mathcad data analysis extension pack

PRECONDITIONING DATA

Principal Component Analysis

Nipals(DATA, numPC, maxiter,"scale/noscale", Acc)

Nipals2(NIPALS, numAddPC)

loadings(NIPALS) **scores(NIPALS)**

PCAeigenvals(NIPALS) **PCAvariance(NIPALS)**

It is not uncommon to find data sets in which there are a large number of correlated or redundant variables. This not only leads to computational inefficiency in the data analysis, but can also lead to numerical problems (e.g. in matrix inversion steps). What is required in such cases is a method by which we can compress the data into a smaller number of orthogonal variables. There is a group of methods for doing this, which are collectively referred to as factor analysis methods. The most fundamental of these methods is Principal Component Analysis (PCA), which compresses the data to its most dominant factors. In this case, the new, orthogonal, variables are called the principal components. The **Nipals** function takes a multivariate set of data, the desired number of principal components, and the maximum number of iterations, and calculates the loadings, scores (i.e. the principal components), eigenvalues, and percentage of the overall variance explained by the principal components. **Nipals2** allows you to extract additional components from the same data without redoing the whole calculation.

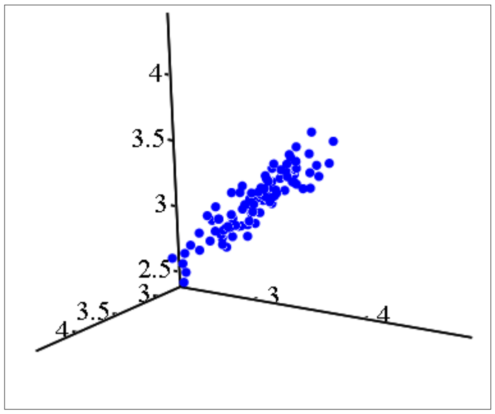
PCA background

To see how PCA works, let's look at a simple data set in which we have three variables (columns), and 100 measurements (rows). This data is artificial, and was created for the purpose of showing how PCA works:

DATA :=

	0	1	2
0	3.616	3.498	3.347
1	3.746	4.509	3.923
2	3.018	2.612	2.612

Since we have only three variables we can plot our data in 3D space:



$\left(DATA^{\langle 0\rangle}, DATA^{\langle 1\rangle}, DATA^{\langle 2\rangle}\right)$

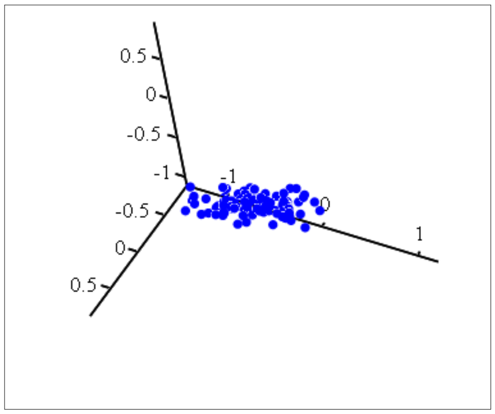
The data is an elliptical cloud of points that almost lies on a plane. This is because the three variables are linearly related, and the deviation from a perfect plane is only because of noise. The first step in PCA is to mean center the data, that is, subtract the mean of each variable. This is done automatically by the **Nipals** function.

$i := 0 .. \text{cols}(DATA) - 1$

$MeanDATA_i := \text{mean}\left(DATA^{\langle i\rangle}\right)$

$i := 0 .. \text{rows}(DATA) - 1$

$Centered^{\langle i\rangle} := \left[\left(DATA^T\right)^{\langle i\rangle} - MeanDATA\right]$ $\qquad$ $Centered := Centered^T$



$\left(\text{Centered}^{\langle 0\rangle}, \text{Centered}^{\langle 1\rangle}, \text{Centered}^{\langle 2\rangle}\right)$

