Inf2 – Foundations of Data Science 2022
Topic: Dealing with high dimensions – PCA

5th November 2022

Learning outcomes
By the end of this topic you should be able to:

1. Explain what sorts of datasets principal components analysis (PCA) can help us to understand

2. Explain the principle of how PCA works

3. Outline the steps involved in the derivation of PCA

4. Interpret the results of a PCA analysis

Recommended reading:
Witten, Frank, Hall and Pal *Data mining*, 4th 3d, pp 304–307 contains an overview of PCA. Different sources use different notation, so it may be least confusing just to follow these notes.

1 The principle of Principal Components Analysis (PCA)

The challenges of high dimensions In the multiple regression topic, in the student grade prediction example, we were beginning to see two challenges of dealing with more than one independent variable:

The challenge of visualisation We can see a lot in the paired correlation plots. With 4 independent variables, the visualisation works, but what about if we had 26 variables? The Scottish Index of Multiple Deprivation (SIMD, Table 1) records 26 variables for each of 6527 data zones in Scotland. A 26×26 grid of scatter plots is going to be difficult to read.

The challenge of interpretation In the grades example, the test grades (independent variables) were correlated, which made the interpretation of the regression coefficients challenging – and this was with only 4 independent variables. In the SIMD example, we might expect many of the 26 variables to be correlated, e.g. the time it takes to drive to the nearest primary school and the time it takes to drive to the nearest secondary school.

Table 1: Scottish Index of Multiple Deprivation, 2016 edition (Scottish Government, 2016). `https://simd.scot`. It has $n = 6527$ data points (postcode zones), each associated with $D = 26$ variables.

Location	Employ-ment	Illness	Attain-ment	Drive Primary	Drive Secondary	Crime	...
Macduff	10	95	5.3	1.5	6.6	249	...
Kemnay	3	40	5.3	2.4	2.4	168	...
Hilton	0	10	6.3	2.2	3.0	144	...
Ruchill	8	130	4.9	1.7	5.6	318	...
Belmont	2	50	6.1	3.1	3.2	129	...
...	...	...	...	...	...	...	...

There is also another problem with high-dimensional data, called the **curse of dimensionality**: essentially a large number of dimensions makes is harder for distance-based methods such as clustering and nearest neighbours to work effectively – we'll come back to the curse of dimensionality in the following lectures on clustering and nearest-neighbour methods.

In **dimensionality reduction** methods these challenges are addressed by reducing the number of dimensions in the data while retaining as much useful information as possible. There are a number of dimensionality reduction methods which differ in what aspects of the data they preserve.

Principal components analysis We are going to discuss one method of dimensionality reduction called **principal components analysis** (PCA).

PCA can be applied to a set of D numeric variables with n datapoints. In contrast to linear regression, all variables are treated equally: there is no dependent variable that we are trying to predict, just a set of variables whose structure we're trying to understand better. The result of PCA is a set of up to D new variables (with n datapoints). We can keep $k \leq D$ of the most informative new variables.

In PCA the objectives are:

1. change the angle we view the data from to see things clearly
2. ignore small details in the data that don't affect the big picture.

We'll specify these objectives more precisely and explain how PCA works later. First, we will show the results when PCA is applied to the SIMD example (Table 1).

Example of PCA We can use PCA to reduce the number of variables D in the SIMD data from $D = 26$ to $k = 2$, allowing us to visualise all $n = 6527$ data points (Figure 1). In this plot, the ith datapoint has coordinates (t_{i1}, t_{i2}) in which each coordinate is a linear combination of

