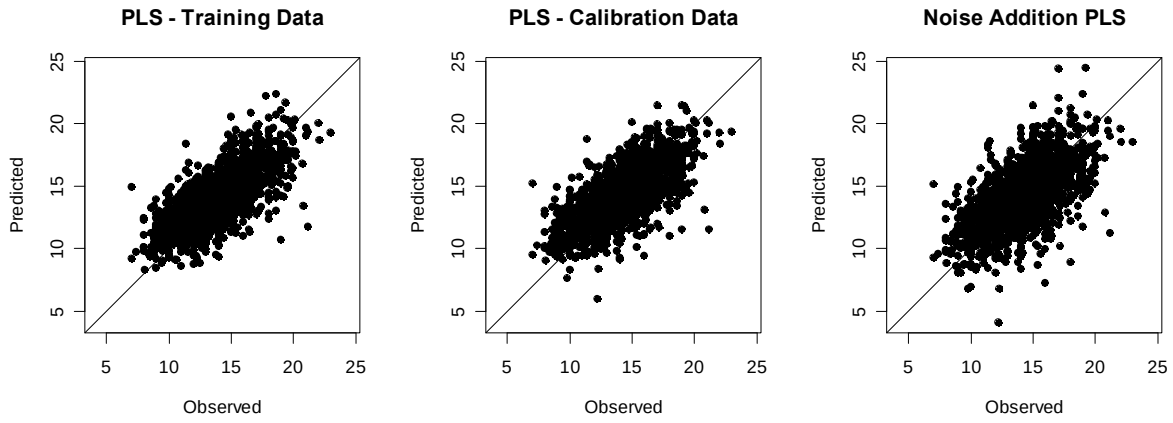

FIGURE 6.7 *Observed vs. Predicted values for Dataset 5.*

The predicted values for the full PLS model (**FIGURE 6.7**, left pane) exhibits a fair amount of scatter around the straight line. This would explain why the fit statistics for Dataset 5 were not good. The NAPLS counterpart is shown in the right pane of **FIGURE 6.7**. There is a fair increase in the spread of the scatter, indicating that the quality of the model deteriorated.

6.9 Dataset 6

The calibration and validation sets each consisted of 70 observations. The remaining 79 observations were kept separate to assess the selected models' goodness of fit. Ten components were retained for modelling purposes.

Noise addition brought about a significant deterioration in the model's quality for Dataset 6. As shown in **TABLE 6.15**, RMSE increased to over three times the baseline, while the correlation between the predicted and observed values decreased by almost 60%. Due to the poor quality of the selected model, R^2_{pred} was negative for every run. R^2_{median} is quite low, showing that the regression coefficients are sensitive to the addition of noise.

TABLE 6.15 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5425**	**0.0%**	**0.8646**	**0.0%**	**0.8564**	**0.0%**	
1	6.4724	-319.6%	0.3302	-61.8%	-1.5285	NA	0.5278
2	6.0739	-293.8%	0.2082	-75.9%	-1.2267	NA	0.5808
3	5.9551	-286.1%	0.5234	-39.5%	-1.1404	NA	0.5980
4	5.9184	-283.7%	0.0392	-95.5%	-1.1141	NA	0.4007
5	7.2492	-370.0%	0.5149	-40.4%	-2.1718	NA	0.3717
6	5.8878	-281.7%	0.5148	-40.5%	-1.0923	NA	0.4758
7	5.4300	-252.0%	0.6083	-29.6%	-0.7796	NA	0.3725
8	7.3259	-375.0%	0.1653	-80.9%	-2.2393	NA	0.4788
9	6.0524	-292.4%	0.4541	-47.5%	-1.2110	NA	0.6251
10	6.8027	-341.0%	0.2143	-75.2%	-1.7931	NA	0.4818
Average	**6.3168**	**-309.5%**	**0.3573**	**-58.7%**	**NA**	**NA**	**0.4913**

TABLE 6.16 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	1.3733	1.5424	6.3168
R^2	0.8898	0.8646	0.3573
R^2_{pred}	0.8862	0.8564	NA

For Dataset 6, a small improvement in the goodness of fit was observed when the validation data was included in the training data for the PLS model. A significant decrease in the model's quality was observed after noise addition was applied.

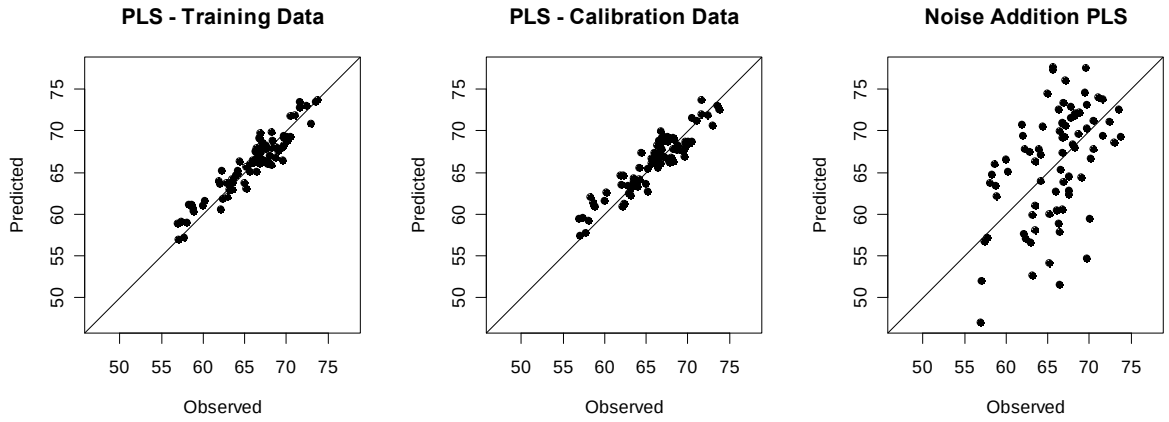

FIGURE 6.8 *Observed vs. Predicted values for Dataset 6.*

The predicted values for the full PLS model (**FIGURE 6.8**, left pane) all plot in a narrow band around the straight line. There is a significant increase in the amount of the scatter for the NAPLS counterpart (right pane of **FIGURE 6.8**), indicating that the quality of the model deteriorated significantly.

6.10 Dataset 7

The calibration and validation sets for Dataset 7 were each allocated 86 observations. A total of 47 observations were allocated to the test set. Modelling was performed by retaining twelve components.

TABLE 6.17 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.6313**	**0.0%**	**0.8035**	**0.0%**	**0.7963**	**0.0%**	
1	7.7605	-375.7%	0.4049	-49.6%	-3.6104	NA	0.3657
2	8.2854	-407.9%	0.0960	-88.0%	-4.2552	NA	0.3713
3	4.7363	-190.3%	0.3381	-57.9%	-0.7173	NA	0.3381
4	6.1384	-276.3%	0.4684	-41.7%	-1.8846	NA	0.3906
5	4.9040	-200.6%	0.3083	-61.6%	-0.8411	NA	0.4342
6	6.7199	-311.9%	0.3635	-54.8%	-2.4570	NA	0.3218
7	10.1004	-519.2%	0.0257	-96.8%	-6.8098	NA	0.4010
8	6.2961	-286.0%	0.2918	-63.7%	-2.0347	NA	0.2982
9	5.7498	-252.5%	0.3285	-59.1%	-1.5309	NA	0.4603
10	7.9318	-386.2%	0.0830	-89.7%	-3.8163	NA	0.3906
Average	**6.8623**	**-320.7%**	**0.2708**	**-66.3%**	**NA**	**NA**	**0.3772**

A significant deterioration in the fit statistics was observed with the application of noise addition. As shown in **TABLE 6.17**, RMSE increased over three times compared to the baseline, while the correlation between the predicted and observed values decreased by almost 70%. The regression coefficients appear to be sensitive to the addition of noise, as R^2_{pred} was negative for every run.

TABLE 6.18 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	1.6407	1.6313	6.8623
R^2	0.8000	0.8035	0.2708
R^2_{pred}	0.7939	0.7963	NA

The additional information in the validation set did not contribute significantly towards improving the quality of the PLS model. Noise addition had a severe negative impact on the quality of the model, as the fit statistics deteriorated significantly.

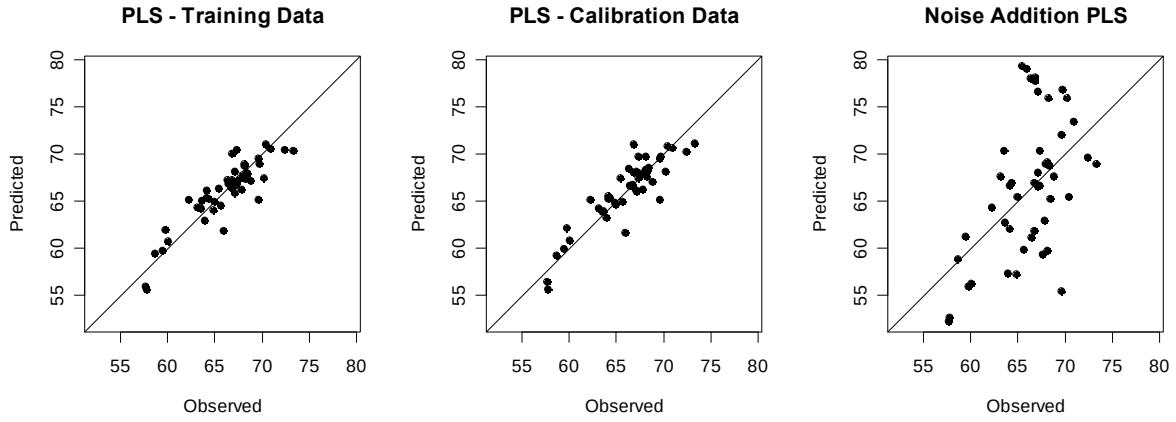

FIGURE 6.9 *Observed vs. Predicted values for Dataset 7.*

The predicted values for the full PLS model (**FIGURE 6.9**, left pane) all plot reasonable closely around the straight line. As shown in the right pane of **FIGURE 6.9**, noise addition significantly increased the amount of scatter. This corresponds to the deterioration that was observed with the fit statistics.

6.11 Dataset 8

The calibration and validation sets for Dataset 8 were each allocated 78 observations. The remaining 64 observations were kept separate in order to assess the quality of the final models. PLS modelling was performed by retaining twelve components.

All the fit statistics deteriorated significantly when noise addition was applied to Dataset 8. As shown in **TABLE 6.19**, RMSE increased considerably compared to the baseline, while the correlation between the predicted and observed values decreased by more than 75%. R^2_{pred} was negative for every run. The regression coefficients appear to be sensitive to the addition of noise, as R^2_{median} remained low for each of the ten runs.

TABLE 6.19 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2960**	**0.0%**	**0.6652**	**0.0%**	**0.6600**	**0.0%**	
1	1.0038	-239.1%	0.1250	-81.2%	-2.9089	NA	0.4266
2	0.7062	-138.6%	0.2730	-59.0%	-0.9349	NA	0.4654
3	1.0748	-263.1%	0.1477	-77.8%	-3.4819	NA	0.4036
4	1.3748	-364.4%	0.0623	-90.6%	-6.3331	NA	0.4111
5	1.2384	-318.4%	0.1032	-84.5%	-4.9500	NA	0.4356
6	1.0508	-255.0%	0.2576	-61.3%	-3.2835	NA	0.4468
7	1.1758	-297.2%	0.2151	-67.7%	-4.3636	NA	0.4722
8	1.1348	-283.4%	0.1121	-83.2%	-3.9960	NA	0.4023
9	0.7835	-164.7%	0.1861	-72.0%	-1.3816	NA	0.4114
10	1.1731	-296.3%	0.1332	-80.0%	-4.3393	NA	0.4309
Average	**1.0716**	**-262.0%**	**0.1615**	**-75.7%**	**NA**	**NA**	**0.4306**

TABLE 6.20 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	0.2523	0.2960	1.0716
R^2	0.7629	0.6652	0.1615
R^2_{pred}	0.7531	0.6600	NA

Decreasing the sample size from the full training set to only the calibration set had a negative effect on the fit of the model. The application of NAPLS to Dataset 8 yielded worse than expected results, as all the fit statistics deteriorated significantly.

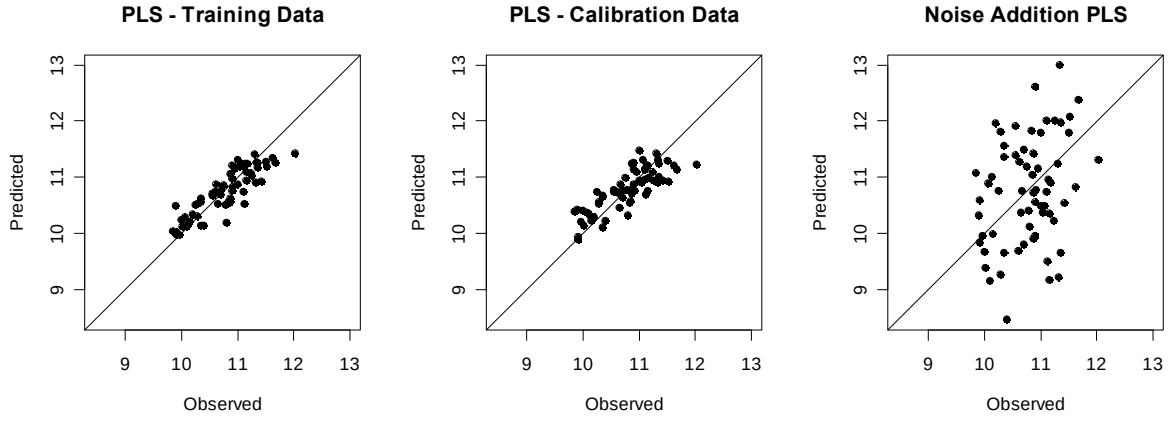

FIGURE 6.10 *Observed vs. Predicted values for Dataset 8.*

The predicted values for the full PLS model (**FIGURE 6.10**, left pane) all plot in a well defined band along the straight line. The NAPLS counterpart is shown in the right pane of **FIGURE 6.10**. There is a

significant increase in the spread of the scatter; to such an extent that the predicted values exhibit more variance than the actual values. This would explain why R^2_{pred} was negative for each run. The NAPLS algorithm failed for Dataset 8, as the final model produced erratic results.

6.12 Dataset 9

From the original 200 observations, 70 observations were allocated to the calibration set, 69 to the validation set and 61 to the test set. A total of twelve components were retained for modelling purposes.

Noise addition failed to yield satisfactory results for Dataset 9. As shown in **TABLE 6.21**, RMSE increased by more than 100%, while the correlation between the predicted and observed values decreased by almost 22%. R^2_{pred} experienced a significant decline and was negative for two of the runs.

TABLE 6.21 *RMSE (Root Mean Square Error for Prediction), R^2 (squared correlation coefficient) and R^2_{pred} (Coefficient of determination for predicted data) for 10 independent runs of NAPLS. R^2_{median} gives the squared correlation between the selected "optimal" model and the median model.*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2061**	**0.0%**	**0.7850**	**0.0%**	**0.7848**	**0.0%**	
1	0.3975	-92.8%	0.4817	-38.6%	0.1997	-74.5%	0.8555
2	0.5563	-169.9%	0.5446	-30.6%	-0.5675	NA	0.9258
3	0.3965	-92.4%	0.6081	-22.5%	0.2036	-74.1%	0.8873
4	0.5593	-171.4%	0.6778	-13.7%	-0.5848	NA	0.9505
5	0.3088	-49.8%	0.6376	-18.8%	0.5171	-34.1%	0.9405
6	0.4089	-98.4%	0.6856	-12.7%	0.1530	-80.5%	0.9068
7	0.3456	-67.7%	0.6118	-22.1%	0.3949	-49.7%	0.9455
8	0.3743	-81.6%	0.6850	-12.7%	0.2905	-63.0%	0.9596
9	0.4014	-94.7%	0.5772	-26.5%	0.1840	-76.6%	0.9198
10	0.3782	-83.5%	0.6342	-19.2%	0.2755	-64.9%	0.9317
Average*	**0.4127**	**-100.2%**	**0.6143**	**-21.7%**	**0.1066**	**-64.7%**	**0.9223**

** Negative values of R^2_{pred} excluded from average*

TABLE 6.22 *A comparison of the quality of fit of PLS versus NAPLS. For PLS, the effect of increasing the sample size, from the calibration set to the full training set, is contrasted against NAPLS.*

Statistic	PLS (Cal + Val)	PLS (Cal)	NAPLS (Cal + Val)
RMSE	0.2036	0.2061	0.4127
R^2	0.7924	0.7850	0.6143
R^2_{pred}	0.7900	0.7848	0.1066

Decreasing the sample size from the full training set to only the calibration set had a negligible effect on the fit of the model. The application of NAPLS to Dataset 9 yielded poor results, as all the fit statistics deteriorated significantly.

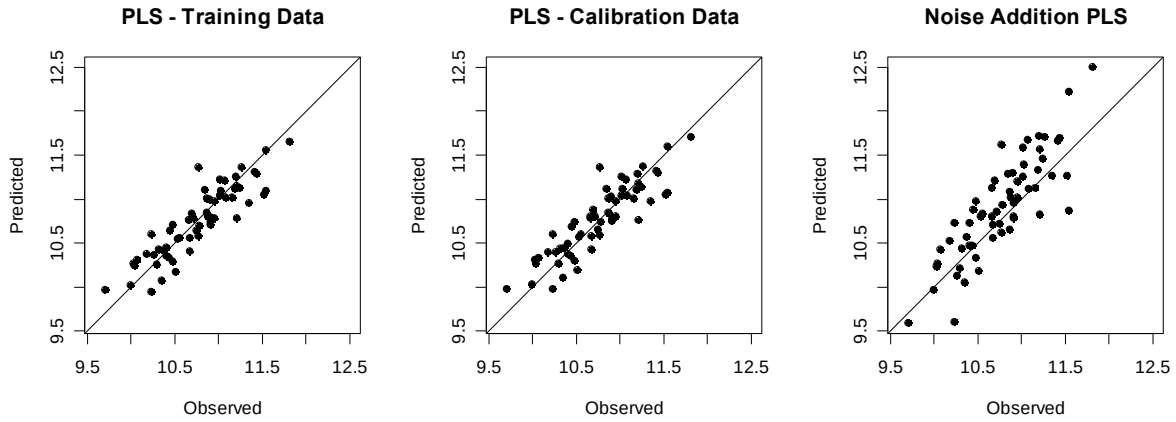

FIGURE 6.11 *Observed vs. Predicted values for Dataset 9.*

There is a small amount of scatter present in the predicted values for the full PLS model, as shown in the left pane of **FIGURE 6.11**. There is a reasonable increase in the amount of scatter for the NAPLS counterpart, as shown in the right pane of **FIGURE 6.11**. This is due to the significant deterioration of the quality of the final model.

