

PCA: Theory Notes and Q&A

FOML Test 1 revision. Built from your handwritten class notes (Adobe Scan, 15 pages). The LDA parts come from your LDA notebook. Part A is the theory in order. Part B is the Q&A with answers sized to the marks.

Part A: Theory	Part B: Q&A
1 Why PCA: multicollinearity 2 Hedge fund example 3 Geometric intuition 4 When to use PCA 5 Covariance matrix and total variance 6 Eigenvectors, eigenvalues, unit vector 7 Variance redistribution and dimensionality reduction 8 Rotation maths (trigonometry) 9 Length of projection (linear algebra) 10 PCA on images (eigenfaces)	Q1 to Q10: concept questions (2 and 3 marks) Q11 to Q17: maths and numericals Q18 to Q23: images, eigenfaces, shapes Q24 to Q26: PCA vs LDA, Set B Q1B, choosing components Quick revision sheet at the end

Part A: Theory

1. Why PCA? The multicollinearity problem

PCA is used a lot in business intelligence and business analytics roles such as marketing, HR and fintech. A common interview question is: **what is PCA and why use it?**

The core problem PCA solves is **multicollinearity**. Take a regression:

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Regression assumes the predictors $x_1, x_2, \dots$ are **not correlated with each other**. The usual statistical fix is to drop one of the correlated variables. In industry that often does not work, because every variable carries useful information. So you use PCA.

What PCA does: it creates new variables called **principal components (PCs)**. They are **linear combinations of the original variables** and they are **uncorrelated** with each other. PC1 is one combination of the variables, PC2 is a different combination of the same variables.

$$PC_1 = a_{11}x_1 + a_{21}x_2 + \dots$$
$$y = \alpha + \beta_1 PC_1 + \beta_2 PC_2 + \dots$$

2. Hedge fund example

Take 3 highly correlated stocks, for example HDFC, ICICI and Axis, or Wipro, Infosys and TCS. When news hits, they can move in three different ways:

Movement	What happens	Matches
Parallel shift	All three move in the same direction	PC1
Curvature change	Two move one way and one moves little	PC2
Slope change	Two move in one direction and the third moves in the opposite direction	PC3

Each of these movements is a different **weighted combination of A, B and C**. If PC1 turns out to be dominated by stock C's movement, then stock C has the most influence. Instead of tracking all 3 stocks and

guessing which pattern is dominating, you just see **which PC is active**.

3. Geometric intuition

A company surveys customers on two things: x_1 = comfort and x_2 = look. Plot the ratings and the points form a tilted oval cloud.

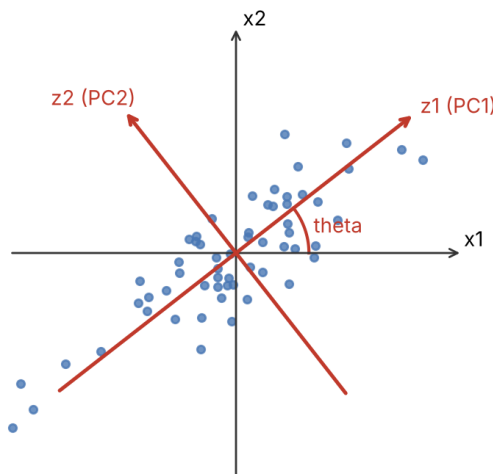


Illustration (made for this PDF, not from your notes): data cloud with the original axes x_1 , x_2 and the rotated axes z_1 (PC1) and z_2 (PC2).

- **Question:** along which single direction are the points most spread out?
- **Answer:** none of the original axes line up with the real spread. So we **rotate the coordinate system**.
- PC1 (z_1) lies along the long direction of the cloud. PC2 (z_2) is perpendicular to it.
- PC1 alone preserves most of the story. It tells the company the dimension along which customer opinions genuinely differ, and that dimension is worth investing R&D into.

4. When should you use PCA?

- Use it when the variables (x_1 , x_2) are **highly correlated**. Your notes use $\text{cov}(x_1, x_2) > 0.80$ as the example.
- Only if they are highly correlated will the scatter plot point in **one dominant direction**.
- Otherwise the plot is a **circular blob**. Rotating the axes gives you nothing useful, because there is no best direction.
- **Hedging example:** hedging means protecting yourself from loss by betting on both sides. Gold and silver are correlated: if gold goes up so does silver. If you do not know the future movement, you buy gold and sell silver. The question hedge funds need answered is: what ratio of gold to silver should I hold to balance this properly? **PCA is used to find that ratio**.

5. Covariance matrix and total variance

Covariance tells you how much multiple variables vary together. For two variables the covariance matrix has the variances on the diagonal and the covariance off the diagonal. The **total variance** (total risk) is the sum of the variances. In your Excel example it came to 24.57.

$$\text{Covariance matrix} = \begin{bmatrix} \text{Var}(x_1) & \text{Cov}(x_1, x_2) \\ \text{Cov}(x_1, x_2) & \text{Var}(x_2) \end{bmatrix}$$

$$\text{Total variance (total risk)} = \text{Var}(x_1) + \text{Var}(x_2)$$

6. Eigenvectors, eigenvalues and the unit vector

A **vector** is a point with a direction and a magnitude. When you multiply a matrix with a vector, it either **rotates** it (changes direction) or **stretches** it (changes length).