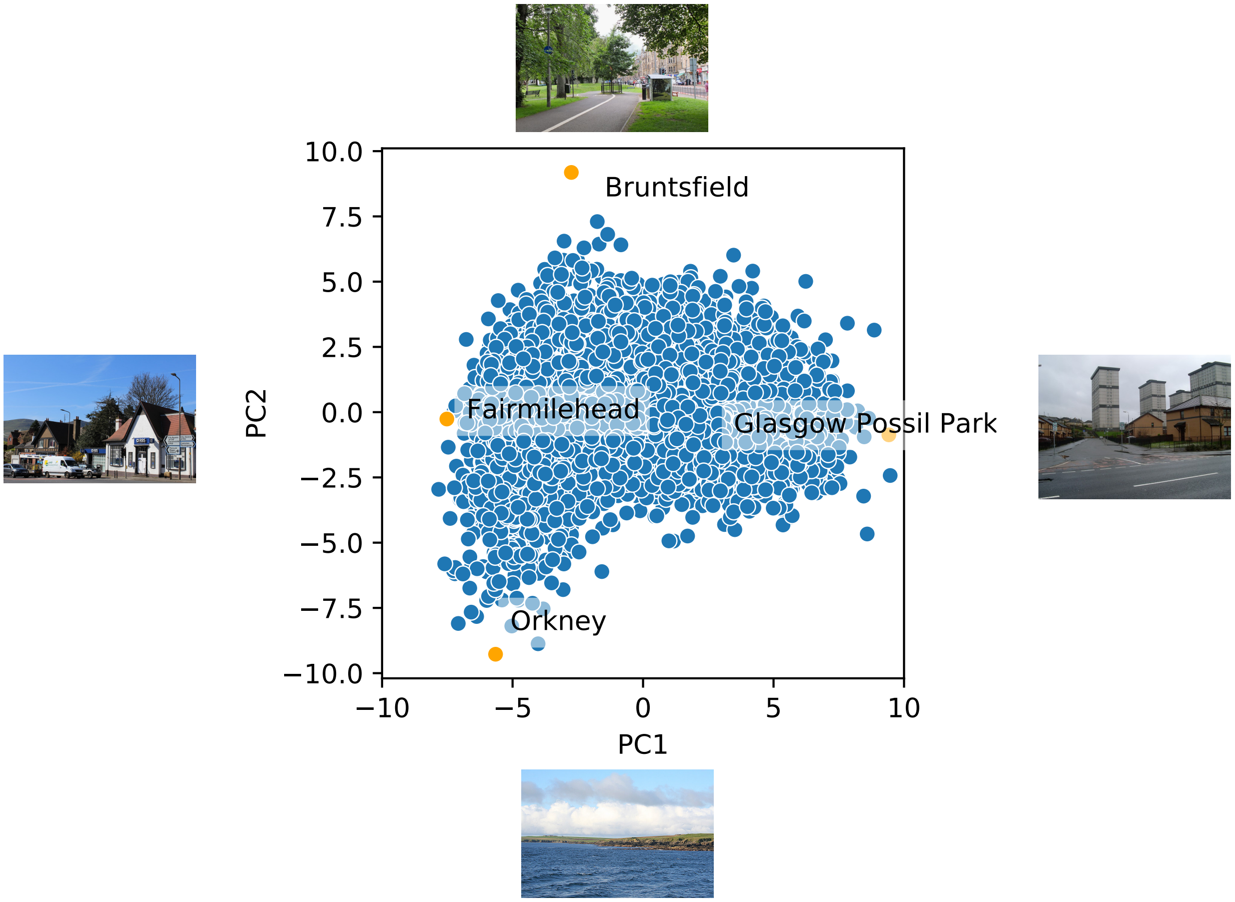


Figure 1: Scatter plot of first and second principal component scores (PC1 and PC2) of 6527 data points in the SIMD dataset (blue dots). Locations of 4 data zones are indicated in orange dots next to an image from that data zone. All photos released under CC licence from geograph.co.uk. Credits: Orkney © Des Colhoun; Possil Park © Stephen Sweeney; Brunstfield © Leslie Barrie; Fairmilehead © Jim Barton.

the standardised[1] data z_{ij} shown in Table 1:

$$\begin{aligned} t_{i1} &= p_{11}z_{i1} + p_{21}z_{i2} + \cdots + p_{D1}z_{iD} \\ t_{i2} &= p_{12}z_{i1} + p_{22}z_{i2} + \cdots + p_{D2}z_{iD} \end{aligned} \tag{1}$$

The weights $p_{11}, p_{21}, \ldots, p_{D1}$ are elements of the **first principal component** and t_{i1} is the first principal **component score** of the ith datapoint; we will explain how to find them later. Likewise, $p_{12}, p_{22}, \ldots, p_{D2}$ form the second principal component and t_{i1} is the second principal **component score** of datapoint i. The weights in the principal component indicate how much influence each original variable has over each principal component score – sometimes they are referred to as **loadings** or **weights**. The axes in Figure 1 are labelled PC1 (first principal component – "PC" stands for principal component) and PC2 (second principal component).

[1] Remember from the video on variance that we standardise the jth variable $x^{(j)}$ by subtracting its mean $\overline{x}^{(j)}$ and dividing by its standard deviation s_j, so that $z_{ij} = (x_{ij} - \overline{x}^{(j)})/s_j$. Generally, the data we supply to PCA do not need to be standardised, but we still do need to subtract the mean in order to compute the component scores.

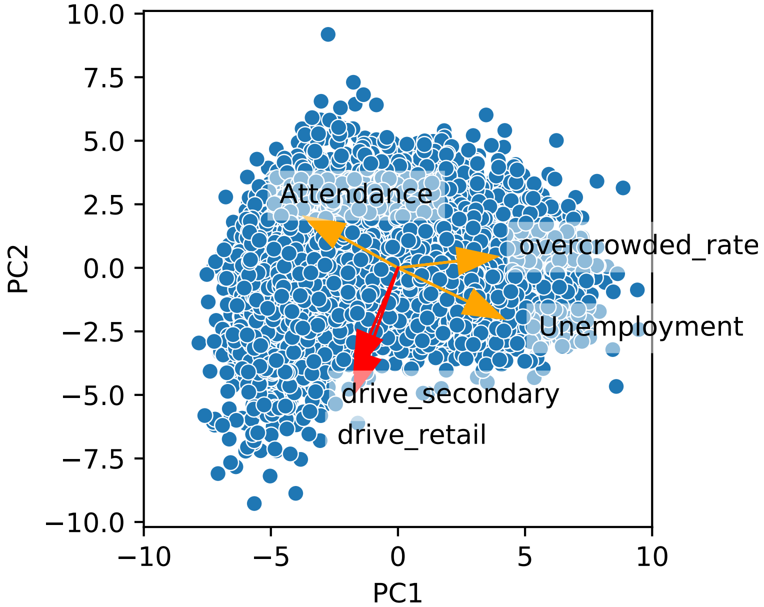


Figure 2: Scatter plots of first and second principal component scores of SIMD data zones (blue dots). The projection of three original variables related to deprivation are shown as orange arrows emanating from the origin. High unemployment and overcrowded rate are found in areas with higher deprivation, whereas high school attendance is found in areas with low deprivation. These vectors are more closely aligned with the first principal component (PC1), which we therefore interpret as "Deprivation". Red arrows indicate the projections of the time take to drive to the nearest secondary school or retail outlet. As these are aligned with PC2, we therefore interpret PC2 as being related to distance to services, or "Remoteness".