6.13 The effect of the size of the calibration set

For all the studies conducted in this chapter, the training data was divided equally between the calibration and validation sets. In order to assess the effect of the size of the calibration set, relative to the validation set, these studies were repeated using 70% of the training data for the calibration set and 30% of the training data for the validation set. For further reference, NAPLS using 70% of the training data for calibration purposes will be referred to as $NAPLS_{70}$. Likewise, NAPLS using 50% of the training data for the calibration set will be referred to as $NAPLS_{50}$.

A comparison between $NAPLS_{70}$ and $NAPLS_{50}$ is given in **TABLE 6.23** and **TABLE 6.24**. For $NAPLS_{70}$ the NAPLS algorithm was repeated ten times. The average of the then runs was then compared to the results of $NAPLS_{50}$ in the previous studies.

For the majority of the cases, the goodness of fit for the baseline model improved due to the increase in size of the calibration set. The results of $NAPLS_{70}$ were not consistent when compared to $NAPLS_{50}$. As shown in **TABLE 6.25** and **TABLE 6.26**, the results of $NAPLS_{70}$ did not always improve upon the results of $NAPLS_{50}$. Also, for Dataset 1, $NAPLS_{50}$ managed to improve upon the results of the baseline. The results for $NAPLS_{70}$, however, were all worse when compared to the baseline.

TABLE 6.23 *A comparison of RMSE for $NAPLS_{50}$ against RMSE for $NAPLS_{70}$*

	Cal : Val - 50:50			Cal : Val - 70:30		
Dataset	**PLS Base**	**NAPLS**	**% Δ**	**PLS Base**	**NAPLS**	**% Δ**
Dataset 1	0.6684	0.5531	17%	0.3968	0.4607	-16%
Dataset 2	0.2387	0.2435	-2%	0.1865	0.1922	-3%
Dataset 3A	0.0739	0.1845	-150%	0.0349	0.1561	-348%
Dataset 3B	0.7135	1.1340	-59%	0.8311	1.1126	-34%
Dataset 3C	1.0804	1.7573	-63%	0.9917	1.7910	-81%
Dataset 4	1.5088	2.5020	-66%	1.2905	1.3636	-6%
Dataset 5	1.7958	1.8916	-5%	1.7632	1.8752	-6%
Dataset 6	1.5425	6.3168	-310%	1.4516	4.6087	-217%
Dataset 7	1.6313	6.8623	-321%	1.9687	7.4926	-281%
Dataset 8	0.2960	1.0716	-262%	0.2648	1.0323	-290%
Dataset 9	0.2061	0.4127	-100%	0.2039	0.4100	-101%

TABLE 6.24 *A comparison of R_2 for $NAPLS_{50}$ against RMSE for $NAPLS_{70}$*

	Cal : Val - 50:50			Cal : Val - 70:30		
Dataset	**PLS Base**	**NAPLS**	**% Δ**	**PLS Base**	**NAPLS**	**% Δ**
Dataset 1	0.9400	0.9478	1%	0.9667	0.9566	-1%
Dataset 2	0.4514	0.4454	-1%	0.6613	0.6480	-2%
Dataset 3A	0.7580	0.3863	-49%	0.8810	0.3688	-58%
Dataset 3B	0.9179	0.8305	-10%	0.7926	0.6811	-14%
Dataset 3C	0.9192	0.7260	-21%	0.8970	0.7708	-14%
Dataset 4	0.6874	0.4496	-35%	0.8468	0.8080	-5%
Dataset 5	0.5048	0.4750	-6%	0.5162	0.4751	-8%
Dataset 6	0.8646	0.3573	-59%	0.8764	0.4537	-48%
Dataset 7	0.8035	0.2708	-66%	0.7267	0.2513	-65%
Dataset 8	0.6652	0.1615	-76%	0.7331	0.2499	-66%
Dataset 9	0.7850	0.6143	-22%	0.7896	0.6333	-20%

A direct comparison between $NAPLS_{50}$ and $NAPLS_{70}$ is given in **TABLE 6.25** and **TABLE 6.26**. The first three columns give the results of PLS, $NAPLS_{50}$ and $NAPLS_{70}$ respectively. $NAPLS_{50-70}$ compares the performance of $NAPLS_{70}$ to the performance of $NAPLS_{50}$. In the last two columns, the results of $NAPLS_{50}$ and $NAPLS_{70}$ are compared with the results of PLS.

In most cases, the increase in the size of the calibration set improved the results of the final model. This is consistent with previous results that compared the full PLS model to the 50% baseline model. For Datasets 1 and 2, $NAPLS_{70}$ performed better than the full PLS model. $NAPLS_{70}$ managed to improve the correlation between the observed and predicted values for Dataset 4. However, the root mean square error for Dataset 4 was marginally worse after the application of NAPLS.

The relative performance of $NAPLS_{70}$ and $NAPLS_{50}$, when compared to the 50% and 70% baselines respectively, was not found to be consistent. In some cases, the absolute change in results was higher for $NAPLS_{70}$, and in other cases it was higher for $NAPLS_{50}$. It does not appear as if the size of the calibration set, relative to the size of the validation set, plays any significant role. Considering that

none of the $NAPLS_{70}$ models managed to outperform their baselines, the improvement in results can only be attributed to the increase in size of the calibration set.

TABLE 6.25 *A comparison of RMSE for PLS, $NAPLS_{50}$, $NAPLS_{70}$*

Dataset	PLS (Full)	$NAPLS_{50}$	$NAPLS_{70}$	$NAPLS_{50-70}$	$NAPLS_{50}$ vs. PLS	$NAPLS_{70}$ vs. PLS
Dataset 1	0.4994	0.5531	0.4607	17%	-11%	8%
Dataset 2	0.2050	0.2435	0.1922	21%	-19%	6%
Dataset 3A	0.0452	0.1845	0.1561	15%	-308%	-245%
Dataset 3B	0.9037	1.1340	1.1126	2%	-25%	-23%
Dataset 3C	0.9687	1.7573	1.7910	-2%	-81%	-85%
Dataset 4	1.3114	2.5020	1.3636	46%	-91%	-4%
Dataset 5	1.6894	1.8916	1.8752	1%	-12%	-11%
Dataset 6	1.4532	6.3168	4.6087	27%	-335%	-217%
Dataset 7	1.6407	6.8623	7.4926	-9%	-318%	-357%
Dataset 8	0.2523	1.0716	1.0323	4%	-325%	-309%
Dataset 9	0.2076	0.4127	0.4100	1%	-99%	-97%

TABLE 6.26 *A comparison of R^2 for PLS, $NAPLS_{50}$, $NAPLS_{70}$*

Dataset	PLS (Full)	$NAPLS_{50}$	$NAPLS_{70}$	$NAPLS_{50-70}$	$NAPLS_{50}$ vs. PLS	$NAPLS_{70}$ vs. PLS
Dataset 1	0.9605	0.9478	0.9566	1%	-1%	0%
Dataset 2	0.6201	0.4454	0.6480	46%	-28%	5%
Dataset 3A	0.9243	0.3863	0.3688	-5%	-58%	-60%
Dataset 3B	0.8861	0.8305	0.6811	-18%	-6%	-23%
Dataset 3C	0.9021	0.7260	0.7708	6%	-20%	-15%
Dataset 4	0.7752	0.4496	0.8080	80%	-42%	4%
Dataset 5	0.5495	0.4750	0.4751	0%	-14%	-14%
Dataset 6	0.8780	0.3573	0.4537	27%	-59%	-48%
Dataset 7	0.8000	0.2708	0.2513	-7%	-66%	-69%
Dataset 8	0.7629	0.1615	0.2499	55%	-79%	-67%
Dataset 9	0.7931	0.6143	0.6333	3%	-23%	-20%

7 The effect of changing the number of iterations

7.1 The effect of the number of iterations on the stability of the model

In an attempt to fine-tune the stability of the results of NAPLS, the predictive ability of the fitted models were assessed by changing the various input parameters and assessing the impact over ten consecutive runs of NAPLS. For each run, the exact same calibration, validation and test sets were used.

The standard deviation of the results can be used as a measure of the stability of the output of the algorithm. However, it is reasonable to expect that a variable with a higher mean (due to a larger measurement scale) will have a larger variation around its mean. The standard deviation is therefore not a good statistic for comparative purposes, as it can be biased depending on the measurement scale used. This problem can be overcome by using the coefficient of variation, which divides the standard deviation by the mean. This produces a standardised statistic that is scale invariant and can be used to compare the results from different datasets.

The results for the different test cases are given in **TABLE 7.1** and presented graphically in **FIGURE 7.1**. With the exception of Datasets 3C, 6 and 7, NAPLS for all the datasets experienced an increase in the variability of the predictive error as the level of noise increased. There does not appear to be a clear relationship between the number of iterations and the variability of the prediction error.

TABLE 7.1 *The coefficient of variation (CV) for RMSE for different settings of the input parameters.*

Coefficient of Var.: RMSE	**NAPLS Parameters (Outer Loops x Inner Loops x Noise%)**				
Dataset	**5x100x5%**	**5x100x10%**	**15x100x10%**	**5x300x10%**	**5x500x10%**
Dataset 1	0.1366	0.1895	0.2080	0.1454	0.0927
Dataset 2	0.0171	0.0468	0.0413	0.0440	0.0740
Dataset 3A	0.2236	0.2435	0.2755	0.2746	0.1814
Dataset 3B	0.1664	0.2437	0.2299	0.2695	0.2401
Dataset 3C	0.2606	0.1679	0.2705	0.1791	0.2848
Dataset 4	0.1215	0.1921	0.1525	0.2135	0.1168
Dataset 5	0.0168	0.0253	0.0289	0.0296	0.0388
Dataset 6	0.1510	0.0992	0.1881	0.1514	0.1068
Dataset 7	0.2752	0.2425	0.1424	0.2895	0.2462
Dataset 8	0.1382	0.1883	0.1169	0.1790	0.1574
Dataset 9	0.1252	0.1990	0.1785	0.1456	0.1850

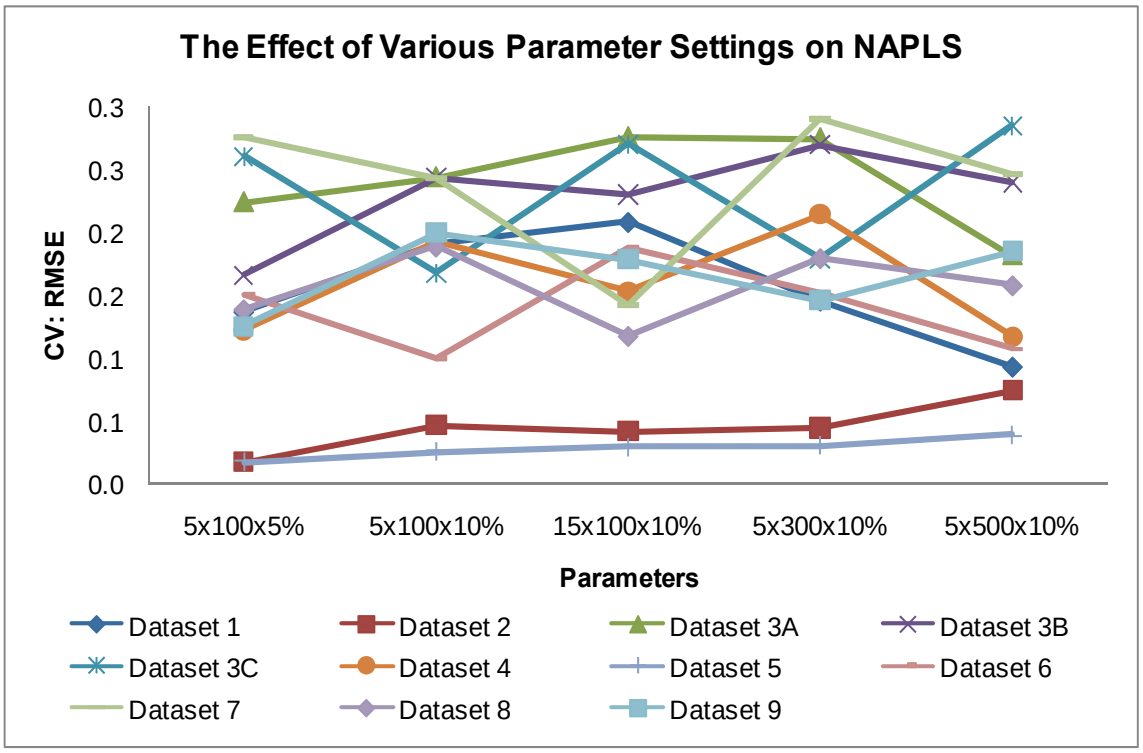

FIGURE 7.1 *The effect of the different parameters on the variability of the prediction error.*

The effect of the different parameters on the variability of R^2 (as measured by the coefficient of variation of R^2) is given in **TABLE 7.2** and presented graphically in **FIGURE 7.2**. There was an increase in the level of variability with the higher level of noise. The results for the various numbers of iterations appear to be inconsistent, as an increase in the number if iterations did not necessarily increase or decrease the variability of R^2. From these results it can be deduced that the NAPLS algorithm in its current form does not necessarily converge towards an optimum result with an increase in the number of iterations.

TABLE 7.2 *The effect of different parameters on the coefficient of variation of the correlation between actual and predicted values*

Coefficient of Var.: R^2	**NAPLS Parameters (Outer Loops x Inner Loops x Noise%)**				
Dataset	**5x100x5%**	**5x100x10%**	**15x100x10%**	**5x300x10%**	**5x500x10%**
Dataset 1	0.0044	0.0101	0.0069	0.0040	0.0049
Dataset 2	0.0411	0.1276	0.1002	0.1070	0.1276
Dataset 3A	0.4463	0.4540	0.6935	0.7081	0.4702
Dataset 3B	0.0587	0.0692	0.1140	0.2309	0.1225
Dataset 3C	0.1123	0.1466	0.1774	0.1846	0.2446
Dataset 4	0.0900	0.2836	0.1851	0.2558	0.1679
Dataset 5	0.0327	0.0376	0.0414	0.0445	0.0511
Dataset 6	0.0932	0.5368	0.3141	0.4417	0.3407
Dataset 7	0.2332	0.5521	0.4669	0.6120	0.6114
Dataset 8	0.3042	0.4286	0.5630	0.6567	0.5222
Dataset 9	0.0695	0.1070	0.0726	0.0536	0.0387

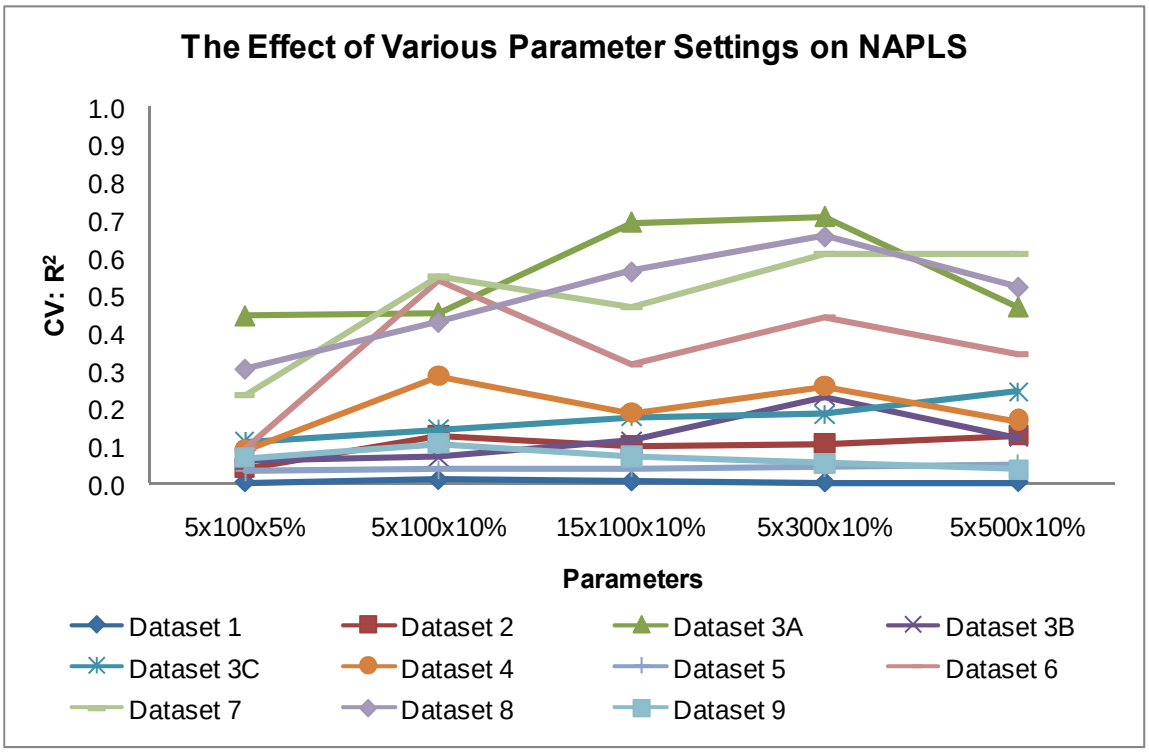

FIGURE 7.2 *The effect of the different parameters on the variability of the correlation between observed and predicted values.*

The effect of the number of iterations on the model's predictive error is given in **FIGURE 7.3**. It should be noted that RMSE for each dataset was first standardised by changing the range of RMSE to [0, 1] per simulation set (i.e. the ten runs for a specific set of parameters). This was done in order to eliminate the effect of the size of the measurement scale, and thereby making it possible to compare the change in RMSE for various datasets. The original results for RMSE are given in **TABLE 7.3** below.

The number of iterations does not have a consistent effect on the size of RMSE. It should be noted that only Dataset 1 benefited from noise addition. Dataset 1 continued to experience an improvement in RMSE, as the number if inner repetitions increased.