The data is now centered about the origin. Now let's create a new variable space using the **Nipals** function. Calculate three principal components, which is the maximum possible since we started with only three variables

$$\text{NumPC} := 3 \qquad \text{MaxIter} := 1000 \qquad \text{Acc} := 10^{-12}$$

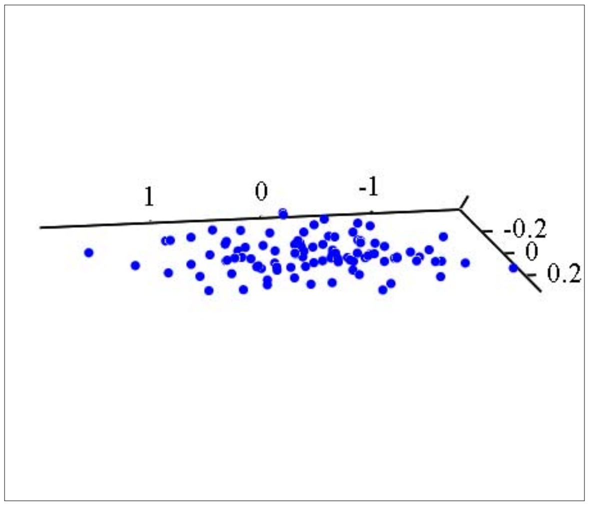
In many applications of PCA it is also desirable to scale the data so that the variables have equal weights, for example when different variables have different units. Scaling each variable (i.e. column of DATA) to unit variance is common, but not appropriate for this data, so we will use no scaling.

$$\text{NIPALS\_Result} := \text{Nipals}(\text{DATA}, \text{NumPC}, \text{MaxIter}, \text{"noscale"}, \text{Acc})$$

The output is a nested matrix (more about the output later). Look at the scores for this data set. The scores are our new variables:

$$\text{SCORES} := \text{NIPALS\_Result}_0 \qquad \text{SCORES} := \text{scores}(\text{NIPALS\_Result})$$

Either syntax works — the **scores** function doesn't require you to remember which entry in the result holds the scores.



$$\left(\text{SCORES}^{\langle 0\rangle}, \text{SCORES}^{\langle 1\rangle}, \text{SCORES}^{\langle 2\rangle}\right)$$

PCA has rotated the data such that the maximum amount of variance can be explained by the first new variable (i.e. the long axis of the elliptical cloud is parallel to the x-axis). The maximum amount of the residual variance, the variance not explained by the first variable, is explained by the second variable. The second variable is of course orthogonal to the first one. Any variance not explained by the first two variables is explained by the third. In this example all the values for the third variable are very small, so for most purposes we can discard it; we have compressed the data.

PCA takes an n-dimensional data space and defines a new set of orthogonal axes (i.e. new variables) that describe the variance in the data in an optimal way. The new variables are optimal in the sense that the first one describes the maximum amount of variation possible (i.e. the maximum variance), the second describes the maximum amount of the remaining variation, etc. These are termed the principal components of the data set. In data sets in which the starting variables are interdependent, or correlated, the higher principal components are close to zero (usually just noise), and can be discarded. The underlying, orthogonal (uncorrelated), variables are termed latent variables, and the number required to describe the data is the rank of the data.

Nipals function

PCA represents the matrix of DATA in the form $\mathrm{DATA} = L \cdot S^T$. The columns of S are called the scores, and the columns of L are called the loading vectors. The columns of S, $S^{\langle a \rangle}$, are the eigenvectors of $\mathrm{DATA} \cdot \mathrm{DATA}^T$, scaled to a length $\sqrt{\tau_a}$, where τ_a are the eigenvalues. It is also worth noting that the loadings are the nonzero eigenvectors of $\mathrm{DATA}^T \cdot \mathrm{DATA}$. This is a large matrix however, so it is not computationally efficient to calculate them this way. Rewrite the equation above as

$$\mathrm{DATA} = L \cdot \mathrm{diag}\left(\sqrt{\tau_a}\right) \cdot V^T$$

which is the singular value decomposition of DATA. For many data sets we could in fact use Mathcad's **svds** function to calculate the scores and loadings, but in general this is undesirable. First, if we use SVD we must calculate all the scores, even though we know we will not need the higher ones. For large, highly redundant data sets SVD is therefore computationally inefficient. Second, SVD may fail numerically if the data set is large and highly redundant. Lastly, the **svds** function requires that the number of columns in the data matrix be larger than the number of rows, which is a problem if we have more measurements than variables (as is the case in the example above).