To see the influence of each original variable on PC1 and PC2 scores, we can project the jth original variable onto the plot by setting z_{ij} to 1 and all the other z's to 0 in Equation 1. In this case, the coordinates we'll be plotting are (p_{j1}, p_{j2}). The orange arrows in Figure 2 show the projections of the variables for unemployment, overcrowding (in housing) and school attendance. Unemployment and overcrowding have high PC1 scores. In contrast, school attendance has a low PC1 score. This all makes sense if we identify the first component score with "Deprivation". We can rephrase the previous sentences as "unemployment and overcrowding are found in areas of high deprivation and high school attendance is found in areas of low deprivation".

The red arrows in Figure 2 show the projections of the time to drive to the nearest retail outlet and time to drive to the nearest secondary school. These vectors have higher magnitude PC2 scores than PC1 scores. We therefore identify PC2 as being to do with "remoteness" – low values of PC2 indicate the zone is more remote.

Note that the correlation between the PC1 and PC2 scores is zero. It is a general property of PCA there are no correlations between the scores of different principal components.

In this particular example, the visualisation shows a unimodal distribution of data with little obvious structure. Later on in the course we will see examples where PCA reveals clusters of data – though still with zero correlation.

Even if no structure is apparent, reducing the dimensionality of the data can be useful for further analysis. For example, suppose we have data on cancer screening rates in each data SIMD zone, we could then do multiple regression of the cancer screening rate on the new deprivation and remoteness variables. This is probably going to give us coefficients that are a lot more interpretable than regressing on all 26 variables.

Projecting principal component scores back into the data space Suppose we have identified the first two principal component scores t_{i1} and t_{i2} of area i. We might wish to project them back into the data space, to see what the original variables looked like. To do this we can use the following equations to give approximations (indicated by the tilde) to the original standardised variables:

$$\begin{aligned}\tilde{z}_{i1} &= p_{11}t_{i1} + p_{12}t_{i2}\\ \tilde{z}_{i2} &= p_{21}t_{i1} + p_{22}t_{i2}\\ &\vdots\\ \tilde{z}_{iD} &= p_{D1}t_{i1} + p_{D2}t_{i2}\end{aligned} \tag{2}$$

We can include more terms for higher PCs, right up to the Dth PC. In general, the jth component of the i data point is given:

$$\begin{aligned}z_{ij} &= p_{j1}t_{i1} + p_{j2}t_{i2} + \dots p_{jD}t_{iD}\\ &= \sum_{k=1}^{D} p_{jk}t_{ik}\end{aligned} \tag{3}$$

Once we've got the standardised variables, we can convert back to the original variables using the formula $x_{ij} = z_{ij}s_j + \overline{x}_j$.

Principal component equations in vector notation The equations used so far may make more sense when expressed as vectors. The jth principal component is actually a vector in the original data space:

$$\mathbf{p}_j = (p_{1j}, p_{2j}, \dots, p_{Dj})^T \tag{4}$$

All the principal component vectors are orthogonal to each other. With this notation we can write Equation 2 as a linear combination of the principal component vectors, weighted by the principal component scores:

$$\mathbf{z}_i = t_{i1}\mathbf{p}_1 + t_{i2}\mathbf{p}_2 + \dots \tag{5}$$

The dots indicate that we could go up to $t_{iD}\mathbf{p}_D$.

We can rewrite Equation 1, in which we computed the scores, as the scalar product of the ith standardised data point and the jth principal component:

$$t_{ij} = \mathbf{z}_i \cdot \mathbf{p}_j \tag{6}$$

We'll extend this notation to matrix notation in the derivation.

Table 2: Imaginary data about Informatics students' preferences for programming languages and drinks.

Student ID	Language	Drink
1	9	8
2	3	1
3	8	7
4	2	2
5	3	3
6	8	6
7	2	3
8	8	8
9	1	2
10	6	7

2 Principle of finding principal components

A 2D example We'll now discuss the principle of how to determine the principal components with an imaginary 2D example. Suppose we ask if is there are different types of Informatics students, perhaps based on their preferences for programming languages and for drinks. We ask students if they prefer, on a scale of 1–9, Haskell (1) to Java (9), and if they prefer Tea (1) to Coffee (9), and find the data in Table 2.

Plotting the data (Figure 3 left) shows that students' preferences for drinks and programming languages are correlated. It seems that we could characterise every Informatics student by one number that is low if they like Haskell and tea, and high if they like Java and coffee. If we could rotate the axes (Figure 3 right), the new x-axis would give us this number.