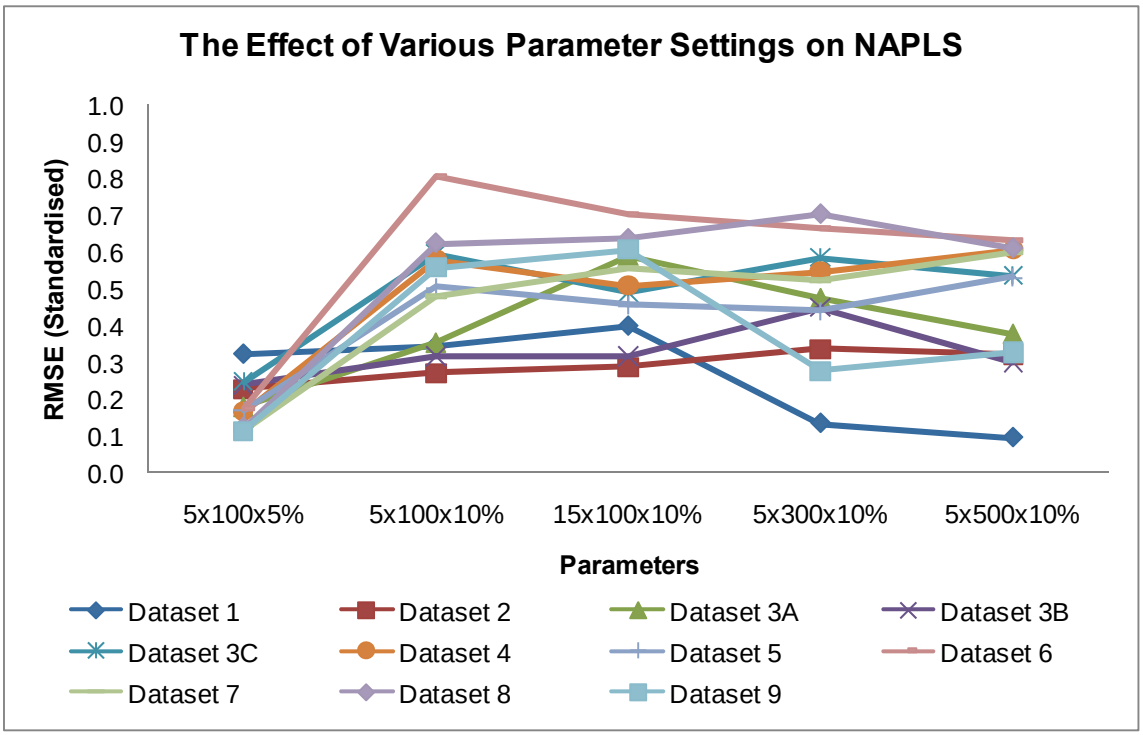

FIGURE 7.3 *The effect of the different parameters on the variability of the correlation between observed and predicted values.*

TABLE 7.3 *The effect of different parameters on the prediction error*

RMSE	NAPLS Parameters (Outer Loops x Inner Loops x Noise%)				
Dataset	**5x100x5%**	**5x100x10%**	**15x100x10%**	**5x300x10%**	**5x500x10%**
Dataset 1	0.5442	0.5531	0.5749	0.4686	0.4530
Dataset 2	0.2405	0.2435	0.2444	0.2476	0.2467
Dataset 3A	0.1295	0.1845	0.2590	0.2225	0.1920
Dataset 3B	1.0171	1.1340	1.1316	1.3290	1.1082
Dataset 3C	1.1488	1.7573	1.5724	1.7429	1.6504
Dataset 4	1.7071	2.5020	2.3643	2.4378	2.5549
Dataset 5	1.8136	1.8916	1.8806	1.8775	1.8986
Dataset 6	3.0329	6.3168	5.7684	5.5690	5.4206
Dataset 7	3.3464	6.8623	7.6446	7.2808	8.0418
Dataset 8	0.5961	1.0716	1.0828	1.1466	1.0582
Dataset 9	0.2658	0.4127	0.4286	0.3199	0.3362

8 Conclusion and Recommendations for Future Research.

A brief overview of principal component regression (PCR) and partial least squares (PLS) regression was given as theoretical background to the practical research that was conducted in this paper. After a brief discussion on model selection, both PLS and Noise Addition PLS (NAPLS) were evaluated on nine different NIR datasets.

Noise addition PLS was implemented with varied results. For Dataset 1, an improvement in the goodness of fit was observed. However, it took more than the required 500 repetitions in order for NAPLS to improve upon the performance of the full PLS model, constructed on all the training data. For the remainder of the cases, NAPLS failed to improve upon the results of PLS. In some cases, such as Datasets 3A, 6, 7 and 8, NAPLS caused a severe deterioration in the fit statistics.

From the test performed here, it is evident that the results of NAPLS were not consistent and in some cases improvements and deteriorations between successive trials were observed. The inconsistency did not appear to improve with an increase in the number of noise additions.

Further research is required in order to determine the optimal level of noise addition, as well as the size of the noise reduction factor. Also, it is necessary that the algorithm is fine-tuned in order to improve the stability of the results. A potential inefficiency of the current algorithm is Step 10 of **ALGORITHM 4.1**. Here, the current model's coefficients are stored even if noise addition did not improve the fit of the model. Additional research needs to be conducted on the feasibility of this approach, as well possible alternatives.

A brief investigation on the influence of the size of the calibration set, relative to the validation set, yielded inconclusive results. The relative amount of improvement or deterioration was not consistent when the results of NAPLS with a larger calibration set, were compared to the results of NAPLS with a smaller calibration set.

During the initial phases of data analysis, such as the Dataset 2, a few cases of influential values were observed. Also, for some of the datasets, special forms of experimental design were evident: For Dataset 1, replicates were used and for Datasets 3A, 3B and 3C, small subgroups were observed. Additional investigation will have to be done in order to determine the effect of these cases on Noise Addition PLS. Also, further research on creating the calibration and validation datasets is required, especially in the light of the experimental design patterns that were observed.

9 Addendum A: NAPLS increasing the size of the calibration set.

TABLE 9.1 *$NAPLS_{70}$ Results for Dataset 1*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.3968**	**0.0%**	**0.9667**	**0.0%**	**0.9616**	**0.0%**	
1	0.4735	-19.3%	0.9464	-2.1%	0.9454	-1.7%	0.9509
2	0.4560	-14.9%	0.9628	-0.4%	0.9493	-1.3%	0.9625
3	0.4120	-3.8%	0.9664	0.0%	0.9586	-0.3%	0.9667
4	0.5736	-44.6%	0.9453	-2.2%	0.9198	-4.3%	0.9820
5	0.3831	3.5%	0.9713	0.5%	0.9642	0.3%	0.9824
6	0.4338	-9.3%	0.9639	-0.3%	0.9541	-0.8%	0.9758
7	0.4902	-23.5%	0.9464	-2.1%	0.9414	-2.1%	0.9747
8	0.3845	3.1%	0.9678	0.1%	0.9640	0.2%	0.9831
9	0.5846	-47.3%	0.9285	-3.9%	0.9167	-4.7%	0.9897
10	0.4157	-4.8%	0.9670	0.0%	0.9579	-0.4%	0.9675
Average	**0.4607**	**-16.1%**	**0.9566**	**-1.0%**	**0.9472**	**-1.5%**	**0.9735**

TABLE 9.2 *$NAPLS_{70}$ Results for Dataset 2*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.1865**	**0.0%**	**0.6613**	**0.0%**	**0.5547**	**0.0%**	
1	0.1942	-4.1%	0.6450	-2.5%	0.5175	-6.7%	0.9194
2	0.1935	-3.7%	0.6419	-2.9%	0.5207	-6.1%	0.9111
3	0.1869	-0.2%	0.6743	2.0%	0.5530	-0.3%	0.9247
4	0.1922	-3.0%	0.6504	-1.7%	0.5274	-4.9%	0.9273
5	0.2016	-8.1%	0.6514	-1.5%	0.4799	-13.5%	0.9222
6	0.1891	-1.4%	0.6412	-3.0%	0.5422	-2.2%	0.9163
7	0.1853	0.7%	0.6712	1.5%	0.5605	1.0%	0.9323
8	0.1880	-0.8%	0.6356	-3.9%	0.5478	-1.2%	0.9056
9	0.1975	-5.9%	0.6202	-6.2%	0.5006	-9.8%	0.9172
10	0.1937	-3.8%	0.6491	-1.8%	0.5199	-6.3%	0.9211
Average	**0.1922**	**-3.0%**	**0.6480**	**-2.0%**	**0.5269**	**-5.0%**	**0.9197**

TABLE 9.3 *$NAPLS_{70}$ Results for Dataset 3A*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.0349**	**0.0%**	**0.8810**	**0.0%**	**0.8707**	**0.0%**	**0**
1	0.1455	-317.2%	0.4962	-43.7%	-1.2516	NA	0.4593
2	0.1469	-321.2%	0.5400	-38.7%	-1.2950	NA	0.4914
3	0.1057	-203.2%	0.4163	-52.7%	-0.1894	NA	0.4564
4	0.1643	-371.3%	0.4025	-54.3%	-1.8724	NA	0.4801
5	0.1420	-307.4%	0.1793	-79.7%	-1.1464	NA	0.4097
6	0.1868	-435.8%	0.0183	-97.9%	-2.7130	NA	0.5197
7	0.1634	-368.7%	0.6728	-23.6%	-1.8407	NA	0.4818
8	0.1102	-216.0%	0.6212	-29.5%	-0.2919	NA	0.4318
9	0.1885	-440.5%	0.1185	-86.5%	-2.7791	NA	0.4864
10	0.2078	-495.9%	0.2229	-74.7%	-3.5933	NA	0.4740
Average	**0.1561**	**-347.7%**	**0.3688**	**-58.1%**	**NA**	**NA**	**0.4691**

TABLE 9.4 *$NAPLS_{70}$ Results for Dataset 3B*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.8311**	**0.0%**	**0.7926**	**0.0%**	**0.7073**	**0.0%**	
1	1.2545	-50.9%	0.6094	-23.1%	0.3332	-52.9%	0.7752
2	1.2444	-49.7%	0.5160	-34.9%	0.3438	-51.4%	0.8346
3	0.9131	-9.9%	0.7185	-9.3%	0.6467	-8.6%	0.7829
4	1.3589	-63.5%	0.5162	-34.9%	0.2175	-69.3%	0.8067
5	0.7727	7.0%	0.8046	1.5%	0.7470	5.6%	0.8609
6	1.4317	-72.3%	0.7258	-8.4%	0.1315	-81.4%	0.8269
7	1.1802	-42.0%	0.6919	-12.7%	0.4098	-42.1%	0.8252
8	0.7984	3.9%	0.7901	-0.3%	0.7299	3.2%	0.8168
9	1.2711	-52.9%	0.6765	-14.7%	0.3154	-55.4%	0.8744
10	0.9011	-8.4%	0.7622	-3.8%	0.6559	-7.3%	0.8143
Average	**1.1126**	**-33.9%**	**0.6811**	**-14.1%**	**0.4531**	**-35.9%**	**0.8218**

TABLE 9.5 *$NAPLS_{70}$ Results for Dataset 3C*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.9917**	**0.0%**	**0.8970**	**0.0%**	**0.8477**	**0.0%**	
1	1.9910	-100.8%	0.6043	-32.6%	0.3861	-54.5%	0.5967
2	2.0290	-104.6%	0.8693	-3.1%	0.3624	-57.2%	0.7249
3	1.9005	-91.6%	0.6962	-22.4%	0.4406	-48.0%	0.6683
4	1.2131	-22.3%	0.8377	-6.6%	0.7721	-8.9%	0.6723
5	1.2443	-25.5%	0.8506	-5.2%	0.7602	-10.3%	0.7700
6	1.7681	-78.3%	0.8776	-2.2%	0.5158	-39.1%	0.7451
7	1.8709	-88.7%	0.6742	-24.8%	0.4579	-46.0%	0.7394
8	2.1497	-116.8%	0.7352	-18.0%	0.2843	-66.5%	0.7125
9	1.5349	-54.8%	0.6802	-24.2%	0.6351	-25.1%	0.7140
10	2.2081	-122.7%	0.8829	-1.6%	0.2449	-71.1%	0.6919
Average	**1.7910**	**-80.6%**	**0.7708**	**-14.1%**	**0.4860**	**-42.7%**	**0.7035**

TABLE 9.6 *$NAPLS_{70}$ Results for Dataset 4*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.2905**	**0.0%**	**0.8468**	**0.0%**	**0.807809**	**0.0%**	
1	1.3376	-3.6%	0.7957	-6.0%	0.7935	-1.8%	0.8901
2	1.4792	-14.6%	0.7495	-11.5%	0.7475	-7.5%	0.8768
3	1.1900	7.8%	0.8519	0.6%	0.8366	3.6%	0.8630
4	1.2501	3.1%	0.8509	0.5%	0.8197	1.5%	0.8937
5	1.3007	-0.8%	0.8236	-2.7%	0.8048	-0.4%	0.9172
6	1.4516	-12.5%	0.8255	-2.5%	0.7568	-6.3%	0.8943
7	1.4925	-15.6%	0.7587	-10.4%	0.7430	-8.0%	0.8498
8	1.3940	-8.0%	0.8062	-4.8%	0.7758	-4.0%	0.8787
9	1.3829	-7.2%	0.8058	-4.8%	0.7793	-3.5%	0.8917
10	1.3571	-5.2%	0.8124	-4.1%	0.7875	-2.5%	0.8699
Average	**1.3636**	**-5.7%**	**0.8080**	**-4.6%**	**0.7844**	**-2.9%**	**0.8825**

TABLE 9.7 *$NAPLS_{70}$ Results for Dataset 5*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.7632**	**0.0%**	**0.5162**	**0.0%**	**0.5082**	**0.0%**	
1	1.8895	-7.2%	0.4584	-11.2%	0.4351	-14.4%	0.7858
2	1.8517	-5.0%	0.4792	-7.2%	0.4575	-10.0%	0.8139
3	1.8158	-3.0%	0.4996	-3.2%	0.4784	-5.9%	0.8208
4	1.8730	-6.2%	0.4824	-6.6%	0.4450	-12.4%	0.8285
5	1.8537	-5.1%	0.4776	-7.5%	0.4563	-10.2%	0.8548
6	1.9264	-9.3%	0.4667	-9.6%	0.4129	-18.8%	0.7737
7	1.9115	-8.4%	0.4508	-12.7%	0.4219	-17.0%	0.8313
8	1.8678	-5.9%	0.4927	-4.6%	0.4480	-11.8%	0.8218
9	1.9136	-8.5%	0.4639	-10.1%	0.4206	-17.2%	0.8241
10	1.8492	-4.9%	0.4800	-7.0%	0.4590	-9.7%	0.8545
Average	**1.8752**	**-6.4%**	**0.4751**	**-8.0%**	**0.4435**	**-12.7%**	**0.8209**

TABLE 9.8 *$NAPLS_{70}$ Results for Dataset 6*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2648**	**0.0%**	**0.7331**	**0.0%**	**0.7279**	**0.0%**	**0**
1	0.8331	-214.6%	0.3825	-47.8%	-1.6929	NA	0.3839
2	0.6987	-163.8%	0.1223	-83.3%	-0.8939	NA	0.4276
3	1.5772	-495.5%	0.2147	-70.7%	-8.6509	NA	0.4679
4	1.2209	-361.0%	0.1898	-74.1%	-4.7827	NA	0.4190
5	0.9805	-270.2%	0.3602	-50.9%	-2.7299	NA	0.3710
6	1.1017	-316.0%	0.4474	-39.0%	-3.7084	NA	0.3692
7	1.0502	-296.5%	0.0942	-87.2%	-3.2786	NA	0.4284
8	1.0409	-293.0%	0.1977	-73.0%	-3.2032	NA	0.4038
9	1.0558	-298.7%	0.2188	-70.2%	-3.3249	NA	0.3774
10	0.7644	-188.6%	0.2709	-63.0%	-1.2668	NA	0.4543
Average	**1.0323**	**-289.8%**	**0.2499**	**-65.9%**	**NA**	**NA**	**0.4102**

TABLE 9.9 *$NAPLS_{70}$ Results for Dataset 7*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2039**	**0.0%**	**0.7896**	**0.0%**	**0.7895**	**0.0%**	**0**
1	0.3446	-69.1%	0.6192	-21.6%	0.3983	-49.5%	0.9281
2	0.3634	-78.3%	0.6171	-21.8%	0.3310	-58.1%	0.8780
3	0.3263	-60.1%	0.7126	-9.8%	0.4606	-41.7%	0.7915
4	0.5898	-189.3%	0.5798	-26.6%	-0.7623	NA	0.9220
5	0.4016	-97.0%	0.6505	-17.6%	0.1830	-76.8%	0.8218
6	0.2896	-42.1%	0.6511	-17.5%	0.5751	-27.2%	0.8710
7	0.4594	-125.3%	0.6326	-19.9%	-0.0688	NA	0.9177
8	0.4237	-107.9%	0.6614	-16.2%	0.0906	-88.5%	0.8908
9	0.4985	-144.5%	0.5368	-32.0%	-0.2588	NA	0.9200
10	0.4026	-97.5%	0.6721	-14.9%	0.1791	-77.3%	0.8451
Average	**0.4100**	**-101.1%**	**0.6333**	**-19.8%**	**0.3168**	**NA**	**0.8786**

TABLE 9.10 *$NAPLS_{70}$ Results for Dataset 8*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.4516**	**0.0%**	**0.8764**	**0.0%**	**0.8728**	**0.0%**	**0**
1	3.9971	-175.4%	0.3231	-63.1%	0.0357	-95.9%	0.3924
2	4.6916	-223.2%	0.4323	-50.7%	-0.3285	NA	0.4917
3	4.0804	-181.1%	0.5737	-34.5%	-0.0049	NA	0.4078
4	4.7761	-229.0%	0.7103	-19.0%	-0.3768	NA	0.4013
5	5.2051	-258.6%	0.3899	-55.5%	-0.6353	NA	0.4404
6	4.1182	-183.7%	0.2974	-66.1%	-0.0236	NA	0.5715
7	5.2929	-264.6%	0.4584	-47.7%	-0.6909	NA	0.4026
8	4.3656	-200.7%	0.4494	-48.7%	-0.1503	NA	0.4838
9	4.8200	-232.0%	0.2866	-67.3%	-0.4022	NA	0.5192
10	4.7402	-226.6%	0.6164	-29.7%	-0.3562	NA	0.3456
Average	**4.6087**	**-217.5%**	**0.4537**	**-48.2%**	**NA**	**NA**	**0.4456**

TABLE 9.11 *$NAPLS_{70}$ Results for Dataset 9*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.9687**	**0.0%**	**0.7267**	**0.0%**	**0.7033**	**0.0%**	**0**
1	6.5943	-235.0%	0.4436	-39.0%	-2.3290	NA	0.3769
2	7.8674	-299.6%	0.2091	-71.2%	-3.7383	NA	0.3438
3	5.9906	-204.3%	0.3127	-57.0%	-1.7473	NA	0.3844
4	6.3080	-220.4%	0.2533	-65.1%	-2.0462	NA	0.3422
5	4.7220	-139.9%	0.3594	-50.5%	-0.7070	NA	0.3622
6	10.0240	-409.2%	0.0672	-90.8%	-6.6922	NA	0.4225
7	11.7090	-494.7%	0.0546	-92.5%	-9.4955	NA	0.3455
8	9.1976	-367.2%	0.1230	-83.1%	-5.4761	NA	0.3570
9	6.3650	-223.3%	0.3947	-45.7%	-2.1014	NA	0.4165
10	6.1476	-212.3%	0.2950	-59.4%	-1.8932	NA	0.2722
Average	**7.4926**	**-280.6%**	**0.2513**	**-65.4%**	**NA**	**NA**	**0.3623**

10 Addendum B: Detailed simulation output

10.1 Summary of NAPLS test results

The detailed output for the various test cases is given in the remaining subsections. These results are summarised in **TABLE 10.1** - **TABLE 10.3** below.