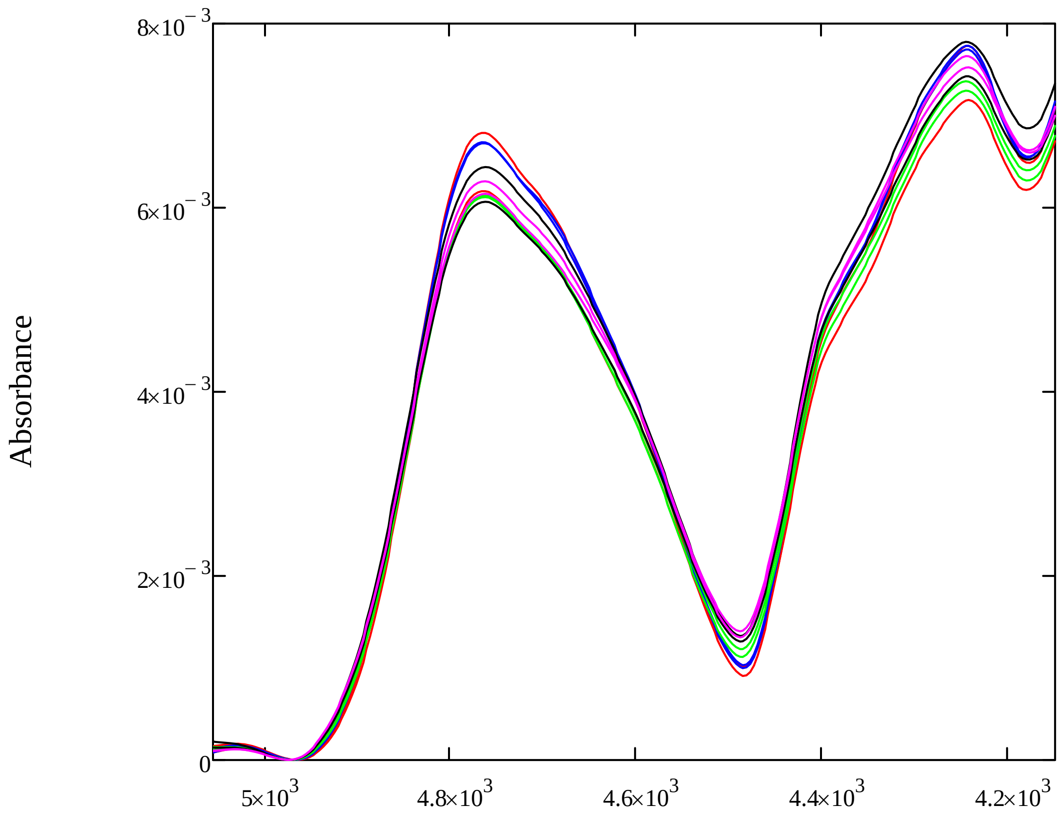
The NIPALS (Nonlinear Iterative Partial Least Squares) algorithm calculates the scores and loadings iteratively, and gets around all of these problems. It is numerically very stable, can handle arbitrary size data sets, and calculates only as many principal components as are needed.

Let's look at a larger, more complex data set. This data represents a portion of the near-infrared (NIR) spectra of pharmaceutical tablets, with five different dosages of the active ingredient (courtesy of Bruker Optics, Inc.). The first column is wavenumber (1/wavelength), in cm^{-1}, there are 5 sequential spectra of each dosage, constituting the remaining 25 columns. The data is from a double blind study in a clinical trial. What is required is that we build a model that allows us to distinguish each dosage based on the spectrum, without knowing what the actual dosages are. This model is needed for QC purposes during the production of more tablets for the trial.