Once we've done the rotation (changed the angle), we end up with the data plotted against a new set of axes, which are the principal components (Figure 4, top). The new x-axis, which tells us a lot about the students' preferences for Java and coffee or tea and Haskell, is the first principal component (PC1). The new y axis is the second principal component (PC2). It is worth noting two points:

- The correlation between the new PC1 and PC2 scores is zero. It is a general property of PCA that correlations between scores is zero.
- We have not lost any information about the data; we can reconstruct the original data by reversing the rotation. It is a general property of PCA that it is possible to reconstruct the data if scores of all D principal components are retained.

The second principal component doesn't seem so informative, so we could just ignore it altogether (Figure 4, bottom). Thus, we have ignored small details in the data that don't affect the big picture. We have performed dimensionality reduction by reducing the number of values describing each data point from two to one.

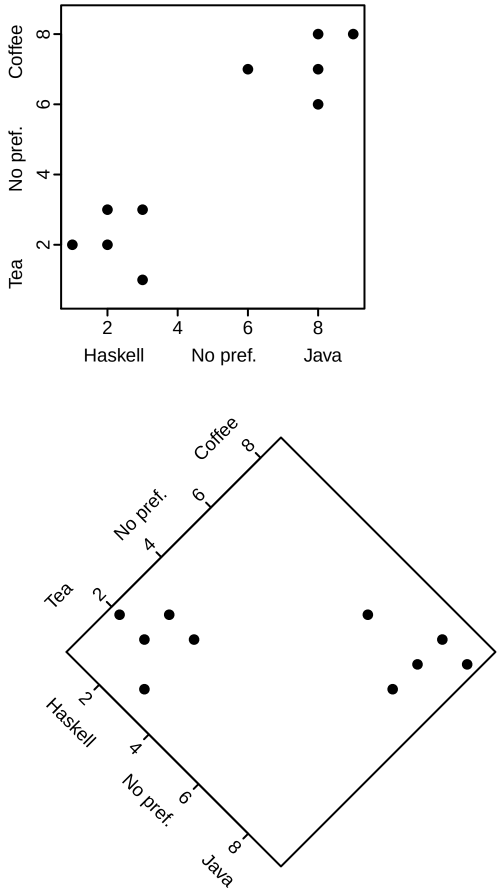
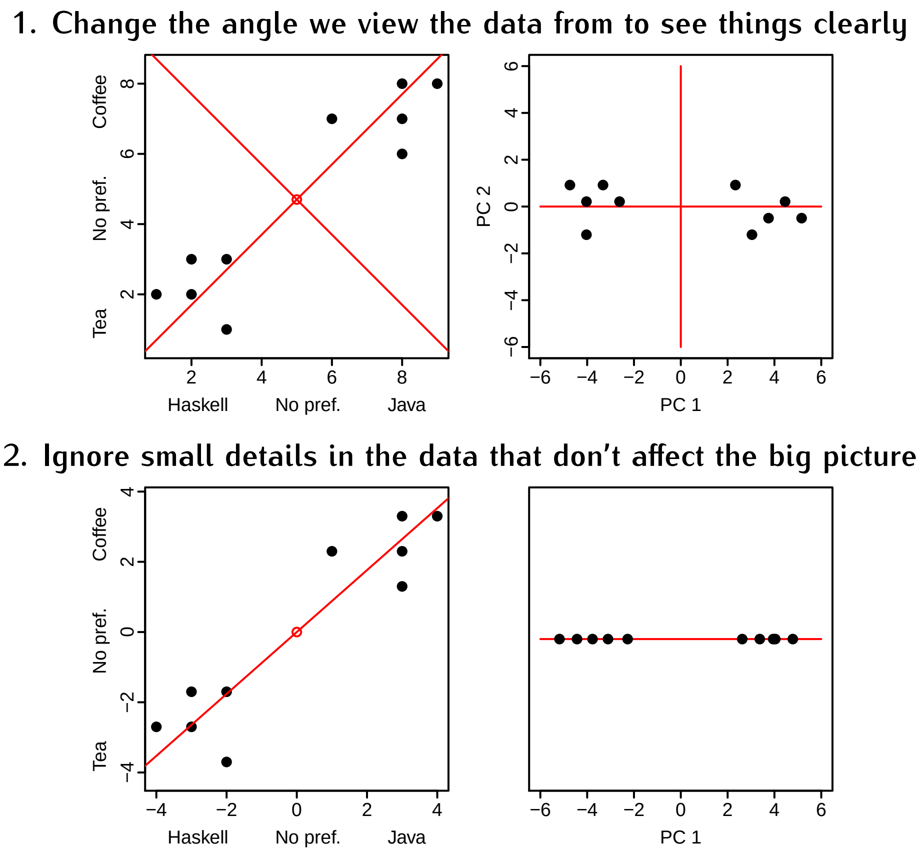


Figure 3: Informatics students' preferences for drinks and programming languages, as plotted initially (left), and rotated (right).



1. Change the angle we view the data from to see things clearly

2. Ignore small details in the data that don't affect the big picture

Figure 4: Visualisation of how PCA achieves the two objectives in the text.

Objective of rotation There are two questions that we haven't answered so far:

1. How do we choose how much to rotate the axes?
2. What counts as "informative"?

The answer to both questions is "variance". In Figure 4 (top), the variance of the data in the PC1 direction is much greater than the variance of the data in the PC2 direction. The high variance PC1 is telling us a lot about the informatics students, whereas the low variance PC2 tells us little. Therefore, in order to choose how to rotate the axes, we use the variance as an objective. In fact there are two ways of formulating PCA:

1. Maximum variance formulation: find an axis that maximises the variance of the data projected onto it
2. Minimum variance formulation: find an axis that minimises the variance of the data projected onto it

It doesn't matter which formulation we use; the answer is the same either way.