TABLE 10.1 *The effect of different parameters on the prediction error*

RMSE	NAPLS Parameters (Outer Loops x Inner Loops x Noise%)				
Dataset	**5x100x5%**	**5x100x10%**	**15x100x10%**	**5x300x10%**	**5x500x10%**
Dataset 1	0.5442	0.5531	0.5749	0.4686	0.4530
Dataset 2	0.2405	0.2435	0.2444	0.2476	0.2467
Dataset 3A	0.1295	0.1845	0.2590	0.2225	0.1920
Dataset 3B	1.0171	1.1340	1.1316	1.3290	1.1082
Dataset 3C	1.1488	1.7573	1.5724	1.7429	1.6504
Dataset 4	1.7071	2.5020	2.3643	2.4378	2.5549
Dataset 5	1.8136	1.8916	1.8806	1.8775	1.8986
Dataset 6	3.0329	6.3168	5.7684	5.5690	5.4206
Dataset 7	3.3464	6.8623	7.6446	7.2808	8.0418
Dataset 8	0.5961	1.0716	1.0828	1.1466	1.0582
Dataset 9	0.2658	0.4127	0.4286	0.3199	0.3362

TABLE 10.2 *The effect of different parameters on the correlation between actual and predicted values*

R^2	NAPLS Parameters (Outer Loops x Inner Loops x Noise%)				
Dataset	**5x100x5%**	**5x100x10%**	**15x100x10%**	**5x300x10%**	**5x500x10%**
Dataset 1	0.9510	0.9478	0.9506	0.9580	0.9586
Dataset 2	0.4452	0.4454	0.4328	0.4168	0.4365
Dataset 3A	0.4978	0.3863	0.3348	0.3864	0.4406
Dataset 3B	0.8564	0.8305	0.8008	0.7338	0.8082
Dataset 3C	0.8568	0.7260	0.7384	0.6970	0.6883
Dataset 4	0.6363	0.4496	0.4957	0.4634	0.4660
Dataset 5	0.5042	0.4750	0.4886	0.4829	0.4864
Dataset 6	0.6496	0.3573	0.4220	0.3511	0.3433
Dataset 7	0.5751	0.2708	0.2433	0.2326	0.2350
Dataset 8	0.2996	0.1615	0.1725	0.1215	0.2759
Dataset 9	0.7274	0.6143	0.6231	0.6849	0.6838

TABLE 10.3 *The effect of different parameters on the correlation between the selected model and the median model*

R^2_{median}	NAPLS Parameters (Outer Loops x Inner Loops x Noise%)				
Dataset	**5x100x5%**	**5x100x10%**	**15x100x10%**	**5x300x10%**	**5x500x10%**
Dataset 1	0.9917	0.9792	0.9864	0.9771	0.9833
Dataset 2	0.9592	0.8571	0.9055	0.8363	0.8233
Dataset 3A	0.5804	0.4781	0.4633	0.4342	0.4818
Dataset 3B	0.9402	0.8590	0.8826	0.8362	0.8134
Dataset 3C	0.8326	0.6792	0.7003	0.6867	0.6902
Dataset 4	0.7699	0.6372	0.6389	0.5935	0.5840
Dataset 5	0.9647	0.9056	0.9175	0.8719	0.8394
Dataset 6	0.5951	0.4913	0.4728	0.5010	0.4714
Dataset 7	0.5309	0.3772	0.3337	0.3832	0.4186
Dataset 8	0.4614	0.4306	0.3340	0.4137	0.4224
Dataset 9	0.9670	0.9223	0.9431	0.9321	0.9248

10.2 NAPLS Results: 5 Outer Loops; 100 Inner Loops; 5% Noise

TABLE 10.4 *NAPLS Results for Dataset 1*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.6684**	**0.0%**	**0.9400**	**0.0%**	**0.8977**	**0.0%**	
1	0.4423	33.8%	0.9586	2.0%	0.9552	6.4%	0.9941
2	0.5312	20.5%	0.9495	1.0%	0.9354	4.2%	0.9906
3	0.5497	17.8%	0.9489	0.9%	0.9308	3.7%	0.9882
4	0.6166	7.7%	0.9444	0.5%	0.9129	1.7%	0.9919
5	0.4966	25.7%	0.9513	1.2%	0.9435	5.1%	0.9951
6	0.6478	3.1%	0.9490	1.0%	0.9039	0.7%	0.9903
7	0.4721	29.4%	0.9535	1.4%	0.9489	5.7%	0.9927
8	0.4635	30.7%	0.9567	1.8%	0.9508	5.9%	0.9908
9	0.6016	10.0%	0.9499	1.0%	0.9171	2.2%	0.9905
10	0.6207	7.1%	0.9484	0.9%	0.9117	1.6%	0.9924
Average	**0.5442**	**18.6%**	**0.9510**	**1.2%**	**0.9310**	**3.7%**	**0.9917**

TABLE 10.5 *NAPLS Results for Dataset 2*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2387**	**0.0%**	**0.4514**	**0.0%**	**0.4496**	**0.0%**	
1	0.2377	0.4%	0.4558	1.0%	0.4539	0.9%	0.9679
2	0.2490	-4.3%	0.4054	-10.2%	0.4008	-10.9%	0.9522
3	0.2448	-2.6%	0.4252	-5.8%	0.4208	-6.4%	0.9474
4	0.2339	2.0%	0.4739	5.0%	0.4712	4.8%	0.9664
5	0.2389	-0.1%	0.4495	-0.4%	0.4485	-0.3%	0.9620
6	0.2419	-1.4%	0.4468	-1.0%	0.4346	-3.4%	0.9589
7	0.2390	-0.1%	0.4510	-0.1%	0.4482	-0.3%	0.9630
8	0.2386	0.0%	0.4513	0.0%	0.4501	0.1%	0.9561
9	0.2410	-1.0%	0.4462	-1.1%	0.4389	-2.4%	0.9639
10	0.2403	-0.7%	0.4473	-0.9%	0.4422	-1.7%	0.9542
Average	**0.2405**	**-0.8%**	**0.4452**	**-1.4%**	**0.4409**	**-1.9%**	**0.9592**

TABLE 10.6 *NAPLS Results for Dataset 3A*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.0739**	**0.0%**	**0.7580**	**0.0%**	**0.6867**	**0.0%**	**0**
1	0.1578	-113.5%	0.7810	3.0%	-0.4285	NA	0.6058
2	0.1296	-75.3%	0.3496	-53.9%	0.0371	-94.6%	0.6186
3	0.1018	-37.7%	0.5732	-24.4%	0.4058	-40.9%	0.5924
4	0.1043	-41.2%	0.5554	-26.7%	0.3756	-45.3%	0.6856
5	0.1640	-121.9%	0.1795	-76.3%	-0.5425	NA	0.5513
6	0.1530	-107.0%	0.1380	-81.8%	-0.3422	NA	0.5956
7	0.1224	-65.6%	0.6797	-10.3%	0.1403	-79.6%	0.6155
8	0.1441	-94.9%	0.4701	-38.0%	-0.1905	NA	0.4208
9	0.1442	-95.2%	0.4920	-35.1%	-0.1933	NA	0.5084
10	0.0736	0.4%	0.7595	0.2%	0.6891	0.4%	0.6097
Average*	**0.1295**	**-75.2%**	**0.4978**	**-34.3%**	**0.3296**	**-52.0%**	**0.5804**

** Negative values of R^2_{pred} excluded from average*

TABLE 10.7 *NAPLS Results for Dataset 3B*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.7135**	**0.0%**	**0.9179**	**0.0%**	**0.9172**	**0.0%**	
1	0.7977	-11.8%	0.9051	-1.4%	0.8965	-2.3%	0.9403
2	0.9025	-26.5%	0.9010	-1.8%	0.8675	-5.4%	0.9529
3	1.0765	-50.9%	0.8294	-9.6%	0.8116	-11.5%	0.9549
4	1.0311	-44.5%	0.8871	-3.4%	0.8271	-9.8%	0.9246
5	0.7704	-8.0%	0.9134	-0.5%	0.9035	-1.5%	0.9458
6	1.0746	-50.6%	0.8182	-10.9%	0.8122	-11.4%	0.9226
7	1.3035	-82.7%	0.7535	-17.9%	0.7237	-21.1%	0.9194
8	0.9294	-30.3%	0.8814	-4.0%	0.8595	-6.3%	0.9602
9	1.2037	-68.7%	0.8298	-9.6%	0.7644	-16.7%	0.9298
10	1.0815	-51.6%	0.8454	-7.9%	0.8098	-11.7%	0.9513
Average	**1.0171**	**-42.6%**	**0.8564**	**-6.7%**	**0.8276**	**-9.8%**	**0.9402**

TABLE 10.8 *NAPLS Results for Dataset 3C*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.0804**	**0.0%**	**0.9192**	**0.0%**	**0.8423**	**0.0%**	
1	1.1511	-6.5%	0.8998	-2.1%	0.8210	-2.5%	0.8691
2	1.2860	-19.0%	0.7792	-15.2%	0.7766	-7.8%	0.7615
3	0.7180	33.5%	0.9384	2.1%	0.9304	10.5%	0.8562
4	1.2210	-13.0%	0.8348	-9.2%	0.7986	-5.2%	0.8474
5	1.1690	-8.2%	0.8657	-5.8%	0.8154	-3.2%	0.8450
6	1.0055	6.9%	0.9210	0.2%	0.8634	2.5%	0.7972
7	0.9588	11.3%	0.9069	-1.3%	0.8758	4.0%	0.8024
8	0.9829	9.0%	0.8984	-2.3%	0.8695	3.2%	0.8429
9	1.8645	-72.6%	0.6174	-32.8%	0.5304	-37.0%	0.8778
10	1.1316	-4.7%	0.9069	-1.3%	0.8270	-1.8%	0.8268
Average	**1.1488**	**-6.3%**	**0.8568**	**-6.8%**	**0.8108**	**-3.7%**	**0.8326**

TABLE 10.9 *NAPLS Results for Dataset 4*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5088**	**0.0%**	**0.6874**	**0.0%**	**0.6751839**	**0.0%**	
1	1.6398	-8.7%	0.6689	-2.7%	0.6164	-8.7%	0.7866
2	1.8306	-21.3%	0.5886	-14.4%	0.5219	-22.7%	0.7956
3	1.8036	-19.5%	0.6247	-9.1%	0.5359	-20.6%	0.8005
4	1.6916	-12.1%	0.6217	-9.6%	0.5917	-12.4%	0.7493
5	1.3910	7.8%	0.7357	7.0%	0.7239	7.2%	0.7432
6	1.7844	-18.3%	0.6176	-10.2%	0.5457	-19.2%	0.7729
7	1.6808	-11.4%	0.6244	-9.2%	0.5969	-11.6%	0.7815
8	1.7191	-13.9%	0.6023	-12.4%	0.5783	-14.3%	0.7545
9	2.1145	-40.1%	0.5548	-19.3%	0.3621	-46.4%	0.6960
10	1.4154	6.2%	0.7237	5.3%	0.7142	5.8%	0.8190
Average	**1.7071**	**-13.1%**	**0.6363**	**-7.4%**	**0.5787**	**-14.3%**	**0.7699**

TABLE 10.10 *NAPLS Results for Dataset 5*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.7958**	**0.0%**	**0.5048**	**0.0%**	**0.4698**	**0.0%**	
1	1.7983	-0.1%	0.5026	-0.4%	0.4684	-0.3%	0.9574
2	1.8188	-1.3%	0.5018	-0.6%	0.4561	-2.9%	0.9687
3	1.7819	0.8%	0.5199	3.0%	0.4780	1.7%	0.9713
4	1.7917	0.2%	0.5167	2.3%	0.4722	0.5%	0.9637
5	1.8336	-2.1%	0.5060	0.2%	0.4473	-4.8%	0.9648
6	1.8356	-2.2%	0.5070	0.4%	0.4460	-5.1%	0.9714
7	1.7980	-0.1%	0.5026	-0.4%	0.4685	-0.3%	0.9503
8	1.8286	-1.8%	0.4927	-2.4%	0.4503	-4.2%	0.9645
9	1.7749	1.2%	0.5260	4.2%	0.4821	2.6%	0.9662
10	1.8749	-4.4%	0.4666	-7.6%	0.4221	-10.2%	0.9684
Average	**1.8136**	**-1.0%**	**0.5042**	**-0.1%**	**0.4591**	**-2.3%**	**0.9647**

TABLE 10.11 *NAPLS Results for Dataset 6*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5425**	**0.0%**	**0.8646**	**0.0%**	**0.8564036**	**0.0%**	
1	3.3993	-120.4%	0.6094	-29.5%	0.3026	-64.7%	0.6341
2	3.2796	-112.6%	0.5976	-30.9%	0.3508	-59.0%	0.6249
3	2.1322	-38.2%	0.7414	-14.2%	0.7256	-15.3%	0.6494
4	3.2972	-113.8%	0.6610	-23.6%	0.3438	-59.9%	0.6819
5	3.7871	-145.5%	0.5790	-33.0%	0.1344	-84.3%	0.5462
6	2.7681	-79.5%	0.7258	-16.0%	0.5375	-37.2%	0.4987
7	2.6453	-71.5%	0.7273	-15.9%	0.5777	-32.5%	0.5810
8	3.0916	-100.4%	0.6095	-29.5%	0.4231	-50.6%	0.5453
9	3.0335	-96.7%	0.6327	-26.8%	0.4446	-48.1%	0.5869
10	2.8953	-87.7%	0.6121	-29.2%	0.4941	-42.3%	0.6029
Average	**3.0329**	**-96.6%**	**0.6496**	**-24.9%**	**0.4334**	**-49.4%**	**0.5951**

TABLE 10.12 *NAPLS Results for Dataset 7*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.6313**	**0.0%**	**0.8035**	**0.0%**	**0.7963**	**0.0%**	**0**
1	2.6108	-60.0%	0.6693	-16.7%	0.4782	-39.9%	0.4812
2	2.8805	-76.6%	0.6957	-13.4%	0.3648	-54.2%	0.4975
3	3.5174	-115.6%	0.5545	-31.0%	0.0529	-93.4%	0.5457
4	3.8398	-135.4%	0.4338	-46.0%	-0.1287	NA	0.4362
5	4.3411	-166.1%	0.6202	-22.8%	-0.4427	NA	0.5286
6	2.7482	-68.5%	0.6871	-14.5%	0.4218	-47.0%	0.5513
7	5.2373	-221.0%	0.2787	-65.3%	-1.0998	NA	0.5242
8	2.2399	-37.3%	0.6476	-19.4%	0.6159	-22.7%	0.5344
9	3.3763	-107.0%	0.5104	-36.5%	0.1273	-84.0%	0.5418
10	2.6730	-63.9%	0.6535	-18.7%	0.4530	-43.1%	0.6681
Average*	**3.3464**	**-105.1%**	**0.5751**	**-28.4%**	**0.3591**	**-54.9%**	**0.5309**

TABLE 10.13 *NAPLS Results for Dataset 8*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2960**	**0.0%**	**0.6652**	**0.0%**	**0.6600**	**0.0%**	**0**
1	0.5047	-70.5%	0.3855	-42.1%	0.0116	-98.2%	0.4854
2	0.6591	-122.7%	0.2705	-59.3%	-0.6855	NA	0.4268
3	0.6886	-132.6%	0.0985	-85.2%	-0.8397	NA	0.3909
4	0.6253	-111.2%	0.3294	-50.5%	-0.5170	NA	0.5262
5	0.6670	-125.3%	0.2265	-65.9%	-0.7260	NA	0.5352
6	0.5740	-93.9%	0.2928	-56.0%	-0.2784	NA	0.4525
7	0.7029	-137.4%	0.2608	-60.8%	-0.9166	NA	0.5465
8	0.5388	-82.0%	0.3781	-43.2%	-0.1264	NA	0.3551
9	0.4810	-62.5%	0.3844	-42.2%	0.1025	-84.5%	0.4578
10	0.5196	-75.5%	0.3699	-44.4%	-0.0474	NA	0.4380
Average*	**0.5961**	**-101.4%**	**0.2996**	**-55.0%**	**NA**	**NA**	**0.4614**

TABLE 10.14 *NAPLS Results for Dataset 9*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2061**	**0.0%**	**0.7850**	**0.0%**	**0.7848**	**0.0%**	
1	0.2874	-39.4%	0.7114	-9.4%	0.5816	-25.9%	0.9540
2	0.2327	-12.9%	0.7654	-2.5%	0.7258	-7.5%	0.9594
3	0.2578	-25.1%	0.7628	-2.8%	0.6634	-15.5%	0.9849
4	0.2355	-14.2%	0.7521	-4.2%	0.7191	-8.4%	0.9511
5	0.2920	-41.7%	0.7489	-4.6%	0.5682	-27.6%	0.9569
6	0.2287	-10.9%	0.7515	-4.3%	0.7351	-6.3%	0.9756
7	0.2520	-22.3%	0.7111	-9.4%	0.6783	-13.6%	0.9690
8	0.3048	-47.8%	0.7257	-7.6%	0.5295	-32.5%	0.9804
9	0.3225	-56.5%	0.5948	-24.2%	0.4732	-39.7%	0.9624
10	0.2448	-18.8%	0.7507	-4.4%	0.6964	-11.3%	0.9761
Average	**0.2658**	**-29.0%**	**0.7274**	**-7.3%**	**0.6371**	**-18.8%**	**0.9670**