DATA :=

	0	1	2
0	5056	$7.891\cdot10^{-5}$	0.0001
1	5053	$8.647\cdot10^{-5}$	0.0002
2	5049	$9.521\cdot10^{-5}$	0.0002
3	5045	0.0001	0.0002
4	5041	0.0001	...

We need to group the tablets by the percentages of the medication in each one, based on their absorbance spectra. We can plot two spectra of each dosage to see what they look like (by convention, wavenumbers are plotted in decreasing order):



Absorbance
8×10⁻³
6×10⁻³
4×10⁻³
2×10⁻³
0
5×10³
4.8×10³
4.6×10³
4.4×10³
4.2×10³

Wavenumber

Two things are apparent looking at this data:

1) There is no part of the spectrum that can be used to easily distinguish one dosage from another – they all have the same basic form and close absorbance values.

2) Most of the data are redundant. We have 236 points in each spectrum, which means 236 measured variables (absorbance for a particular wavelength of light), but the variation of these points is clearly interrelated.

Start by reducing the dimensionality of the data, and then use the reduced data to determine dosage. First, split X (independent variables - the wavenumbers) and Y (dependent variables - the measured absorbances) data into a vector and a matrix. To match the common convention, put the spectra for each tablet across a row (transpose) so that each column corresponds to an independent variable.

$$\text{Wavnum} := \text{DATA}^{\langle 0 \rangle}$$

$$\text{A1} := \text{submatrix}(\text{DATA}, 0, \text{rows}(\text{DATA}) - 1, 1, \text{cols}(\text{DATA}) - 1)^{T}$$

The **Nipals** function mean centers the data, subtracting the mean spectrum from each row. Remember this when constructing estimates of the spectra with the results. As in the previous example, scaling the data is not appropriate. To begin with we will just get two principal components.

$$\text{numPC} := 2 \qquad \text{maxiter} := 10^{3}$$

$$\text{Nout} := \text{Nipals}(\text{A1}, \text{numPC}, \text{maxiter}, \text{"noscale"}, \text{Acc})$$

Nipals output

$$\text{Nout} = \begin{pmatrix} \{25,2\} \\ \{236,2\} \\ \{2,1\} \\ \{2,1\} \\ \{25,236\} \\ \{6,1\} \end{pmatrix}$$

The output nested matrix has the calculated scores in the first entry, the loadings in the second entry, the cumulative percentage of variance explained in the third entry, and the eigenvalues in the fourth entry. The last two nested matrices are used by the **Nipals2** function if you wish to extract more components.

$$\text{Sc} := \text{Nout}_{0} \qquad \text{Lo} := \text{Nout}_{1}$$

$\text{Cumvar} := \text{Nout}_2 \quad \text{Cumvar} = \begin{pmatrix} 0.69173 \\ 0.98806 \end{pmatrix} \quad \text{PCAvariance}(\text{Nout}) = \begin{pmatrix} 0.69173 \\ 0.98806 \end{pmatrix}$

We can see that, with 2 components, we've explained 98.8% of the variance in the system.

Eigenvalues can also be extracted:

$$\text{PCAeigenvals(Nout)} = \begin{pmatrix} 1.06827 \times 10^{-4} \\ 4.5764 \times 10^{-5} \end{pmatrix} \qquad \text{Nout}_3 = \begin{pmatrix} 1.06827 \times 10^{-4} \\ 4.5764 \times 10^{-5} \end{pmatrix}$$

Estimates of the original spectra can be constructed by multiplying the matrix of loading vectors by the matrix of scores, and adding the mean spectrum subtracted by **Nipals**. The first spectrum is the first column of the reconstruction matrix, and so on. The scores represent the proportions in which the loading vectors are added to recreate the original spectra; they can be thought of as intensities.