Explained variance The variance in the original x (Programming language) and y (Drink) directions was 9.7 and 7.7. The sum of these two variances is the total variance, i.e. 17.4. It turns out that the sum of the variance along the principal components is exactly the same. However, the variance of the PC1 scores is 16.5, i.e. 96% of the total variance. We therefore say the PC1 explains 96% of the variance.

More than 2D In general, we can find D principal components in D dimensions. The principal components are all orthogonal to each other, and each principal component explains a certain fraction of the variance. We order the principal components from the one that explains most variance to the one that explains least.

In the SIMD example, the first principal component explains 41.7% of the data and the second explains a further 15.2%. Thus, the first two principal components together explain 56.9% of the variance. We can visualise how much each principal component explains in a **scree plot** or **cumulative scree plot** (Figure 5).

How many components to choose? Obviously if we are visualising data, we can only look straightforwardly at up to 3 dimensions. The scree plot helps us to choose how many components to include if we are using PCA as a preprocessing step. A rule of thumb is to use the components to the left of the "knee" or "elbow" of the scree plot, i.e. the point where the gradient changes sharply. In Figure 5 this point is indicated in red, and the rule of thumb would suggest that we use PC1 and PC2. There are more principled ways of choosing, which we won't cover at this point, and it may also be that successful application of PCA requires more components.

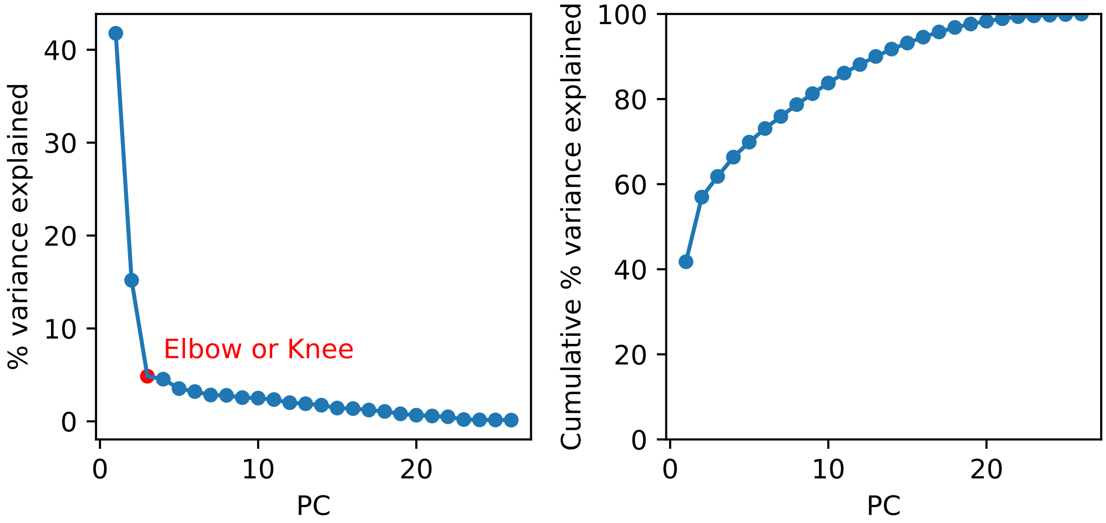


Figure 5: Scree plot for PCA applied to SIMD example (left). The elbow (or knee) is indicated in red. The Cumulative scree plot (right).

In the next video, we'll look at the maths of how to find the directions of the principal components and the associated variances. However, you should already know enough to skip to the video after, which is about applying PCA to help with a regression problem.

3 PCA and regression

PCA as preprocessing PCA is often used as a preprocessing step before another method, e.g. linear regression or K-means. Here we'll see how it can help simplify the grades example from the linear regression lecture. Figure 6 shows the results of applying PCA to the independent variables in this example. Note the correlations between the PC scores are all zero; the general property of PCA already mentioned. However, the correlations between the PC scores and the Grade are non-zero.

We can regress the Grade y on the principal component scores $t^{(1)}$, $t^{(2)} \dots$:

$$y = \hat{\beta}_0 + \hat{\beta}_1 t^{(1)} + \hat{\beta}_2 t^{(2)} + \dots \tag{7}$$

When we regress on all 4 PC scores, we get exactly the same predictions and coefficient of determination as we do for regressing on all variables (Table 3). This makes sense, since by keeping all 4 components we have not lost any information about the data. It is more surprising that the coefficient of determination with if we regress on only the first two PC scores is almost as high. Furthermore, the adjusted coefficient of determination is actually higher when regression on the first two principal components, due to there being fewer variables. There is no combination of any two of the original variables that gives as high a coefficient of determination.