10.3 NAPLS Results: 5 Outer Loops; 100 Inner Loops; 10% Noise

TABLE 10.15 *NAPLS Results for Dataset 1*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.6684**	**0.0%**	**0.9400**	**0.0%**	**0.8977**	**0.0%**	
1	0.5442	18.6%	0.9459	0.6%	0.9322	3.8%	0.9781
2	0.5982	10.5%	0.9419	0.2%	0.9180	2.3%	0.9680
3	0.8157	-22.0%	0.9448	0.5%	0.8476	-5.6%	0.9807
4	0.5105	23.6%	0.9529	1.4%	0.9403	4.8%	0.9728
5	0.5588	16.4%	0.9286	-1.2%	0.9285	3.4%	0.9866
6	0.4585	31.4%	0.9534	1.4%	0.9518	6.0%	0.9641
7	0.5838	12.7%	0.9491	1.0%	0.9219	2.7%	0.9903
8	0.5281	21.0%	0.9411	0.1%	0.9361	4.3%	0.9853
9	0.4813	28.0%	0.9585	2.0%	0.9469	5.5%	0.9851
10	0.4518	32.4%	0.9614	2.3%	0.9532	6.2%	0.9807
Average	**0.5531**	**17.2%**	**0.9478**	**0.8%**	**0.9277**	**3.3%**	**0.9792**

TABLE 10.16 *NAPLS Results for Dataset 2*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2387**	**0.0%**	**0.4514**	**0.0%**	**0.4496**	**0.0%**	
1	0.2581	-8.1%	0.3820	-15.4%	0.3563	-20.8%	0.8516
2	0.2417	-1.3%	0.4409	-2.3%	0.4353	-3.2%	0.8633
3	0.2674	-12.0%	0.3417	-24.3%	0.3091	-31.3%	0.8790
4	0.2415	-1.2%	0.4384	-2.9%	0.4364	-2.9%	0.8388
5	0.2436	-2.1%	0.4325	-4.2%	0.4266	-5.1%	0.8550
6	0.2354	1.4%	0.5172	14.6%	0.4647	3.3%	0.8596
7	0.2373	0.6%	0.4606	2.0%	0.4560	1.4%	0.8680
8	0.2310	3.2%	0.5057	12.0%	0.4846	7.8%	0.8579
9	0.2457	-2.9%	0.4224	-6.4%	0.4169	-7.3%	0.8247
10	0.2329	2.4%	0.5124	13.5%	0.4759	5.8%	0.8734
Average	**0.2435**	**-2.0%**	**0.4454**	**-1.3%**	**0.4262**	**-5.2%**	**0.8571**

TABLE 10.17 *NAPLS Results for Dataset 3A*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.0739**	**0.0%**	**0.7580**	**0.0%**	**0.6867**	**0.0%**	**0**
1	0.1394	-88.6%	0.2093	-72.4%	-0.1148	NA	0.5618
2	0.2159	-192.0%	0.1127	-85.1%	-1.6721	NA	0.3922
3	0.2105	-184.8%	0.4298	-43.3%	-1.5422	NA	0.4610
4	0.1982	-168.2%	0.6854	-9.6%	-1.2532	NA	0.4882
5	0.1466	-98.4%	0.5381	-29.0%	-0.2330	NA	0.5222
6	0.1910	-158.5%	0.4015	-47.0%	-1.0929	NA	0.4336
7	0.2547	-244.6%	0.2697	-64.4%	-2.7212	NA	0.4297
8	0.2283	-208.8%	0.4485	-40.8%	-1.9887	NA	0.4278
9	0.1298	-75.6%	0.2451	-67.7%	0.0344	-95.0%	0.4257
10	0.1301	-76.1%	0.5232	-31.0%	0.0287	-95.8%	0.6386
Average	**0.1845**	**-149.6%**	**0.3863**	**-49.0%**	**NA**	**NA**	**0.4781**

TABLE 10.18 *NAPLS Results for Dataset 3B*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.7135**	**0.0%**	**0.9179**	**0.0%**	**0.9172**	**0.0%**	
1	0.9873	-38.4%	0.8687	-5.4%	0.8415	-8.3%	0.8544
2	1.0211	-43.1%	0.8886	-3.2%	0.8305	-9.5%	0.8522
3	1.8895	-164.8%	0.6966	-24.1%	0.4195	-54.3%	0.8475
4	0.9687	-35.8%	0.8515	-7.2%	0.8474	-7.6%	0.8114
5	1.1908	-66.9%	0.7833	-14.7%	0.7694	-16.1%	0.8712
6	1.0825	-51.7%	0.8828	-3.8%	0.8095	-11.7%	0.8685
7	1.0745	-50.6%	0.8210	-10.6%	0.8122	-11.4%	0.9208
8	0.9260	-29.8%	0.8673	-5.5%	0.8606	-6.2%	0.8297
9	1.0945	-53.4%	0.8228	-10.4%	0.8052	-12.2%	0.8557
10	1.1051	-54.9%	0.8226	-10.4%	0.8014	-12.6%	0.8784
Average	**1.1340**	**-58.9%**	**0.8305**	**-9.5%**	**0.7797**	**-15.0%**	**0.8590**

TABLE 10.19 *NAPLS Results for Dataset 3C*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.0804**	**0.0%**	**0.9192**	**0.0%**	**0.8423**	**0.0%**	
1	2.4270	-124.6%	0.5914	-35.7%	0.2043	-75.7%	0.6265
2	1.6247	-50.4%	0.8316	-9.5%	0.6434	-23.6%	0.6654
3	1.7832	-65.0%	0.6412	-30.2%	0.5705	-32.3%	0.6337
4	1.6371	-51.5%	0.8792	-4.3%	0.6380	-24.3%	0.6865
5	1.6168	-49.6%	0.6484	-29.5%	0.6469	-23.2%	0.6149
6	1.2475	-15.5%	0.8666	-5.7%	0.7898	-6.2%	0.8380
7	1.7334	-60.4%	0.6850	-25.5%	0.5941	-29.5%	0.7405
8	1.8260	-69.0%	0.6622	-28.0%	0.5496	-34.8%	0.6554
9	1.7907	-65.7%	0.6547	-28.8%	0.5669	-32.7%	0.6568
10	1.8865	-74.6%	0.7999	-13.0%	0.5193	-38.4%	0.6742
Average	**1.7573**	**-62.6%**	**0.7260**	**-21.0%**	**0.5723**	**-32.1%**	**0.6792**

TABLE 10.20 *NAPLS Results for Dataset 4*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5088**	**0.0%**	**0.6874**	**0.0%**	**0.6751839**	**0.0%**	**0**
1	2.6426	-75.1%	0.4392	-36.1%	0.0037	-99.5%	0.6071
2	2.2990	-52.4%	0.5404	-21.4%	0.2459	-63.6%	0.6944
3	3.2694	-116.7%	0.2578	-62.5%	-0.5250	NA	0.6088
4	2.4800	-64.4%	0.5621	-18.2%	0.1225	-81.9%	0.7006
5	1.7216	-14.1%	0.6012	-12.5%	0.5771	-14.5%	0.7334
6	1.7926	-18.8%	0.6317	-8.1%	0.5415	-19.8%	0.5959
7	2.5698	-70.3%	0.3737	-45.6%	0.0578	-91.4%	0.5958
8	2.5995	-72.3%	0.3767	-45.2%	0.0359	-94.7%	0.5951
9	2.6041	-72.6%	0.4018	-41.5%	0.0325	-95.2%	0.6376
10	3.0417	-101.6%	0.3117	-54.7%	-0.3200	NA	0.6030
Average*	**2.5020**	**-65.8%**	**0.4496**	**-34.6%**	**0.2021**	**-70.1%**	**0.6372**

** Negative values of R^2_{pred} excluded from average*

TABLE 10.21 *NAPLS Results for Dataset 5*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.7958**	**0.0%**	**0.5048**	**0.0%**	**0.4698**	**0.0%**	
1	1.8401	-2.5%	0.4830	-4.3%	0.4434	-5.6%	0.8847
2	1.9093	-6.3%	0.4573	-9.4%	0.4007	-14.7%	0.9125
3	1.8424	-2.6%	0.4942	-2.1%	0.4419	-5.9%	0.9051
4	1.8957	-5.6%	0.4609	-8.7%	0.4092	-12.9%	0.8994
5	1.9338	-7.7%	0.4609	-8.7%	0.3852	-18.0%	0.8921
6	1.9857	-10.6%	0.4487	-11.1%	0.3518	-25.1%	0.9025
7	1.8654	-3.9%	0.4880	-3.3%	0.4279	-8.9%	0.9254
8	1.8323	-2.0%	0.5038	-0.2%	0.4480	-4.6%	0.9104
9	1.9014	-5.9%	0.4724	-6.4%	0.4056	-13.7%	0.9131
10	1.9102	-6.4%	0.4808	-4.8%	0.4001	-14.8%	0.9111
Average	**1.8916**	**-5.3%**	**0.4750**	**-5.9%**	**0.4114**	**-12.4%**	**0.9056**

TABLE 10.22 *NAPLS Results for Dataset 6*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5425**	**0.0%**	**0.8646**	**0.0%**	**0.8564**	**0.0%**	
1	6.4724	-319.6%	0.3302	-61.8%	-1.5285	NA	0.5278
2	6.0739	-293.8%	0.2082	-75.9%	-1.2267	NA	0.5808
3	5.9551	-286.1%	0.5234	-39.5%	-1.1404	NA	0.5980
4	5.9184	-283.7%	0.0392	-95.5%	-1.1141	NA	0.4007
5	7.2492	-370.0%	0.5149	-40.4%	-2.1718	NA	0.3717
6	5.8878	-281.7%	0.5148	-40.5%	-1.0923	NA	0.4758
7	5.4300	-252.0%	0.6083	-29.6%	-0.7796	NA	0.3725
8	7.3259	-375.0%	0.1653	-80.9%	-2.2393	NA	0.4788
9	6.0524	-292.4%	0.4541	-47.5%	-1.2110	NA	0.6251
10	6.8027	-341.0%	0.2143	-75.2%	-1.7931	NA	0.4818
Average	**6.3168**	**-309.5%**	**0.3573**	**-58.7%**	**NA**	**NA**	**0.4913**

TABLE 10.23 *NAPLS Results for Dataset 7*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.6313**	**0.0%**	**0.8035**	**0.0%**	**0.7963**	**0.0%**	
1	7.7605	-375.7%	0.4049	-49.6%	-3.6104	NA	0.3657
2	8.2854	-407.9%	0.0960	-88.0%	-4.2552	NA	0.3713
3	4.7363	-190.3%	0.3381	-57.9%	-0.7173	NA	0.3381
4	6.1384	-276.3%	0.4684	-41.7%	-1.8846	NA	0.3906
5	4.9040	-200.6%	0.3083	-61.6%	-0.8411	NA	0.4342
6	6.7199	-311.9%	0.3635	-54.8%	-2.4570	NA	0.3218
7	10.1004	-519.2%	0.0257	-96.8%	-6.8098	NA	0.4010
8	6.2961	-286.0%	0.2918	-63.7%	-2.0347	NA	0.2982
9	5.7498	-252.5%	0.3285	-59.1%	-1.5309	NA	0.4603
10	7.9318	-386.2%	0.0830	-89.7%	-3.8163	NA	0.3906
Average	**6.8623**	**-320.7%**	**0.2708**	**-66.3%**	**NA**	**NA**	**0.3772**

TABLE 10.24 *NAPLS Results for Dataset 8*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2960**	**0.0%**	**0.6652**	**0.0%**	**0.6600**	**0.0%**	
1	1.0038	-239.1%	0.1250	-81.2%	-2.9089	NA	0.4266
2	0.7062	-138.6%	0.2730	-59.0%	-0.9349	NA	0.4654
3	1.0748	-263.1%	0.1477	-77.8%	-3.4819	NA	0.4036
4	1.3748	-364.4%	0.0623	-90.6%	-6.3331	NA	0.4111
5	1.2384	-318.4%	0.1032	-84.5%	-4.9500	NA	0.4356
6	1.0508	-255.0%	0.2576	-61.3%	-3.2835	NA	0.4468
7	1.1758	-297.2%	0.2151	-67.7%	-4.3636	NA	0.4722
8	1.1348	-283.4%	0.1121	-83.2%	-3.9960	NA	0.4023
9	0.7835	-164.7%	0.1861	-72.0%	-1.3816	NA	0.4114
10	1.1731	-296.3%	0.1332	-80.0%	-4.3393	NA	0.4309
Average	**1.0716**	**-262.0%**	**0.1615**	**-75.7%**	**NA**	**NA**	**0.4306**

TABLE 10.25 *NAPLS Results for Dataset 9*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2061**	**0.0%**	**0.7850**	**0.0%**	**0.7848**	**0.0%**	
1	0.3975	-92.8%	0.4817	-38.6%	0.1997	-74.5%	0.8555
2	0.5563	-169.9%	0.5446	-30.6%	-0.5675	NA	0.9258
3	0.3965	-92.4%	0.6081	-22.5%	0.2036	-74.1%	0.8873
4	0.5593	-171.4%	0.6778	-13.7%	-0.5848	NA	0.9505
5	0.3088	-49.8%	0.6376	-18.8%	0.5171	-34.1%	0.9405
6	0.4089	-98.4%	0.6856	-12.7%	0.1530	-80.5%	0.9068
7	0.3456	-67.7%	0.6118	-22.1%	0.3949	-49.7%	0.9455
8	0.3743	-81.6%	0.6850	-12.7%	0.2905	-63.0%	0.9596
9	0.4014	-94.7%	0.5772	-26.5%	0.1840	-76.6%	0.9198
10	0.3782	-83.5%	0.6342	-19.2%	0.2755	-64.9%	0.9317
Average	**0.4127**	**-100.2%**	**0.6143**	**-21.7%**	**0.1066**	**-64.7%**	**0.9223**

10.4 NAPLS Results: 15 Outer Loops; 100 Inner Loops; 10% Noise

TABLE 10.26 *NAPLS Results for Dataset 1*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.6684**	**0.0%**	**0.9400**	**0.0%**	**0.8977**	**0.0%**	
1	0.5469	18.2%	0.9442	0.4%	0.9315	3.8%	0.9855
2	0.7902	-18.2%	0.9535	1.4%	0.8570	-4.5%	0.9864
3	0.4528	32.2%	0.9583	1.9%	0.9530	6.2%	0.9888
4	0.7363	-10.2%	0.9450	0.5%	0.8758	-2.4%	0.9854
5	0.5523	17.4%	0.9468	0.7%	0.9301	3.6%	0.9772
6	0.5159	22.8%	0.9539	1.5%	0.9390	4.6%	0.9861
7	0.4762	28.8%	0.9584	1.9%	0.9481	5.6%	0.9859
8	0.5394	19.3%	0.9395	-0.1%	0.9334	4.0%	0.9896
9	0.4565	31.7%	0.9565	1.8%	0.9523	6.1%	0.9913
10	0.6824	-2.1%	0.9499	1.1%	0.8933	-0.5%	0.9876
Average	**0.5749**	**14.0%**	**0.9506**	**1.1%**	**0.9214**	**2.6%**	**0.9864**

TABLE 10.27 *NAPLS Results for Dataset 2*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2387**	**0.0%**	**0.4514**	**0.0%**	**0.4496**	**0.0%**	
1	0.2503	-4.9%	0.4184	-7.3%	0.3948	-12.2%	0.8893
2	0.2254	5.5%	0.5157	14.2%	0.5090	13.2%	0.9028
3	0.2591	-8.6%	0.3748	-17.0%	0.3513	-21.9%	0.9137
4	0.2419	-1.4%	0.4411	-2.3%	0.4346	-3.3%	0.9103
5	0.2460	-3.1%	0.4270	-5.4%	0.4153	-7.6%	0.8911
6	0.2506	-5.0%	0.3983	-11.8%	0.3933	-12.5%	0.9121
7	0.2558	-7.2%	0.3788	-16.1%	0.3675	-18.3%	0.8997
8	0.2400	-0.5%	0.4448	-1.4%	0.4436	-1.3%	0.9130
9	0.2393	-0.3%	0.4569	1.2%	0.4468	-0.6%	0.9404
10	0.2353	1.4%	0.4721	4.6%	0.4649	3.4%	0.8826
Average	**0.2444**	**-2.4%**	**0.4328**	**-4.1%**	**0.4221**	**-6.1%**	**0.9055**

TABLE 10.28 *NAPLS Results for Dataset 3A*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.0739**	**0.0%**	**0.7580**	**0.0%**	**0.6867**	**0.0%**	**0**
1	0.2093	-183.2%	0.4619	-39.1%	-1.5124	NA	0.3571
2	0.3206	-333.7%	0.3060	-59.6%	-4.8933	NA	0.5674
3	0.3896	-427.1%	0.5705	-24.7%	-7.7040	NA	0.4320
4	0.2035	-175.4%	0.7388	-2.5%	-1.3759	NA	0.4042
5	0.1779	-140.7%	0.1629	-78.5%	-0.8149	NA	0.4673
6	0.1862	-151.9%	0.0902	-88.1%	-0.9887	NA	0.3893
7	0.3302	-346.8%	0.1611	-78.7%	-5.2545	NA	0.4481
8	0.2967	-301.3%	0.4125	-45.6%	-4.0469	NA	0.5315
9	0.2279	-208.3%	0.0002	-100.0%	-1.9782	NA	0.5174
10	0.2485	-236.2%	0.4442	-41.4%	-2.5421	NA	0.5182
Average	**0.2590**	**-250.5%**	**0.3348**	**-55.8%**	**NA**	**NA**	**0.4633**

TABLE 10.29 *NAPLS Results for Dataset 3B*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.7135**	**0.0%**	**0.9179**	**0.0%**	**0.9172**	**0.0%**	
1	0.9949	-39.4%	0.8448	-8.0%	0.8391	-8.5%	0.8809
2	0.8392	-17.6%	0.8888	-3.2%	0.8855	-3.5%	0.8763
3	1.3067	-83.1%	0.7674	-16.4%	0.7224	-21.2%	0.8899
4	1.4826	-107.8%	0.6581	-28.3%	0.6426	-29.9%	0.8821
5	1.1126	-55.9%	0.8085	-11.9%	0.7987	-12.9%	0.8909
6	1.1279	-58.1%	0.8337	-9.2%	0.7931	-13.5%	0.8864
7	0.9338	-30.9%	0.8863	-3.4%	0.8582	-6.4%	0.9081
8	1.3746	-92.7%	0.6991	-23.8%	0.6928	-24.5%	0.8920
9	0.7217	-1.2%	0.9222	0.5%	0.9153	-0.2%	0.8639
10	1.4225	-99.4%	0.6991	-23.8%	0.6710	-26.8%	0.8560
Average	**1.1316**	**-58.6%**	**0.8008**	**-12.8%**	**0.7819**	**-14.8%**	**0.8826**