$$\text{Estimate} := \text{Lo}\cdot\text{Sc}^{T} \qquad\qquad i := 0 .. \text{cols(A1)} - 1$$

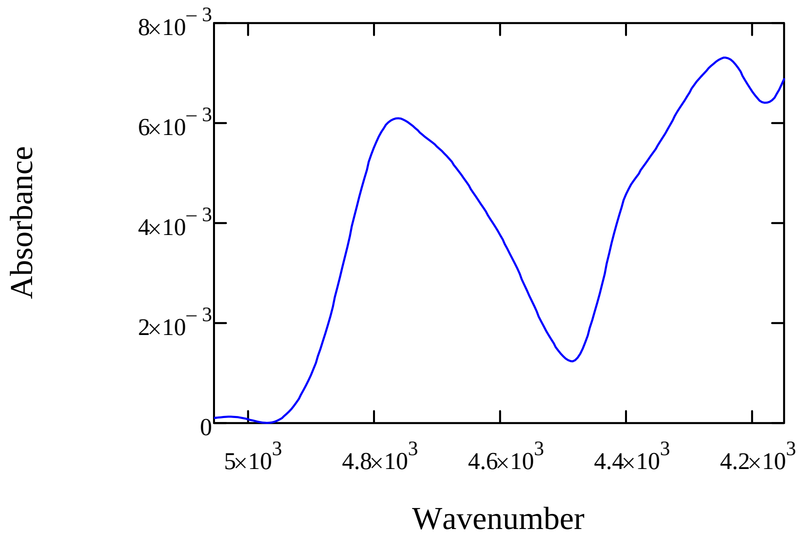
$$\text{MeanA}_i := \text{mean}\left(\text{A1}^{\langle i\rangle}\right)$$

You can pick which spectrum you wish to reconstruct:

$$\text{spec} := 12$$

We see that all the spectra can be well represented using only two factors.

$$\text{EstSpec1} := \text{Estimate}^{\langle \text{spec}\rangle} + \text{MeanA}$$



Using the Principal Components

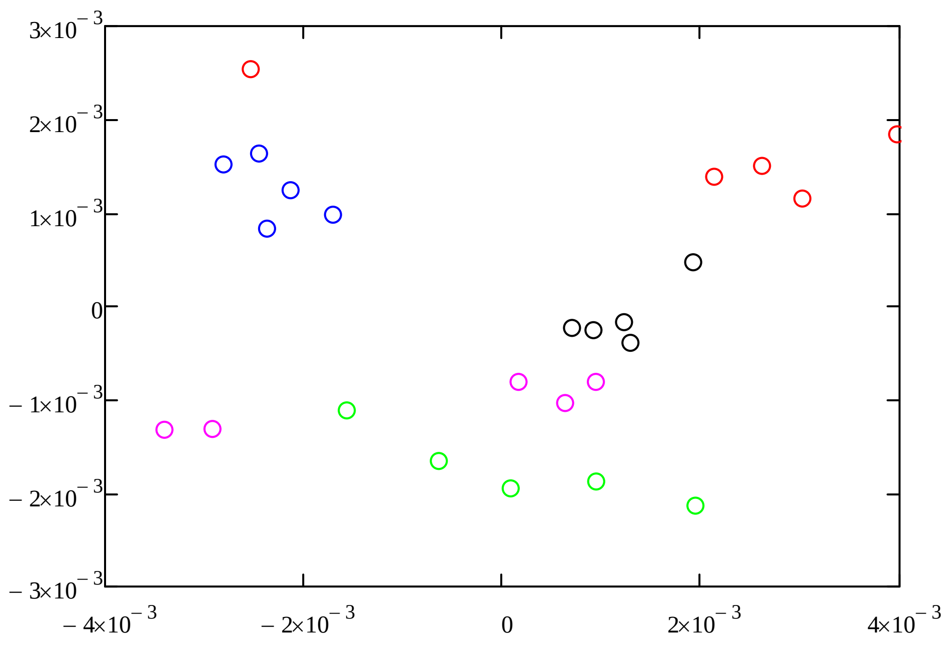
Let's examine the new variables, i.e. the scores, and see what they tell us about the system of spectra.