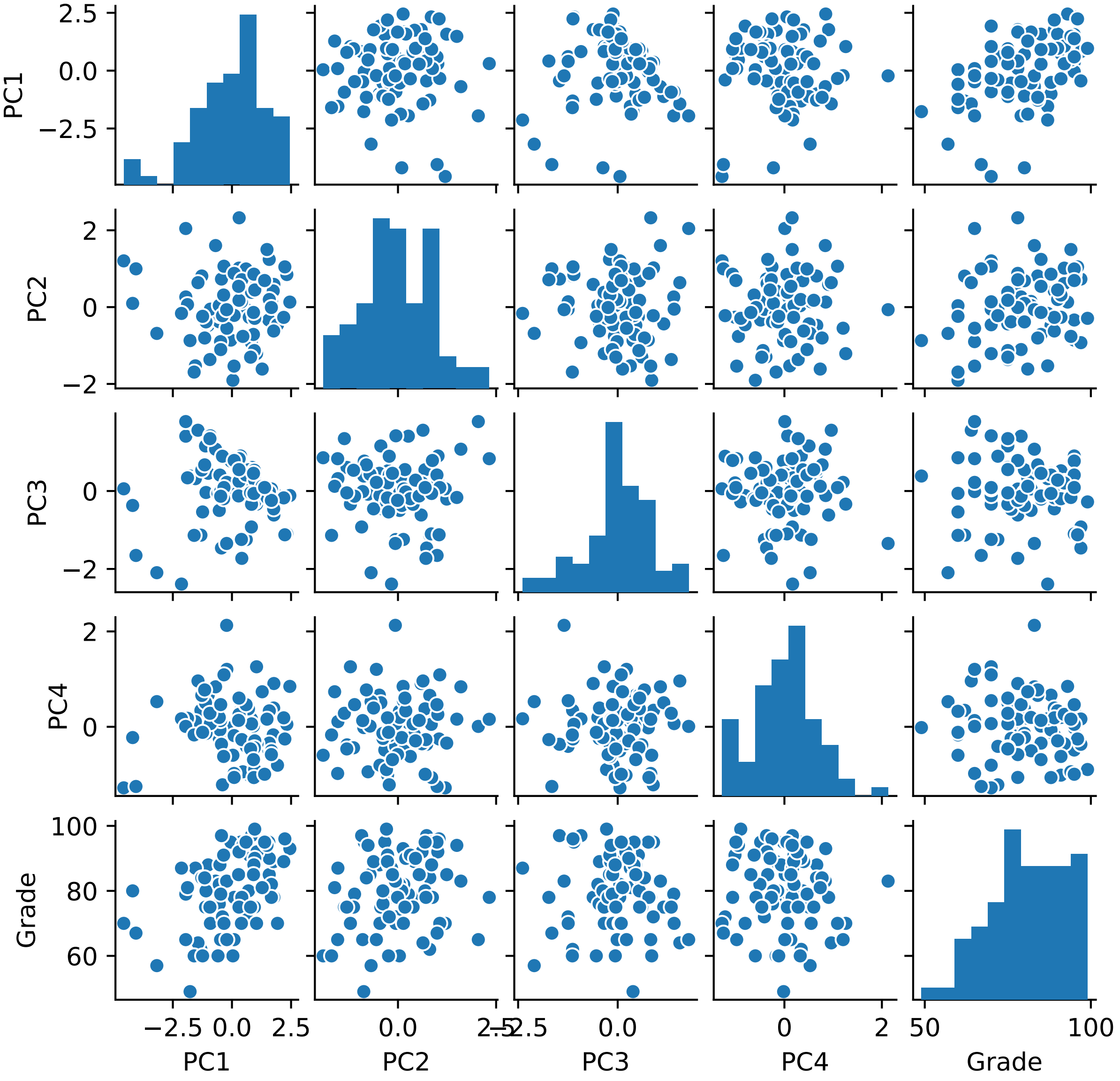


Figure 6: PCA scores of independent variables in Grades example (see Multiple regression lecture notes).

Table 3: Coefficient of determination and adjusted coefficient of determination for regression of grades on original variables and on 2 or 4 PC scores.

	4 Original variables	4 PC scores	2 PC scores
R^2	0.289	0.289	0.282
R^2_a	0.251	0.251	0.263

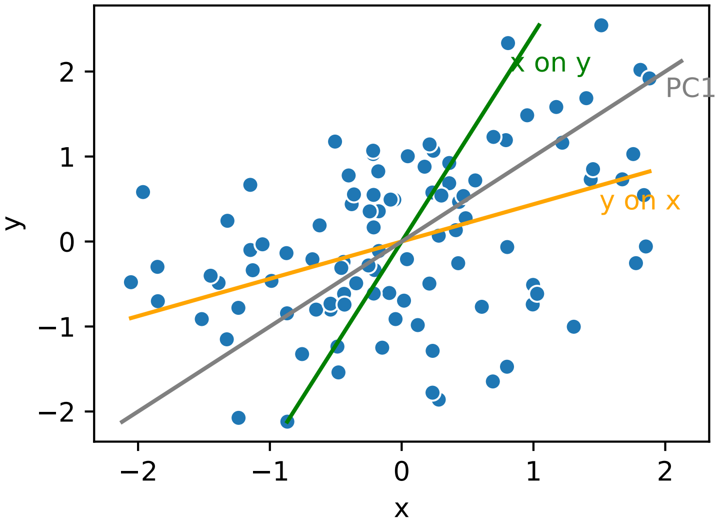


Figure 7: Regression of y on x, x on y and the first principal component of data with a correlation coefficient $r = 0.5$.

This example demonstrates that PCA can be a useful preprocessing step for regression, by decorrelating the variables.