TABLE 10.30 *NAPLS Results for Dataset 3C*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.0804**	**0.0%**	**0.9192**	**0.0%**	**0.8423**	**0.0%**	
1	1.8277	-69.2%	0.6932	-24.6%	0.5488	-34.9%	0.7010
2	1.0512	2.7%	0.8937	-2.8%	0.8507	1.0%	0.6711
3	1.9260	-78.3%	0.5343	-41.9%	0.4989	-40.8%	0.6763
4	1.8303	-69.4%	0.7540	-18.0%	0.5475	-35.0%	0.7160
5	1.5431	-42.8%	0.7429	-19.2%	0.6784	-19.5%	0.7241
6	2.0384	-88.7%	0.7359	-19.9%	0.4387	-47.9%	0.6292
7	1.2370	-14.5%	0.8262	-10.1%	0.7933	-5.8%	0.7629
8	2.1024	-94.6%	0.5073	-44.8%	0.4029	-52.2%	0.7414
9	1.0084	6.7%	0.8724	-5.1%	0.8626	2.4%	0.7163
10	1.1594	-7.3%	0.8236	-10.4%	0.8184	-2.8%	0.6642
Average	**1.5724**	**-45.5%**	**0.7384**	**-19.7%**	**0.6440**	**-23.5%**	**0.7003**

TABLE 10.31 *NAPLS Results for Dataset 4*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5088**	**0.0%**	**0.6874**	**0.0%**	**0.6752**	**0.0%**	
1	2.9228	-93.7%	0.4079	-40.7%	-0.2189	NA	0.6338
2	2.3024	-52.6%	0.5453	-20.7%	0.2437	-63.9%	0.6450
3	2.6523	-75.8%	0.5078	-26.1%	-0.0037	NA	0.6738
4	2.2632	-50.0%	0.5455	-20.6%	0.2692	-60.1%	0.6148
5	1.8924	-25.4%	0.6103	-11.2%	0.4891	-27.6%	0.6752
6	2.8332	-87.8%	0.3511	-48.9%	-0.1453	NA	0.6130
7	2.3029	-52.6%	0.5254	-23.6%	0.2433	-64.0%	0.6075
8	2.1427	-42.0%	0.4491	-34.7%	0.3449	-48.9%	0.6398
9	1.8601	-23.3%	0.6203	-9.8%	0.5063	-25.0%	0.6388
10	2.4708	-63.8%	0.3942	-42.7%	0.1290	-80.9%	0.6472
Average*	**2.3643**	**-56.7%**	**0.4957**	**-27.9%**	**0.3179**	**-52.9%**	**0.6389**

** Negative values of R^2_{pred} excluded from average*

TABLE 10.32 *NAPLS Results for Dataset 5*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.7958**	**0.0%**	**0.5048**	**0.0%**	**0.4698**	**0.0%**	
1	1.8745	-4.4%	0.4899	-3.0%	0.4223	-10.1%	0.9289
2	1.9315	-7.6%	0.4734	-6.2%	0.3866	-17.7%	0.9175
3	1.9376	-7.9%	0.4561	-9.7%	0.3828	-18.5%	0.9156
4	1.8329	-2.1%	0.4957	-1.8%	0.4477	-4.7%	0.9094
5	1.8243	-1.6%	0.4961	-1.7%	0.4529	-3.6%	0.9162
6	1.9001	-5.8%	0.5149	2.0%	0.4064	-13.5%	0.9182
7	1.7810	0.8%	0.5130	1.6%	0.4785	1.9%	0.9281
8	1.9191	-6.9%	0.4795	-5.0%	0.3945	-16.0%	0.9140
9	1.9385	-7.9%	0.4628	-8.3%	0.3822	-18.7%	0.9088
10	1.8661	-3.9%	0.5042	-0.1%	0.4275	-9.0%	0.9178
Average	**1.8806**	**-4.7%**	**0.4886**	**-3.2%**	**0.4181**	**-11.0%**	**0.9175**

TABLE 10.33 *NAPLS Results for Dataset 6*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5425**	**0.0%**	**0.8646**	**0.0%**	**0.8564**	**0.0%**	
1	5.4341	-252.3%	0.4897	-43.4%	-0.7823	NA	0.4765
2	4.4978	-191.6%	0.5161	-40.3%	-0.2210	NA	0.3890
3	6.1733	-300.2%	0.3216	-62.8%	-1.3001	NA	0.4803
4	5.1216	-232.0%	0.3736	-56.8%	-0.5832	NA	0.5514
5	6.8941	-347.0%	0.6145	-28.9%	-1.8687	NA	0.4491
6	5.7434	-272.4%	0.2305	-73.3%	-0.9909	NA	0.3305
7	7.2043	-367.1%	0.3164	-63.4%	-2.1326	NA	0.4688
8	4.7914	-210.6%	0.6121	-29.2%	-0.3856	NA	0.5588
9	7.3197	-374.6%	0.3161	-63.4%	-2.2338	NA	0.4734
10	4.5043	-192.0%	0.4297	-50.3%	-0.2245	NA	0.5502
Average	**5.7684**	**-274.0%**	**0.4220**	**-51.2%**	**NA**	**NA**	**0.4728**

TABLE 10.34 *NAPLS Results for Dataset 7*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.6313**	**0.0%**	**0.8035**	**0.0%**	**0.7962742**	**0.0%**	
1	7.7605	-375.7%	0.4049	-49.6%	-3.6104	NA	0.3524
2	6.3565	-289.7%	0.3124	-61.1%	-2.0932	NA	0.3245
3	5.8230	-256.9%	0.3155	-60.7%	-1.5957	NA	0.3080
4	8.5764	-425.7%	0.0697	-91.3%	-4.6309	NA	0.3083
5	7.5125	-360.5%	0.3637	-54.7%	-3.3205	NA	0.3795
6	9.0281	-453.4%	0.2179	-72.9%	-5.2396	NA	0.3206
7	7.2616	-345.1%	0.2407	-70.0%	-3.0368	NA	0.3545
8	8.3335	-410.8%	0.2716	-66.2%	-4.3165	NA	0.2830
9	6.8662	-320.9%	0.0816	-89.8%	-2.6091	NA	0.3538
10	8.9279	-447.3%	0.1550	-80.7%	-5.1019	NA	0.3525
Average	**7.6446**	**-368.6%**	**0.2433**	**-69.7%**	**NA**	**NA**	**0.3337**

TABLE 10.35 *NAPLS Results for Dataset 8*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2960**	**0.0%**	**0.6652**	**0.0%**	**0.6600**	**0.0%**	**0**
1	0.7828	-164.5%	0.3419	-48.6%	-1.3775	NA	0.3160
2	1.0508	-255.0%	0.2576	-61.3%	-3.2835	NA	0.2924
3	1.1758	-297.2%	0.2151	-67.7%	-4.3636	NA	0.4574
4	1.2219	-312.8%	0.2388	-64.1%	-4.7923	NA	0.2830
5	0.9718	-228.3%	0.0380	-94.3%	-2.6641	NA	0.3905
6	1.1768	-297.5%	0.2210	-66.8%	-4.3724	NA	0.4017
7	1.1274	-280.9%	0.1350	-79.7%	-3.9313	NA	0.2976
8	1.1189	-278.0%	0.1164	-82.5%	-3.8574	NA	0.2879
9	1.0933	-269.3%	0.0614	-90.8%	-3.6372	NA	0.3193
10	1.1086	-274.5%	0.0993	-85.1%	-3.7677	NA	0.2940
Average*	**1.0828**	**-265.8%**	**0.1725**	**-74.1%**	**NA**	**NA**	**0.3340**

TABLE 10.36 *NAPLS Results for Dataset 9*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2061**	**0.0%**	**0.7850**	**0.0%**	**0.7848**	**0.0%**	**0**
1	0.5563	-169.9%	0.5446	-30.6%	-0.5675	NA	0.9073
2	0.5205	-152.5%	0.6476	-17.5%	-0.3725	NA	0.9533
3	0.3456	-67.7%	0.6118	-22.1%	0.3949	-49.7%	0.9481
4	0.3965	-92.4%	0.6613	-15.8%	0.2036	-74.1%	0.9657
5	0.3756	-82.2%	0.6938	-11.6%	0.2853	-63.6%	0.9448
6	0.3828	-85.7%	0.6415	-18.3%	0.2577	-67.2%	0.9139
7	0.3952	-91.7%	0.6593	-16.0%	0.2090	-73.4%	0.9457
8	0.3586	-74.0%	0.5979	-23.8%	0.3485	-55.6%	0.9500
9	0.5239	-154.2%	0.5935	-24.4%	-0.3904	NA	0.9469
10	0.4308	-109.0%	0.5797	-26.2%	0.0599	-92.4%	0.9548
Average*	**0.4286**	**-107.9%**	**0.6231**	**-20.6%**	**0.2513**	**-68.0%**	**0.9431**

10.5 NAPLS Results: 5 Outer Loops; 300 Inner Loops; 10% Noise

TABLE 10.37 *NAPLS Results for Dataset 1*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.6684**	**0.0%**	**0.9400**	**0.0%**	**0.8977**	**0.0%**	
1	0.4338	35.1%	0.9586	2.0%	0.9569	6.6%	0.9796
2	0.4671	30.1%	0.9586	2.0%	0.9500	5.8%	0.9868
3	0.6476	3.1%	0.9557	1.7%	0.9039	0.7%	0.9793
4	0.5069	24.2%	0.9552	1.6%	0.9411	4.8%	0.9506
5	0.4328	35.2%	0.9592	2.0%	0.9571	6.6%	0.9875
6	0.4491	32.8%	0.9540	1.5%	0.9538	6.3%	0.9648
7	0.4294	35.8%	0.9617	2.3%	0.9578	6.7%	0.9958
8	0.4686	29.9%	0.9523	1.3%	0.9497	5.8%	0.9655
9	0.4172	37.6%	0.9653	2.7%	0.9601	7.0%	0.9871
10	0.4338	35.1%	0.9593	2.0%	0.9569	6.6%	0.9739
Average	**0.4686**	**29.9%**	**0.9580**	**1.9%**	**0.9487**	**5.7%**	**0.9771**

TABLE 10.38 *NAPLS Results for Dataset 2*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2387**	**0.0%**	**0.4514**	**0.0%**	**0.4496**	**0.0%**	
1	0.2644	-10.8%	0.3616	-19.9%	0.3246	-27.8%	0.8474
2	0.2343	1.8%	0.4751	5.2%	0.4696	4.4%	0.8402
3	0.2490	-4.3%	0.4095	-9.3%	0.4010	-10.8%	0.8535
4	0.2330	2.4%	0.4781	5.9%	0.4753	5.7%	0.8477
5	0.2482	-4.0%	0.4184	-7.3%	0.4049	-10.0%	0.8351
6	0.2662	-11.5%	0.3341	-26.0%	0.3156	-29.8%	0.8438
7	0.2438	-2.2%	0.4281	-5.2%	0.4256	-5.3%	0.8090
8	0.2483	-4.0%	0.4080	-9.6%	0.4043	-10.1%	0.8197
9	0.2474	-3.7%	0.4147	-8.1%	0.4087	-9.1%	0.7991
10	0.2418	-1.3%	0.4408	-2.3%	0.4351	-3.2%	0.8674
Average	**0.2476**	**-3.8%**	**0.4168**	**-7.7%**	**0.4065**	**-9.6%**	**0.8363**

TABLE 10.39 *NAPLS Results for Dataset 3A*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.0739**	**0.0%**	**0.7580**	**0.0%**	**0.6867**	**0.0%**	**0**
1	0.2053	-177.8%	0.5909	-22.0%	-1.4176	NA	0.3833
2	0.1818	-146.0%	0.5551	-26.8%	-0.8960	NA	0.4987
3	0.3027	-309.5%	0.0000	-100.0%	-4.2547	NA	0.4108
4	0.1539	-108.2%	0.7548	-0.4%	-0.3583	NA	0.4589
5	0.2132	-188.5%	0.1269	-83.3%	-1.6074	NA	0.4365
6	0.3544	-379.5%	0.0185	-97.6%	-6.2037	NA	0.4425
7	0.2360	-219.3%	0.6907	-8.9%	-2.1951	NA	0.4615
8	0.1960	-165.1%	0.4693	-38.1%	-1.2023	NA	0.3407
9	0.1993	-169.7%	0.2595	-65.8%	-1.2791	NA	0.4341
10	0.1824	-146.7%	0.3982	-47.5%	-0.9071	NA	0.4747
Average	**0.2225**	**-201.0%**	**0.3864**	**-49.0%**	**NA**	**NA**	**0.4342**

TABLE 10.40 *NAPLS Results for Dataset 3B*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.7135**	**0.0%**	**0.9179**	**0.0%**	**0.9172**	**0.0%**	
1	1.0731	-50.4%	0.8469	-7.7%	0.8128	-11.4%	0.9006
2	1.0514	-47.4%	0.8986	-2.1%	0.8202	-10.6%	0.7866
3	2.1533	-201.8%	0.3244	-64.7%	0.2461	-73.2%	0.8438
4	1.5126	-112.0%	0.6447	-29.8%	0.6280	-31.5%	0.8023
5	1.3150	-84.3%	0.7565	-17.6%	0.7188	-21.6%	0.7975
6	1.3829	-93.8%	0.7236	-21.2%	0.6890	-24.9%	0.8107
7	1.2702	-78.0%	0.7626	-16.9%	0.7376	-19.6%	0.8655
8	1.5505	-117.3%	0.6510	-29.1%	0.6091	-33.6%	0.7805
9	0.9065	-27.1%	0.8717	-5.0%	0.8664	-5.5%	0.8833
10	1.0747	-50.6%	0.8585	-6.5%	0.8122	-11.5%	0.8915
Average	**1.3290**	**-86.3%**	**0.7338**	**-20.1%**	**0.6940**	**-24.3%**	**0.8362**

TABLE 10.41 *NAPLS Results for Dataset 3C*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.0804**	**0.0%**	**0.9192**	**0.0%**	**0.8423**	**0.0%**	
1	1.9470	-80.2%	0.5861	-36.2%	0.4879	-42.1%	0.5462
2	1.4102	-30.5%	0.7669	-16.6%	0.7314	-13.2%	0.6432
3	1.3154	-21.7%	0.7905	-14.0%	0.7663	-9.0%	0.7423
4	1.5777	-46.0%	0.7185	-21.8%	0.6637	-21.2%	0.7438
5	1.4579	-34.9%	0.8432	-8.3%	0.7129	-15.4%	0.6654
6	1.6356	-51.4%	0.8595	-6.5%	0.6386	-24.2%	0.6885
7	2.0829	-92.8%	0.6233	-32.2%	0.4139	-50.9%	0.8002
8	2.0862	-93.1%	0.5925	-35.5%	0.4121	-51.1%	0.6945
9	2.1773	-101.5%	0.4578	-50.2%	0.3596	-57.3%	0.6424
10	1.7385	-60.9%	0.7314	-20.4%	0.5917	-29.8%	0.7000
Average	**1.7429**	**-61.3%**	**0.6970**	**-24.2%**	**0.5778**	**-31.4%**	**0.6867**

TABLE 10.42 *NAPLS Results for Dataset 4*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5088**	**0.0%**	**0.6874**	**0.0%**	**0.6751839**	**0.0%**	**0**
1	1.7382	-15.2%	0.6356	-7.5%	0.5689	-15.7%	0.6141
2	2.0370	-35.0%	0.5567	-19.0%	0.4080	-39.6%	0.5486
3	1.9997	-32.5%	0.6598	-4.0%	0.4295	-36.4%	0.6703
4	2.3188	-53.7%	0.4620	-32.8%	0.2329	-65.5%	0.6038
5	3.2997	-118.7%	0.3607	-47.5%	-0.5535	NA	0.6716
6	3.3205	-120.1%	0.3110	-54.8%	-0.5731	NA	0.5374
7	2.5442	-68.6%	0.4303	-37.4%	0.0764	-88.7%	0.6437
8	2.4601	-63.0%	0.4152	-39.6%	0.1365	-79.8%	0.4730
9	2.2169	-46.9%	0.4497	-34.6%	0.2988	-55.7%	0.5499
10	2.4431	-61.9%	0.3530	-48.6%	0.1484	-78.0%	0.6225
Average*	**2.4378**	**-61.6%**	**0.4634**	**-32.6%**	**0.2874**	**NA**	**0.5935**

** Negative values of R^2_{pred} excluded from average*

TABLE 10.43 *NAPLS Results for Dataset 5*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.7958**	**0.0%**	**0.5048**	**0.0%**	**0.4698**	**0.0%**	
1	1.9869	-10.6%	0.4650	-7.9%	0.3509	-25.3%	0.8423
2	1.8075	-0.6%	0.5117	1.4%	0.4629	-1.5%	0.8574
3	1.8063	-0.6%	0.5026	-0.5%	0.4636	-1.3%	0.8761
4	1.8601	-3.6%	0.4919	-2.6%	0.4312	-8.2%	0.8894
5	1.9454	-8.3%	0.4392	-13.0%	0.3778	-19.6%	0.8407
6	1.8758	-4.5%	0.4674	-7.4%	0.4215	-10.3%	0.8899
7	1.8764	-4.5%	0.4799	-4.9%	0.4211	-10.4%	0.8911
8	1.8961	-5.6%	0.4887	-3.2%	0.4089	-13.0%	0.8519
9	1.8620	-3.7%	0.5013	-0.7%	0.4300	-8.5%	0.8650
10	1.8583	-3.5%	0.4813	-4.7%	0.4323	-8.0%	0.9150
Average	**1.8775**	**-4.5%**	**0.4829**	**-4.3%**	**0.4200**	**-10.6%**	**0.8719**