We will rearrange our scores into two matrices. Each column of the matrix represents the scores for one of the 5 tablet dosages.

$p := 0..4 \qquad q := 0..4$

$Xdata_{p,q} := \left(Sc^{\langle 0 \rangle}\right)_{p+q\cdot 5} \qquad Ydata_{p,q} := \left(Sc^{\langle 1 \rangle}\right)_{p+q\cdot 5}$

We can plot the scores for the first factor against those for the second. Each dosage is shown in a different color.



Some grouping of the data is evident, but we still cannot adequately distinguish each dosage from the others. Adding a third score to the plot might help. We can use the **Nipals2** function to calculate a few more principal components without having to repeat the previous calculations.

$morePC := 4 \qquad Nout := Nipals2(Nout, morePC)$

$$Nout = \begin{pmatrix} \{25,6\} \\ \{236,6\} \\ \{6,1\} \\ \{6,1\} \\ \{25,236\} \\ \{6,1\} \end{pmatrix}$$

The output matrix is of the same form as that for **Nipals**, but with additional columns and rows in the nested entries to correspond to the additional calculated components. Note that the number of scores and loadings has now increased to 6.

$S1 := Nout_0$ $L1 := Nout_1$

Note that

$L1 := \mathrm{loadings}(Nout)$ also finds the loadings.

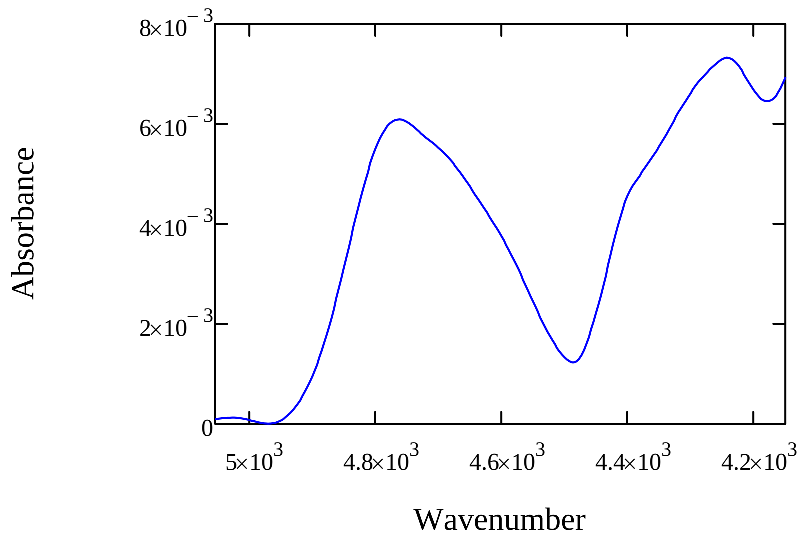
$\text{Cumvar1} := \text{Nout}_2$ $\qquad$ $\text{Cumvar1} = \begin{pmatrix} 0.69173 \\ 0.98806 \\ 0.99416 \\ 0.99619 \\ 0.99779 \\ 0.99844 \end{pmatrix}$

These are the new estimated spectra:

$\text{Estimate} := \text{L1} \cdot \text{S1}^T$

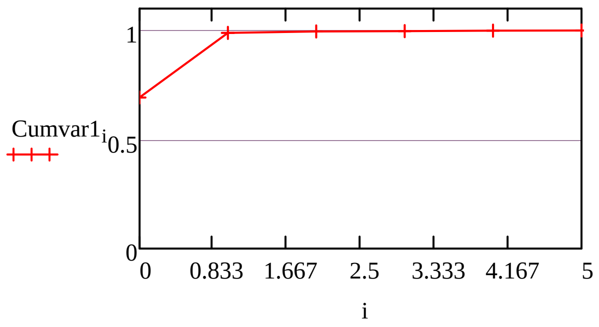
$\text{spec} := 12$

$\text{EstSpec1} := \text{Estimate}^{\langle \text{spec} \rangle} + \text{MeanA}$