PCA and linear regression lines Thinking back to linear regression, we remember the distinction between the regression lines of y on x and x on y. In two dimensions there is now a 3rd line: the first principal component. This goes right between the regression lines, and is probably what you would think the line of best fit to the data is. In fact, it is a line of best fit. It's the line that minimises the sum of the squared distances from the data points to the line, rather than minimising the error in predicting y or x.

4 Derivation of PCA

Overview of derivation Here are the steps we'll take in our derivation:

1. Define variance along the original axes

2. Project data onto rotated axes

3. Compute variance in these axes

4. Find direction of the axis that maximises variance of data projected onto it (1st principal component, PC 1)

5. Interpret

6. Find the 2nd principal component (PC 2)

7. Quantify what is lost by dimensionality reduction

This list may seem overwhelming, but it actually boils down to about 4 lines of code (assuming some helper functions), shown in Listing 1.

Step 1: Defining variance along original axes We've already met a lot of the mathematical machinery we need in the multiple regression topic. We'll assume now that we have D variables $x^{(1)}, \dots, x^{(D)}$, and that we have defined zero-mean versions of the variables $x^*_{ij} = x_{ij} - \overline{x}^{(j)}$. Usually, as in the SIMD example, we start off with standardised variables anyway. We can arrange these zero-mean variables in an $n \times D$ matrix,

$$\mathsf{X} = \begin{pmatrix} x^*_{11} & \cdots & x^*_{1D} \\ \vdots & & \vdots \\ x^*_{n1} & \cdots & x^*_{nD} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^{(1)} & \cdots & \mathbf{x}^{(D)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} \tag{8}$$

which it can be helpful to write in terms of the D $n \times 1$ vectors representing all the data in each dimension, or as the transposes of the n $D \times 1$ vectors representing each data point.

We've also met the covariance matrix, the $D \times D$ matrix that's derived from the data matrix:

$$\mathsf{S} = \begin{pmatrix} s_{11} & \cdots & s_{1D} \\ \vdots & & \vdots \\ s_{D1} & \cdots & s_{DD} \end{pmatrix} = \frac{1}{n-1}\mathsf{X}^T\mathsf{X} \tag{9}$$

The variance in the original axes is s_{11} and $_{22}$. The covariance matrix in our toy example is:

$$\mathsf{S} = \begin{pmatrix} 9.7 & 8.0 \\ 8.0 & 7.7 \end{pmatrix}$$

Step 2: Project data onto a new axis We'll define the new axis by the **unit vector p** (Figure 8). The projection of a data point $\mathbf{x}_i$ onto this axis (its **component score**) is (as per Equation 1)

$$t_i = \mathbf{p}^T\mathbf{x}_i = p_1 x_{i1} + p_2 x_{i2} \tag{10}$$

Listing 1: Listing of PCA. We are assuming the existance of helperfunctions `standardize()` and `sort_eigenvalues()`.

```
import numpy as np

def standardize(X)...

def sort_eigenvalues(lambda, P)...

def pca(X):
    """Given a data matrix X with n rows and D columns,
       return principal components (P) and
       eigenvalues (lambda)"""
    # Standardise the data X
    Z = standardize(X)
    # Compute the covariance matrix S
    S = np.cov(Z)
    # Find unsorted eigenvectors (P) and eigenvalues (lambda)
    lambdas, P = np.linalg.eig(S)
    # Sort the eigenvectors and eigenvalues
    # in order of largest eigenvalues
    lambdas, P = sort_eigenvalues(lambdas, P)
    # The eigenvalues (lambdas) are proprtional
    # to the amount of variance explained
    for i in range(len(lambdas)):
        print('PC' + str(i+1) + '␣explains␣' +
        str(round((lambdas[i] / np.sum(lambdas))*100)) +
        '%␣of␣the␣variance.')
    return(lambdas, P)
```

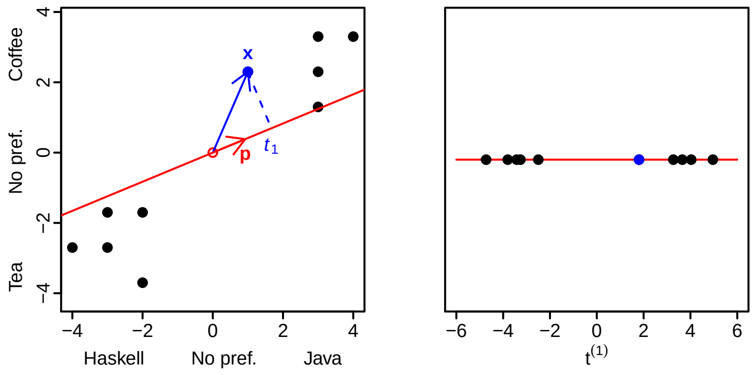


Figure 8: Projection of a data point **x** onto a unit vector **p**.