TABLE 10.44 *NAPLS Results for Dataset 6*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5425**	**0.0%**	**0.8646**	**0.0%**	**0.8564**	**0.0%**	
1	5.3794	-248.8%	0.3699	-57.2%	-0.7466	NA	0.5442
2	5.4359	-252.4%	0.3669	-57.6%	-0.7835	NA	0.5048
3	7.0898	-359.6%	0.3119	-63.9%	-2.0339	NA	0.4233
4	4.8513	-214.5%	0.3616	-58.2%	-0.4205	NA	0.5876
5	5.2746	-242.0%	0.6197	-28.3%	-0.6792	NA	0.5797
6	6.6312	-329.9%	0.0782	-91.0%	-1.6540	NA	0.3799
7	4.0866	-164.9%	0.5181	-40.1%	-0.0080	NA	0.5853
8	5.8793	-281.2%	0.3365	-61.1%	-1.0863	NA	0.4893
9	5.4749	-255.0%	0.3919	-54.7%	-0.8092	NA	0.4737
10	5.5869	-262.2%	0.1565	-81.9%	-0.8839	NA	0.4421
Average	**5.5690**	**-261.0%**	**0.3511**	**-59.4%**	**NA**	**NA**	**0.5010**

TABLE 10.45 *NAPLS Results for Dataset 7*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.6313**	**0.0%**	**0.8035**	**0.0%**	**0.7962742**	**0.0%**	
1	6.2965	-286.0%	0.2091	-74.0%	-2.0351	NA	0.3730
2	5.2925	-224.4%	0.4276	-46.8%	-1.1443	NA	0.3275
3	7.4302	-355.5%	0.0677	-91.6%	-3.2263	NA	0.4154
4	10.9430	-570.8%	0.1340	-83.3%	-8.1672	NA	0.3142
5	4.1176	-152.4%	0.3689	-54.1%	-0.2979	NA	0.4367
6	7.3702	-351.8%	0.3151	-60.8%	-3.1584	NA	0.3493
7	6.7528	-313.9%	0.4177	-48.0%	-2.4909	NA	0.3091
8	9.4616	-480.0%	0.0922	-88.5%	-5.8533	NA	0.4582
9	9.3608	-473.8%	0.0670	-91.7%	-5.7081	NA	0.5346
10	5.7830	-254.5%	0.2266	-71.8%	-1.5602	NA	0.3142
Average	**7.2808**	**-346.3%**	**0.2326**	**-71.1%**	**NA**	**NA**	**0.3832**

TABLE 10.46 *NAPLS Results for Dataset 8*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2960**	**0.0%**	**0.6652**	**0.0%**	**0.6600**	**0.0%**	
1	1.0451	-253.1%	0.0586	-91.2%	-3.2377	NA	0.3679
2	1.0187	-244.1%	0.2791	-58.0%	-3.0263	NA	0.4737
3	1.2233	-313.2%	0.1285	-80.7%	-4.8056	NA	0.4127
4	0.9040	-205.4%	0.0888	-86.7%	-2.1703	NA	0.4080
5	1.4302	-383.2%	0.1321	-80.1%	-6.9358	NA	0.4186
6	1.0759	-263.5%	0.1277	-80.8%	-3.4911	NA	0.4032
7	0.8648	-192.2%	0.1629	-75.5%	-1.9016	NA	0.4046
8	1.4185	-379.2%	0.1943	-70.8%	-6.8062	NA	0.3980
9	1.1193	-278.1%	0.0340	-94.9%	-3.8601	NA	0.3993
10	1.3665	-361.6%	0.0093	-98.6%	-6.2441	NA	0.4512
Average	**1.1466**	**-287.4%**	**0.1215**	**-81.7%**	**NA**	**NA**	**0.4137**

TABLE 10.47 *NAPLS Results for Dataset 9*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2061**	**0.0%**	**0.7850**	**0.0%**	**0.7847772**	**0.0%**	
1	0.2956	-43.4%	0.6915	-11.9%	0.5574	-29.0%	0.9235
2	0.3053	-48.1%	0.7537	-4.0%	0.5277	-32.8%	0.9079
3	0.3510	-70.3%	0.6214	-20.8%	0.3758	-52.1%	0.9562
4	0.2917	-41.5%	0.6844	-12.8%	0.5691	-27.5%	0.9489
5	0.3945	-91.4%	0.6683	-14.9%	0.2119	-73.0%	0.9265
6	0.3006	-45.8%	0.6889	-12.2%	0.5424	-30.9%	0.9387
7	0.2830	-37.3%	0.6701	-14.6%	0.5944	-24.3%	0.9301
8	0.2623	-27.2%	0.7012	-10.7%	0.6515	-17.0%	0.9329
9	0.3154	-53.0%	0.6493	-17.3%	0.4962	-36.8%	0.9253
10	0.3993	-93.7%	0.7205	-8.2%	0.1923	-75.5%	0.9307
Average	**0.3199**	**-55.2%**	**0.6849**	**-12.7%**	**0.4719**	**-39.9%**	**0.9321**

10.6 NAPLS Results: 5 Outer Loops; 500 Inner Loops; 10% Noise

TABLE 10.48 *NAPLS Results for Dataset 1*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.6684**	**0.0%**	**0.9400**	**0.0%**	**0.8977**	**0.0%**	
1	0.5283	21.0%	0.9496	1.0%	0.9361	4.3%	0.9831
2	0.4210	37.0%	0.9611	2.2%	0.9594	6.9%	0.9900
3	0.4407	34.1%	0.9594	2.1%	0.9555	6.4%	0.9900
4	0.4305	35.6%	0.9611	2.2%	0.9575	6.7%	0.9875
5	0.4236	36.6%	0.9617	2.3%	0.9589	6.8%	0.9913
6	0.4947	26.0%	0.9543	1.5%	0.9439	5.2%	0.9621
7	0.4369	34.6%	0.9584	2.0%	0.9563	6.5%	0.9898
8	0.4152	37.9%	0.9664	2.8%	0.9605	7.0%	0.9788
9	0.5125	23.3%	0.9544	1.5%	0.9398	4.7%	0.9812
10	0.4269	36.1%	0.9595	2.1%	0.9583	6.7%	0.9792
Average	**0.4530**	**32.2%**	**0.9586**	**2.0%**	**0.9526**	**6.1%**	**0.9833**

TABLE 10.49 *NAPLS Results for Dataset 2*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2387**	**0.0%**	**0.4514**	**0.0%**	**0.4496**	**0.0%**	
1	0.2335	2.2%	0.4900	8.6%	0.4734	5.3%	0.7943
2	0.2445	-2.5%	0.4269	-5.4%	0.4223	-6.1%	0.8306
3	0.2335	2.2%	0.4900	8.6%	0.4734	5.3%	0.7943
4	0.2445	-2.5%	0.4269	-5.4%	0.4223	-6.1%	0.8306
5	0.2917	-22.2%	0.3145	-30.3%	0.1780	-60.4%	0.8428
6	0.2360	1.1%	0.4830	7.0%	0.4619	2.7%	0.8549
7	0.2402	-0.6%	0.4513	0.0%	0.4425	-1.6%	0.8232
8	0.2390	-0.1%	0.4592	1.7%	0.4481	-0.3%	0.8189
9	0.2392	-0.2%	0.4510	-0.1%	0.4472	-0.5%	0.8224
10	0.2653	-11.1%	0.3728	-17.4%	0.3201	-28.8%	0.8206
Average	**0.2467**	**-3.4%**	**0.4365**	**-3.3%**	**0.4089**	**-9.1%**	**0.8233**

TABLE 10.50 *NAPLS Results for Dataset 3A*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.0739**	**0.0%**	**0.7580**	**0.0%**	**0.6867**	**0.0%**	**0**
1	0.2099	-183.9%	0.3430	-54.8%	-1.5258	NA	0.6184
2	0.1945	-163.1%	0.2803	-63.0%	-1.1690	NA	0.4025
3	0.2116	-186.3%	0.3203	-57.7%	-1.5682	NA	0.4730
4	0.1704	-130.6%	0.7292	-3.8%	-0.6658	NA	0.4383
5	0.1769	-139.3%	0.4170	-45.0%	-0.7943	NA	0.4320
6	0.1846	-149.7%	0.1557	-79.5%	-0.9537	NA	0.5915
7	0.1910	-158.5%	0.6244	-17.6%	-1.0931	NA	0.3915
8	0.1264	-70.9%	0.7358	-2.9%	0.0844	-87.7%	0.4861
9	0.1910	-158.4%	0.2462	-67.5%	-1.0923	NA	0.4901
10	0.2637	-256.8%	0.5543	-26.9%	-2.9888	NA	0.4951
Average	**0.1920**	**-159.8%**	**0.4406**	**-41.9%**	**NA**	**NA**	**0.4818**

TABLE 10.51 *NAPLS Results for Dataset 3B*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.7135**	**0.0%**	**0.9179**	**0.0%**	**0.9172**	**0.0%**	
1	1.3356	-87.2%	0.7673	-16.4%	0.7099	-22.6%	0.8100
2	0.9581	-34.3%	0.8638	-5.9%	0.8507	-7.2%	0.8227
3	1.0117	-41.8%	0.8684	-5.4%	0.8336	-9.1%	0.7774
4	0.9312	-30.5%	0.8798	-4.2%	0.8590	-6.3%	0.8138
5	1.0644	-49.2%	0.8531	-7.1%	0.8158	-11.1%	0.8160
6	1.4346	-101.1%	0.6694	-27.1%	0.6653	-27.5%	0.8047
7	1.5622	-119.0%	0.6127	-33.3%	0.6032	-34.2%	0.8263
8	1.1024	-54.5%	0.8051	-12.3%	0.8024	-12.5%	0.7709
9	1.0203	-43.0%	0.8331	-9.2%	0.8307	-9.4%	0.8589
10	0.6611	7.3%	0.9290	1.2%	0.9289	1.3%	0.8329
Average	**1.1082**	**-55.3%**	**0.8082**	**-12.0%**	**0.7900**	**-13.9%**	**0.8134**

TABLE 10.52 *NAPLS Results for Dataset 3C*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.0804**	**0.0%**	**0.9192**	**0.0%**	**0.8423**	**0.0%**	
1	1.5766	-45.9%	0.7417	-19.3%	0.6642	-21.1%	0.7960
2	2.0443	-89.2%	0.5416	-41.1%	0.4355	-48.3%	0.6879
3	1.1369	-5.2%	0.8260	-10.1%	0.8254	-2.0%	0.8204
4	2.4732	-128.9%	0.3336	-63.7%	0.1737	-79.4%	0.6989
5	1.3813	-27.8%	0.7482	-18.6%	0.7423	-11.9%	0.6777
6	0.9784	9.4%	0.9092	-1.1%	0.8707	3.4%	0.6731
7	1.7785	-64.6%	0.6495	-29.3%	0.5727	-32.0%	0.6838
8	1.2680	-17.4%	0.8406	-8.5%	0.7828	-7.1%	0.5930
9	1.7930	-65.9%	0.6975	-24.1%	0.5658	-32.8%	0.6676
10	2.0740	-92.0%	0.5946	-35.3%	0.4189	-50.3%	0.6034
Average	**1.6504**	**-52.8%**	**0.6883**	**-25.1%**	**0.6052**	**-28.1%**	**0.6902**

TABLE 10.53 *NAPLS Results for Dataset 4*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5088**	**0.0%**	**0.6874**	**0.0%**	**0.6751839**	**0.0%**	**0**
1	2.0671	-37.0%	0.5946	-13.5%	0.3904	-42.2%	0.5912
2	2.7128	-79.8%	0.4556	-33.7%	-0.0500	NA	0.5077
3	2.6335	-74.5%	0.3913	-43.1%	0.0105	-98.4%	0.6021
4	2.5262	-67.4%	0.4251	-38.2%	0.0895	-86.7%	0.5895
5	2.2880	-51.6%	0.4723	-31.3%	0.2531	-62.5%	0.5191
6	2.2343	-48.1%	0.5677	-17.4%	0.2878	-57.4%	0.6056
7	2.8429	-88.4%	0.4216	-38.7%	-0.1531	NA	0.6224
8	2.8311	-87.6%	0.4275	-37.8%	-0.1436	NA	0.5491
9	2.4303	-61.1%	0.5445	-20.8%	0.1573	-76.7%	0.6569
10	2.9824	-97.7%	0.3600	-47.6%	-0.2690	NA	0.5964
Average*	**2.5549**	**-69.3%**	**0.4660**	**-32.2%**	**0.1981**	**-70.7%**	**0.5840**

** Negative values of R^2_{pred} excluded from average*

TABLE 10.54 *NAPLS Results for Dataset 5*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.7958**	**0.0%**	**0.5048**	**0.0%**	**0.4698**	**0.0%**	
1	2.0081	-11.8%	0.4768	-5.6%	0.3371	-28.3%	0.8384
2	1.8317	-2.0%	0.4903	-2.9%	0.4484	-4.6%	0.8284
3	1.9298	-7.5%	0.4552	-9.8%	0.3878	-17.5%	0.8350
4	1.9723	-9.8%	0.4736	-6.2%	0.3605	-23.3%	0.8472
5	1.8294	-1.9%	0.5017	-0.6%	0.4498	-4.3%	0.8372
6	1.9466	-8.4%	0.4994	-1.1%	0.3770	-19.8%	0.8317
7	1.8215	-1.4%	0.5001	-0.9%	0.4545	-3.3%	0.8644
8	1.8495	-3.0%	0.4993	-1.1%	0.4376	-6.9%	0.8215
9	1.8264	-1.7%	0.5257	4.1%	0.4516	-3.9%	0.8486
10	1.9708	-9.7%	0.4415	-12.5%	0.3614	-23.1%	0.8412
Average	**1.8986**	**-5.7%**	**0.4864**	**-3.7%**	**0.4066**	**-13.5%**	**0.8394**

TABLE 10.55 *NAPLS Results for Dataset 6*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.5425**	**0.0%**	**0.8646**	**0.0%**	**0.8564036**	**0.0%**	**0**
1	4.7364	-207.1%	0.2650	-69.4%	-0.3540	NA	0.4545
2	5.9396	-285.1%	0.1889	-78.2%	-1.1293	NA	0.3957
3	5.7206	-270.9%	0.2246	-74.0%	-0.9752	NA	0.4140
4	5.0099	-224.8%	0.3358	-61.2%	-0.5149	NA	0.3926
5	4.6981	-204.6%	0.4246	-50.9%	-0.3322	NA	0.5555
6	5.6314	-265.1%	0.3814	-55.9%	-0.9140	NA	0.4970
7	6.4051	-315.3%	0.2106	-75.6%	-1.4761	NA	0.5059
8	4.8344	-213.4%	0.5214	-39.7%	-0.4106	NA	0.5433
9	5.4342	-252.3%	0.4038	-53.3%	-0.7823	NA	0.5226
10	5.7966	-275.8%	0.4773	-44.8%	-1.0280	NA	0.4324
Average*	**5.4206**	**-251.4%**	**0.3433**	**-60.3%**	**NA**	**NA**	**0.4714**

TABLE 10.56 *NAPLS Results for Dataset 7*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**1.6313**	**0.0%**	**0.8035**	**0.0%**	**0.7962742**	**0.0%**	**0**
1	11.9231	-630.9%	0.0478	-94.1%	-9.8830	NA	0.6106
2	6.6087	-305.1%	0.4645	-42.2%	-2.3435	NA	0.3471
3	5.5012	-237.2%	0.1975	-75.4%	-1.3167	NA	0.3063
4	6.5638	-302.4%	0.4294	-46.6%	-2.2982	NA	0.4352
5	7.8893	-383.6%	0.0468	-94.2%	-3.7648	NA	0.4260
6	7.8556	-381.5%	0.2883	-64.1%	-3.7241	NA	0.4726
7	7.5024	-359.9%	0.2276	-71.7%	-3.3089	NA	0.3509
8	6.9255	-324.5%	0.3060	-61.9%	-2.6717	NA	0.3769
9	8.9376	-447.9%	0.2318	-71.2%	-5.1151	NA	0.4238
10	10.7111	-556.6%	0.1104	-86.3%	-7.7828	NA	0.4366
Average*	**8.0418**	**-393.0%**	**0.2350**	**-70.8%**	**NA**	**NA**	**0.4186**

TABLE 10.57 *NAPLS Results for Dataset 8*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2960**	**0.0%**	**0.6652**	**0.0%**	**0.6600**	**0.0%**	**0**
1	0.9441	-218.9%	0.4707	-29.2%	-2.4580	NA	0.3699
2	1.2642	-327.1%	0.0505	-92.4%	-5.2003	NA	0.4512
3	0.8502	-187.2%	0.2962	-55.5%	-1.8046	NA	0.4158
4	1.0178	-243.8%	0.2353	-64.6%	-3.0185	NA	0.3209
5	1.2147	-310.3%	0.3266	-50.9%	-4.7241	NA	0.5542
6	0.9441	-218.9%	0.4707	-29.2%	-2.4580	NA	0.3699
7	1.2642	-327.1%	0.0505	-92.4%	-5.2003	NA	0.4512
8	0.8502	-187.2%	0.2962	-55.5%	-1.8046	NA	0.4158
9	1.0178	-243.8%	0.2353	-64.6%	-3.0185	NA	0.3209
10	1.2147	-310.3%	0.3266	-50.9%	-4.7241	NA	0.5542
Average*	**1.0582**	**-257.5%**	**0.2759**	**-58.5%**	**NA**	**NA**	**0.4224**

TABLE 10.58 *NAPLS Results for Dataset 9*

Run	RMSE		R^2		R^2_{pred}		R^2_{median}
		% Δ		% Δ		% Δ	
PLS	**0.2061**	**0.0%**	**0.7850**	**0.0%**	**0.7848**	**0.0%**	
1	0.3610	-75.2%	0.6980	-11.1%	0.3397	-56.7%	0.9313
2	0.3712	-80.1%	0.6675	-15.0%	0.3019	-61.5%	0.9295
3	0.4142	-101.0%	0.6493	-17.3%	0.1309	-83.3%	0.9278
4	0.2654	-28.8%	0.6818	-13.1%	0.6431	-18.1%	0.9075
5	0.2693	-30.6%	0.7225	-8.0%	0.6327	-19.4%	0.9280
6	0.3610	-75.2%	0.6980	-11.1%	0.3397	-56.7%	0.9313
7	0.3712	-80.1%	0.6675	-15.0%	0.3019	-61.5%	0.9295
8	0.4142	-101.0%	0.6493	-17.3%	0.1309	-83.3%	0.9278
9	0.2654	-28.8%	0.6818	-13.1%	0.6431	-18.1%	0.9075
10	0.2693	-30.6%	0.7225	-8.0%	0.6327	-19.4%	0.9280
Average	**0.3362**	**-63.1%**	**0.6838**	**-12.9%**	**0.4097**	**-47.8%**	**0.9248**