Now that we have a larger number of principal components, it's instructive to look at the behavior of the cumulative explained variance:

$i := 0 .. \text{last}(\text{Cumvar1})$



Cumvar1$_i$
1
0.5
0
0
0.833
1.667
2.5
3.333
4.167
5
i

The fact that only 69% of the variance is explained by the first PC is perhaps initially a little surprising. If the only source of variance were the dosage of active ingredient then it would be possible to describe the data completely with only one variable. Yet we have at least two significant sources of variance. We note also that a third PC is required to explain more than 99% of the variance, and, as we will see below, it is this third PC that is key to being able to group the data by dosage.

This brings us to an important point about PCA: it compresses the data to the most dominant factors, not the most relevant factors. The amount of active ingredient in each tablet is in fact quite small, and there are other sources of variation (e.g. light scattering) that cause bigger changes in the spectra. Each source of physical variance is distributed between all the factors. There are factor analysis methods that attempt to compress to the most relevant factors, but they all require some sort of a-priori information (otherwise, "most relevant" has no meaning). PCA requires no such information, and is therefore applicable to any multivariate data set in which the variables are at least partially redundant.

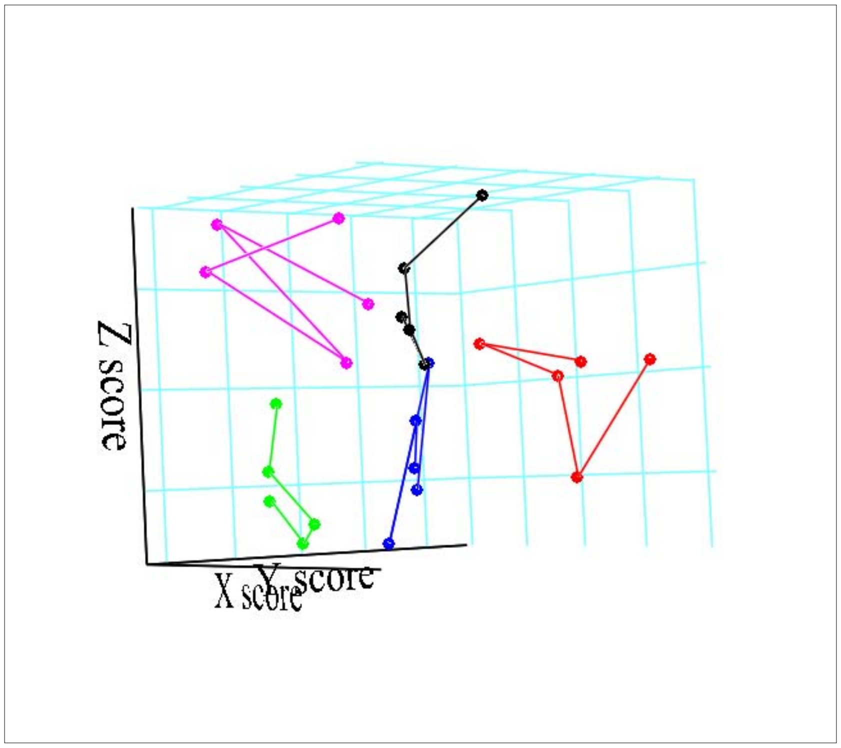
Now we can plot any three scores in 3D space.

$\mathrm{Xscore} := 0 \qquad \mathrm{Yscore} := 1 \qquad \mathrm{Zscore} := 2$

$$\mathrm{Xdata}_{p,q} := \left(\mathrm{Sc}^{\langle \mathrm{Xscore} \rangle}\right)_{p+q\cdot 5}$$

$$\mathrm{Ydata}_{p,q} := \left(\mathrm{Sc}^{\langle \mathrm{Yscore} \rangle}\right)_{p+q\cdot 5}$$

$$\mathrm{Zdata}_{p,q} := \left(\mathrm{S1}^{\langle \mathrm{Zscore} \rangle}\right)_{p+q\cdot 5}$$



We can see that the tablets fall into five separate groups. By reducing the dimensionality of the data from 236 to only the three most dominant principal components we can visualize the grouping of the tablets.