Step 3: Compute variance in these axes The definition of the variance of the $t^{(1)}$ is:

$$s_t^2 = \frac{1}{n-1}\sum_{i=1}^{n} t_i^2 \tag{11}$$

If we substitute Equation 10 into this equation we get the following:

$$\begin{aligned} s_t^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(p_1 x_{i1} + p_2 x_{i2})^2 \\ &= \frac{1}{n-1}\sum_{i=1}^{n}(p_1 p_2)\begin{pmatrix} \sum_i x_{i1}x_{x1} & \sum_i x_{i1}x_{x2} \\ \sum_i x_{i2}x_{x1} & \sum_i x_{i2}x_{x2} \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \\ &= \mathbf{p}^T\mathbf{S}\mathbf{p} \end{aligned} \tag{12}$$

Our old friend the covariance matrix has reappeared. Although we've demonstrated this in 2 dimensions, the equation is still valid in D dimension.

Step 4: Find direction of axis that maximises variance of data projected onto it (1st principal component, PC 1) We now have an expression for the variance in the component scores for any direction of **p** we now want to find the direction of **p** that maximises that variance. We have a constraint that **p** is of unit length, so $|\mathbf{p}| = 1$.

This is a constrained optimisation problem, which we can solve using Lagrange multipliers and differentiation. We won't show the details here, but the result is the following equation:

$$\mathbf{S}\mathbf{p} = \lambda\mathbf{p}$$

Hopefully you recognise this equation. Its solutions are:

1. $\lambda = \lambda_1$, $\mathbf{p} = \mathbf{e}_1$, where λ_1 is the biggest **eigenvalue** of **S** and $\mathbf{e}_1$ is the associated **eigenvector**

2. $\lambda = \lambda_2$, $\mathbf{p} = \mathbf{e}_2$, where λ_2 is the second biggest **eigenvalue** of **S** and $\mathbf{e}_2$ is the associated **eigenvector**

We choose the **first principal component $\mathbf{p}_1$** to be the eigenvector $\mathbf{e}_1$ with the largest eigenvalue λ_1. λ_1 is the variance of the 1st component scores $\mathbf{t}^{(1)}$.

Step 5: Interpret Finding the eigenvalues and eigenvectors for our example, we arrive at the first principal component being:

$$\mathbf{p}_1 = \begin{pmatrix} 0.75 \\ 0.66 \end{pmatrix} \begin{matrix} \text{Java} \\ \text{Coffee} \end{matrix} \qquad \lambda_1 = s_{t^{(1)}}^2 = \mathbf{p}_1^T\mathbf{S}\mathbf{p}_1 = 16.5$$

The first component score $\mathbf{t}^{(1)}$ is the "Java-coffeeness" of a student.

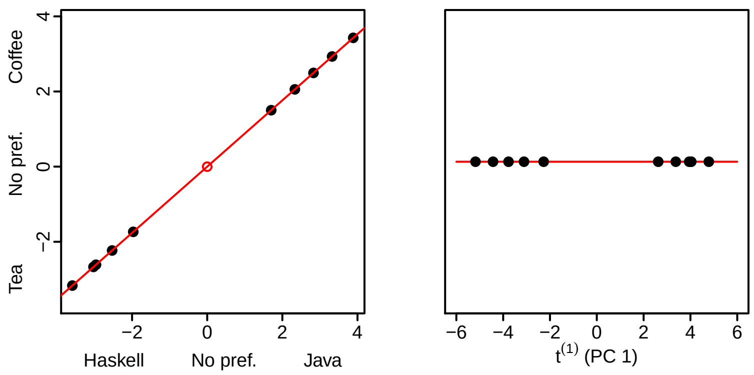


Figure 9: Projection into the original space.

Step 6: Find the 2nd principal component (PC 2) In 2D our job is already done, since there is only one direction perpendicular to $\mathbf{p}_1$, and eigenvectors (and therefore principal components) are always orthogonal to each other. It's the other eigenvector of $\mathbf{S}$, $\mathbf{p}_2 = \mathbf{e}_2$, with eigenvalue λ_2, which is the variance of the 2nd component scores $\mathbf{t}^{(2)}$.

In D dimensions, the principal components are the D eigenvectors of the $D \times D$ matrix $\mathbf{S}$. It's helpful to introduce more matrix notation here. We arrange the principal components in the **rotation matrix**:

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1 & \mathbf{p}_2 \end{pmatrix}$$

Step 7: Quantify what is lost by dimensionality reduction We can reverse the transformation from the scores to the original data

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T$$

If we drop the 2nd PC from $\mathbf{P}$ and the 2nd PC scores from $\mathbf{T}$, we can reconstruct a 1-dimensional version of the original data:

$$\tilde{\mathbf{X}}^{(1)} = \mathbf{t}^{(1)}\mathbf{p}_1^T$$

We can see (Figure 9) that the first principal component (the "Java-coffeeness") score of a student tells us **a lot** about them – but how much? Consider the **total variance**, the sum of the variances of the data:

$$\sum_{i=1}^{D} s_i^2 = \sum_{i=1}^{D} \lambda_i$$

It is equal to the sum of the eigenvalues of the covariance matrix. Thus the fraction of the total variance "explained" by the ith principal component is:

$$\frac{\lambda_i}{\sum_{j=1}^{D} \lambda_j}$$

In our toy example,

$$\frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{16.5}{16.5 + 0.61} = 96\%$$

Thus we can now be more precise about how much the first principal component (the "Java-coffeeness") score of a student tells us about them: 96% of of the variance.

References

Scottish Government (2016). 'Scottish index of multiple deprivation (SIMD) 2016'. URL https://www.webarchive.org.uk/wayback/archive/20200117165925/https://www2.gov.scot/SIMD