11 Addendum C: Code examples

R FUNCTION 11.1 *f.NoisePLS*

```
f.NoisePLS <- function(input.object, error.trace=T, outer.reps=5, inner.reps=100,
noise.level=0.1, use.Scale=F) {
# This function implements the NAPLS algorithm
# The input object is a list that specifies all the parameters needed for modelling


  # Calculate Root Mean Square Error of Prediction
  f.RMSEP <- function(object, newdata, pos=input.object$pos.dep.var, nComps=1) {
    y <- newdata[,pos]
    yhat <- predict.mvr(object, newdata)[,,nComps]
    sqrt(mean((y-yhat)^2))
  }

  # Calculate R2 of Actual and Predicted Values
  f.R2 <- function(object, newdata, pos=input.object$pos.dep.var, nComps=1) {
    y <- newdata[,pos]
    yhat <- predict.mvr(object, newdata)[,,nComps]
    #(sum((yhat-mean(yhat))^2)) / (sum((y-mean(y))^2))
    (cor(y,yhat))^2

  }
  # Calculate the coefficient of determination for predicted values
  f.R2.Pred <- function (object, newdata, pos=input.object$pos.dep.var, nComps=1) {
    y <- newdata[,pos]
    yhat <- predict.mvr(object, newdata=newdata)
    R2.Pred <- 1 - sum((yhat[,,nComps]-y)^2) / sum((y - mean(y))^2)
    R2.Pred
  }

  # Read Objects Parameters
  model <- input.object$model

  # Number of independent variables; Add 1 for intercept
  p <- input.object$p + 1

  # Get name of dependent variable as well as position in input data frame
  pos.dep.var <- input.object$pos.dep.var

  # Number of components
  nComps = input.object$nComps

  # Size of calibration dataset
  n.Cal <- input.object$n.train

  # Create Coefficient Matrix
  beta.coeff <- matrix(nrow=outer.reps*inner.reps, ncol=p)

  # Step 1 - Split Data into Cal / Val / Tes
  Data.Cal      <- input.object$train.data
  Data.Val      <- input.object$val.data
  Data.Tst      <- input.object$test.data
  Y.Mean        <- mean(rbind(Data.Cal, Data.Val)[,pos.dep.var])

  # Step 2 - PLS model using calibration set
  PLSOut <- plsr(model, data = Data.Cal, method="simpls", ncomp=nComps, scale=use.Scale)
  if (error.trace == T) {cat("Step 2 done\n")}

  # Step 3 - Calinration model's fit statistics
  RMSE.Cal.Start <- f.RMSEP(object=PLSOut, newdata=Data.Cal, nComps=nComps)
  RMSE.Val.Start <- f.RMSEP(object=PLSOut, newdata=Data.Val, nComps=nComps )
  RMSE.Tst.Start <- f.RMSEP(object=PLSOut, newdata=Data.Tst, nComps=nComps)
```

R FUNCTION 11.1 *f.NoisePLS (Continued)*

```
  R2.Cal.Start <- f.R2(object=PLSOut, newdata=Data.Cal, nComps=nComps)
  R2.Val.Start <- f.R2(object=PLSOut, newdata=Data.Val, nComps=nComps)
  R2.Tst.Start <- f.R2(object=PLSOut, newdata=Data.Tst, nComps=nComps)

  R2.Pred.Cal.Start <- f.R2.Pred(object=PLSOut, newdata=Data.Cal, nComps=nComps)
  R2.Pred.Val.Start <- f.R2.Pred(object=PLSOut, newdata=Data.Val, nComps=nComps)
  R2.Pred.Tst.Start <- f.R2.Pred(object=PLSOut, newdata=Data.Tst, nComps=nComps)

  if (error.trace == T) {cat("Step 3 done\n")}

  RMSE.Begin <- RMSE.Val.Start
  PLSOut.Begin <- PLSOut

  for (outer_loop in 1:outer.reps) { # Step 10
    if (error.trace == T) {cat("-----------------------------------------------------\nStep
10 loop",outer_loop,"\n")}

    repeats <- 0
    RMSE.Begin <- 100
    PLSOut.Begin <- PLSOut
    Data.Cal <- input.object$train.data

    for (count in 1:inner.reps) {
      if (error.trace == T) {cat("  Step 9 loop\n")}

      # Step 7
      Data.Cal.New <- Data.Cal
      Data.Cal.New[,pos.dep.var] <- Data.Cal.New[,pos.dep.var] + (rnorm(n.Cal) * Y.Mean *
noise.level * 0.90^repeats)

      if (error.trace == T) {cat("    Step 7 done\n")}

      # Step 8
      PLSOut.New <- plsr(model, data = Data.Cal.New, method="simpls", ncomp=nComps,
scale=use.Scale)
      RMSE.New <- f.RMSEP(object=PLSOut.New, newdata=Data.Val, nComps=nComps)

      beta.coeff[(outer_loop-1)*inner.reps + count,] <- coef.mvr(PLSOut.New, intercept=T)

      if (error.trace == T) {cat("    Step 8: RMSE.Begin =",RMSE.Begin, "RMSE.New
=",RMSE.New,"\n")}

      # If noise addition improved the model's fit:
      # Reduce noise factor and save "noisy" calibration data
      if (RMSE.New < RMSE.Begin) {
        repeats <- repeats + 1
        RMSE.Begin <- RMSE.New
        PLSOut.Begin <- PLSOut.New
        Data.Cal <- Data.Cal.New
      }
    }
  }

  # Determine the optimal NAPLS model
  optimal.stats <- f.NoisePLS.Model(input.object, beta.coeff)

  # Output all the fit statistics
  list("R2.Pred.Start"  = R2.Pred.Tst.Start,
       "RMSE.Start"     = RMSE.Tst.Start,
       "R2.Start"       = R2.Tst.Start,

       "R2.Pred.Optimal" = optimal.stats$R2.Pred.Optimal,
       "RMSE.Optimal"    = optimal.stats$RMSE.Optimal,
       "R2.Optimal"      = optimal.stats$R2.Optimal,
       "Best.Model.Cor"  = optimal.stats$Best.Model.Cor,
       "yhat.opt"        = optimal.stats$yhat.opt)
}
```

R FUNCTION 11.2 *f.NoisePLS.Model*

```
f.NoisePLS.Model <- function(input.object, Models) {
# This function is called by f.NoisePLS to calculate the optimal NAPLS Model

  # Get Predicted Values
  f.yhat <- function(use.model, newdata, p) {
    yhat <- rep(NA,nrow(newdata))
    for (i in 1:nrow(newdata)) {
      this.data <- c(1, as.matrix((newdata[i,1:p])))
      yhat[i] <- sum(this.data * use.model)
    }
    yhat
  }

  # Get Y Values
  f.y <- function(newdata, p) {
    y <- newdata[,p+1]
  }

  # Calculate Root Mean Square Error of Prediction
  f.RMSEP <- function(y, yhat) {
    sqrt(mean((y-yhat)^2))
  }

  # Calculate R2 of Actual and Predicted Values
  f.R2 <- function(y, yhat) {
    (cor(y,yhat))^2
  }

  # Calculate the coefficient of determination for predicted values
  f.R2.Pred <- function (y, yhat) {
    1 - sum((yhat-y)^2) / sum((y - mean(y))^2)
  }

  # Get Model with highest R2 (correlation) with Median Model
  f.GetMedianModel <- function(Models) {
    beta.coeff.median <- apply(Models, 2, median)
    n.row <-  nrow(Models)
    distances <- rep(0,n.row)
    for (i in 1:n.row) {
      distances[i] <- (cor(beta.coeff.median, Models[i,]))^2
    }
    median.model.no <- c(1:nrow(Models))[distances==max(distances)]
    median.model.no <- min(median.model.no)
    list("optimum.model" = Models[median.model.no,], "Model.Cor" = max(distances))
  }

  Median.Model.Tmp <- f.GetMedianModel(Models)
  Median.Model <- Median.Model.Tmp$optimum.model
  Median.Model.R2 <- Median.Model.Tmp$Model.Cor

  # Get observed and predicted values
  y     <- f.y(newdata=input.object$test.data, input.object$p)
  yhat  <- f.yhat(use.model=Median.Model, newdata=input.object$test.data, input.object$p)

  # Calculate fit statistics for optimal model
  rmse.optimal <- f.RMSEP(y, yhat)
  r2.optimal   <- f.R2(y, yhat)
  R2.Pred.optimal <- f.R2.Pred(y, yhat)

  # Output Results
  list( "Best.Model.Cor"  = Median.Model.R2,
        "RMSE.Optimal"    = rmse.optimal,
        "R2.Optimal"      = r2.optimal,
        "R2.Pred.Optimal" = R2.Pred.optimal,
        "yhat.opt"        = yhat)
}
```

R FUNCTION 11.3 *f.PLS.TT.Plot*

```
f.PLS.TT.Plot <- function(object) {
# This function plots the T scores for Component 1 against Component 2

  # Get PLS Model
  use.data <- rbind(object$train.data, object$val.data)
  model <- object$model
  temp <- plsr(data=use.data, formula=model, method="simpls", scale=FALSE)

  # Get T scores for components 1 and 2
  TScores1 <- scores(temp)[,1]
  TScores2 <- scores(temp)[,2]

  # Calculate max of scores - needed so that the zero point is in the center of the graph
  maxT1 <- max(abs(TScores1))
  maxT2 <- max(abs(TScores2))

  use.dimnames <- dimnames(scores(temp))[[1]]
  par.old <- par()
  par(pty='s')
  plot(x=TScores1, y=TScores2, pch=20, main="T-T Score Plot", xlab="Component 1",
ylab="Component 2", xaxt='n', yaxt='n', xlim=c(-maxT1, maxT1), ylim=c(-maxT2, maxT2))
  par(lty='dashed')
  abline(h=0, col='gray')
  abline(v=0, col='gray')
  par(lty='solid')
  text(x=TScores1, y=TScores2, labels=use.dimnames, pos=1, offset=0.1, cex=1.25)
}
```

R FUNCTION 11.4 *f.PLS.TvsU.Plot*

```
f.PLS.TvsU.Plot <- function(object, comp=1, use.scale=FALSE) {
# This Function plots the T and U scores for the given component

  # Draw regression line
  f.reg.line <- function (mod, col = palette()[2], lwd = 2, lty = 1, ...) {
    if (!is.null(class(mod$na.action)) && class(mod$na.action) ==
        "exclude")
        class(mod$na.action) <- "omit"
    coef <- coefficients(mod)
    if (length(coef) != 2)
        stop(" Requires simple linear regression.")
    x <- model.matrix(mod)[, 2]
    y <- fitted.values(mod)
    min <- which.min(x)
    max <- which.max(x)
    lines(c(x[min], x[max]), c(y[min], y[max]), col = col, lty = lty,
        lwd = lwd, ...)
  }

  use.data <- rbind(object$train.data, object$val.data)
  model <- object$model
  temp <- plsr(data=use.data, formula=model, method="simpls", scale=use.scale)
  TScores <- scores(temp)[,comp]
  UScores <- Yscores(temp)[,comp]
  use.dimnames <- dimnames(scores(temp))[[1]]

  plot(x=TScores, y=UScores, pch=20, main=paste("X-Y Relation Outliers: Component ", comp,
sep=""), xlab="T Scores", ylab="U Scores")
  text(x=TScores, y=UScores, labels=use.dimnames, pos=1, offset=0.1)

  abline(h=0)
  abline(v=0)

  ls.line <- lm(formula = UScores ~ TScores)
  f.reg.line(ls.line)
}
```

R FUNCTION 11.5 *f.PLS.Crossval*

```
f.PLS.Crossval <- function (object, K=10, maxComps=10, Scale=F) {
# Calculate K-fold cross-validation
# Plots the cross-validation error and proportion of variance explained by each component

  par(pty='s')
  use.data <- rbind(object$train.data, object$val.data)

  ncomp <- min(floor(nrow(use.data) * (K-1)/K), ncol(use.data)-1, maxComps)
  n <- nrow(use.data)

  CV.matrix <- matrix(nrow = K, ncol = ncomp)
  CV2.matrix <- matrix(nrow = K, ncol = ncomp)

  random.order <- sample(1:n)
  block.size <- n / K
  use.data <- use.data[random.order,]
  model <- object$model

  for (i in 1:K) {
    lvout.start <- (i-1)* block.size + 1
    lvout.end <- (i)* block.size
    train.data <- use.data[-(lvout.start:lvout.end),]
    test.data <- use.data[(lvout.start:lvout.end),]
    PLSOut <- plsr(model, data = train.data, method="simpls", ncomp=ncomp, scale=Scale)
    temp.R2 <- f.R2(in.object=PLSOut, new.data=test.data)
    temp.RMSEP <- f.RMSEP(in.object=PLSOut, new.data=test.data)

    for (j in 1:ncomp) {
      CV.matrix[i,j] <- temp.RMSEP[j]
      CV2.matrix[i,j] <- temp.R2[j]
    }
  }

  par(mfrow=c(1,2))

  # Plot cross-validation error
  CV.Error<-matrix(apply(CV.matrix, 2, mean), ncol=1)
  plot(CV.Error, type='l', xlab="Number of components", ylab="Cross-Validation Error",
main=paste(K, "-fold Validation", sep=""))
  points(CV.Error, pch=19)

  # Scree plot
  PLSOut <- plsr(model, data = use.data, method="simpls", ncomp=ncomp)
  Expl.Var <- explvar(PLSOut)
  plot(Expl.Var, type = "l", xlab="Component", ylab="Variance Explained by Component",
main=paste("Scree Plot: Variance explained by first",ncomp, "components") )
  points(Expl.Var, pch=19)

  par(mfrow=c(1,1))
}
```

R FUNCTION 11.6 *Sample code to create an object for NAPLS modeling*

```
# Sample code to create an object for NAPLS modeling
use.dataset <- list("model"=Y~., "samples"=sample(102),
                          "n" = 102, "p" = NULL,
                          "n.train" = 63, "n.validate" = 27, "n.test" = 12, nComps=9,
                          "pos.dep.var" = 1558,
                          "full.data" = input.data,
                          "train.data" = NULL,
                          "val.data" = NULL,
                          "test.data" = NULL)
# Add own code to allocate train.data, val.data and test.data
```

R FUNCTION 11.7 *f.PLS.Leverage*

```
f.PLS.Leverage <- function(input.object, use.scale=F) {
# Calculates the leverage for each of the observations
# Returns leverage for each point in descending order

  Data.Train  <- rbind(input.object$train.data, input.object$val.data)
  model <- input.object$model
  nComps <- input.object$nComps
  n <- nrow(Data.Train)

  PLSOut      <- plsr(model, data = Data.Train, method="simpls", ncomp=nComps,
scale=use.scale)
  object.names <- dimnames(Data.Train)[[1]]
  tt <- matrix(nrow=1, ncol=nComps)
  leverage <- matrix(ncol=1, nrow=n, dimnames=list(object.names, "Leverage"))

  for (i in 1:nComps) {
    t.i <- matrix(scores(PLSOut)[,i], ncol=1)
    tt[i] <- t(t.i) %*% t.i
  }

  for (j in 1:n) { # leverage for object j {
    leverage[j] <- 1/n
    for (i in 1:nComps) {
      leverage[j] <- leverage[j] + ((scores(PLSOut)[j,i])^2 / tt[i])
    }
  }
  sort.order <- sort(leverage, decreasing=T, index.return = T)$ix
  leverage[sort.order,, drop=F]
}
```

R FUNCTION 11.8 *f.PLS.Remove.Leverage*

```
f.PLS.Remove.Leverage <- function(input.object, tries=5, use.scale=F) {
# This function will iteratively fit a PLS model and remove
# the observation with the largest leverage
# The number of points to remove is specified by tries
# Detailed output of leverage is given at each iteration

  this.data <- input.object
  this.data$nComps <- input.object$nComps
  removed.objects <- matrix(nrow=1, ncol=tries)
  output.leverage <- matrix(ncol=tries, nrow=10)
  output.leverage.obj <- matrix(ncol=tries, nrow=10)

  for (i in 1:tries) {
    leverage.i              <- f.PLS.Leverage(this.data, use.scale=use.scale)
    biggest.leverage.i      <- dimnames(leverage.i[1,,drop=F])[[1]]
    removed.objects[i]      <- biggest.leverage.i
    output.leverage[1:10,i] <- leverage.i[1:10]
    output.leverage.obj[1:10,i] <- dimnames(leverage.i)[[1]][1:10]

    this.data$train.data    <-
this.data$train.data[!dimnames(this.data$train.data)[[1]]==biggest.leverage.i,]
    this.data$val.data      <-
this.data$val.data[!dimnames(this.data$val.data)[[1]]==biggest.leverage.i,]

  }
  list(removed.objects, output.leverage.obj, output.leverage)
}
```

12 References

Dardenne, P. and Fernández Pierna, J.A., (2006), “A new method to improve the accuracy of the near infrared models: noise addition partial least squares method”, *Journal of Near Infrared Spectroscopy,* 14, 349-355.

Esbensen, K.H., (2001), *An Introduction to Multivariate Data Analysis and Experimental Design*, CAMO

Hastie, T., Tibshirani, R. and Friedman, J., (2001), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer

Helland, I., (1988), “On the structure of partial least squares regression”, *Communications in Statistics, Simulation and Computation*, 17(2), 581-607.

Næs, T., Isaksson, T., Fearn T. and Davies, T., (2002) *A user-friendly guide to Multivariate Calibration and Classification,* NIR Publications.

R, (2007), Statistical Programming Software, version 2.6.1, The R Foundation for Statistical Computing (http://www.r-project.org/)

Rencher, A.C., (2002), *Methods of Multivariate Analysis - Second Edition*, New York: Wiley

Sjöström, M., Wold, S., Lindberg, W., Persson, J.A., Martens, H., (1983), “A multivariate calibration problem in analytical chemistry solved by partial least-squares models in latent variables.”, *Analytica Chimica Acta*, Amsterdam, 150, 61-70.

The pls Package, (2007), PLS package for R, version 2.1-0. Wehrens, R. and Mevik, B. (http://mevik.net/work/software/pls.html)

The Unscrambler 9.1 Camo (USA, Norway and India) 30-day free trial installation available at www.camo.com

Tobias, R.D., “An Introduction to Partial Least Squares Regression”, SAS Institute Inc. (http://support.sas.com/techsup/technote/ts509.pdf).