Predicting new data

Now that we have a model, we can take the spectrum of an unknown tablet, and project it into our new variable space.

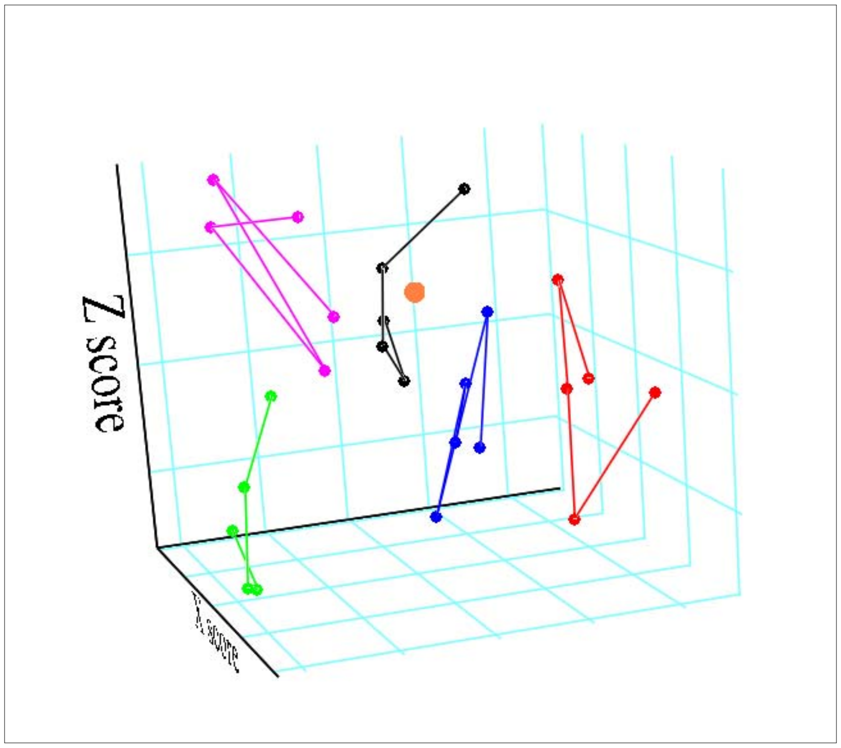
Unknown :=

	0
0	$1.2847 \cdot 10^{-5}$
1	$1.0279 \cdot 10^{-5}$
2	$7.884 \cdot 10^{-6}$
3	$5.586 \cdot 10^{-6}$

To do this we use our loadings to get an estimate of the scores for the new spectrum. We only want to use the first three loadings, so we will remove the extra columns:

$$T := \text{Unknown}^{T} \cdot \text{submatrix}(L1, 0, \text{rows}(L1) - 1, 0, 2)$$

$Ux_0 := T_{0,0}$ $\quad Uy_0 := T_{0,1}$ $\quad Uz_0 := T_{0,2}$



We see that our unknown tablet (orange) belongs in dosage group 3.

PCA is an important data preprocessing step for large, redundant data sets. Data is compressed to a smaller number of variables reflecting the majority of the variation. In principle, any analysis (cluster analysis, regression, etc.) can be performed on the compressed data. We can also use PCA in the context of exploratory data analysis: we can find the rank of the data, we can look at the loadings, etc., to determine what the data tells us and what steps we should take next.

References

Spectroscopic data courtesy of Bruker Optics, Inc., all rights reserved, used by permission.

Wold, H. and Krishnaiah, P.R. (Ed) (1966), *Multivariate Analysis*, Academic Press, New York. p. 391.