So when you multiply the covariance matrix with a random vector, the output is a new vector with a changed direction and length. Keep multiplying the same covariance matrix with the new vectors again and again, and the vectors keep drifting toward the **same general direction**: the direction of maximum variability, the long dominant axis. You stop at a vector where multiplying again does not rotate it any more, it only stretches it. That vector is the **eigenvector**, a line passing through the origin. The value by which it stretches is the **eigenvalue**.

$$A \cdot V = \lambda \cdot V$$

$$A = \text{covariance matrix}, \quad V = \text{eigenvector}, \quad \lambda = \text{eigenvalue}$$

Unit vector: a vector whose length is exactly 1. We need it because the equation $A \cdot V = \lambda \cdot V$ holds true for any scalar multiple of V ($2V$, $3V$, $0.5V$ and so on), so we standardise the vector to length 1.

$$\|v\| = \sqrt{a^2 + b^2}$$

$$u = \frac{v}{\|v\|}$$

$$v = (3, 5): \quad \|v\| = \sqrt{34}, \quad u = \left(\frac{3}{\sqrt{34}}, \frac{5}{\sqrt{34}} \right) = (0.51, 0.86)$$

7. Variance redistribution and dimensionality reduction

Total variability in your system never changes. Rotation only **redistributes** how it is split across your axes.

$$\text{Var}(x_1) + \text{Var}(x_2) = \text{Var}(z_1) + \text{Var}(z_2) = \lambda_1 + \lambda_2$$

- Example from your notes: before rotation the spread was 50% - 50% across x_1 and x_2 . After rotating to align with the eigenvector directions (z_1, z_2) the same variability gets reorganised: z_1 captures **96%** and z_2 captures **4%**.
- **Dimensionality reduction**: throw away z_2 and keep z_1 .
- If you have **100 variables**, you get 100 eigenvalues, and it is possible that just the **top 5 capture 90%** of the variability.

- The objective of the rotation is to get $\text{Var}(z_1) > \text{Var}(z_2)$.

8. Rotation maths using trigonometry

Rotate the axes by an angle θ . A point $M(x_1, x_2)$ now has new coordinates (z_1, z_2) along the new axes. Each new coordinate is the **shadow (projection)** of the old coordinates on the new axis.

- **Project x_1 onto z_1 :** $\cos(\theta) = \text{adjacent} / \text{hypotenuse} = \text{shadow of } x_1 \text{ on } z_1 / x_1$, so the shadow is $x_1 \cos(\theta)$.
- **Project x_2 onto z_1 :** the angle is $90 - \theta$, so $\cos(90 - \theta) = \sin(\theta)$ and the shadow is $x_2 \sin(\theta)$.
- **Add both together:** $z_1 = x_1 \cos(\theta) + x_2 \sin(\theta)$.
- **Project x_1 onto z_2 :** the angle between x_1 and z_2 is $90 + \theta$, so $\cos(90 + \theta) = -\sin(\theta)$. The shadow is $-x_1 \sin(\theta)$. (Your notes cross out the first attempt, $z_2 = -x_1 \sin(\theta)$ from $\sin(\theta) = z_2 / -x_1$, and keep the $\cos(90 + \theta)$ method.)
- **Project x_2 onto z_2 :** the shadow is $x_2 \cos(\theta)$.
- **Add both together:** $z_2 = -x_1 \sin(\theta) + x_2 \cos(\theta)$.

$$\begin{aligned} z_1 &= x_1 \cos \theta + x_2 \sin \theta \\ z_2 &= -x_1 \sin \theta + x_2 \cos \theta \\ x_1^2 + x_2^2 &= z_1^2 + z_2^2 \end{aligned}$$

In matrix form, the matrix with cos and sin entries is made of **unit vectors** (one per new axis):

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$Z = A^T X$$

$$A = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

The columns of A are the unit vectors along z_1 and z_2 . Multiplying A transpose with A gives the **identity matrix**, which shows that the new axes are perpendicular unit vectors (a basis).

$$\begin{aligned}
 A^T A &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \\
 &= \begin{bmatrix} \cos^2 \theta + \sin^2 \theta & 0 \\ 0 & \cos^2 \theta + \sin^2 \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I
 \end{aligned}$$

9. Length of projection (linear algebra view)

Easy case: length of the projection of point (3, 2) on the x-axis. Take the unit vector along that axis, (1, 0), and multiply with the point.

$$u = (1, 0) : \quad [1 \ 0] \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = 3$$

Tilted axis (like z_1 and z_2): project (4, 3) onto the axis pointing along the vector (3, 4). First find the unit vector for that axis, then the length, then the coordinates.

$$\begin{aligned}
 u &= \left(\frac{3}{\sqrt{9+16}}, \frac{4}{\sqrt{9+16}} \right) = (0.6, 0.8) \\
 \text{length} &= [0.6 \ 0.8] \cdot \begin{bmatrix} 4 \\ 3 \end{bmatrix} = 0.6(4) + 0.8(3) = 4.8 \\
 \text{coordinates} &= 4.8 \times (0.6, 0.8) = (2.88, 3.84)
 \end{aligned}$$

Why must it be a unit vector? If you directly multiply [3 4] with the point you get 24, which is 5 times larger. The 5 is the length of (3, 4). The unit vector removes that extra scaling, so the number you get is the true length of the projection.

$$[3 \ 4] \cdot \begin{bmatrix} 4 \\ 3 \end{bmatrix} = 24 = 5 \times 4.8$$

Using (3, 4) directly inflates the answer by its own length $\sqrt{3^2 + 4^2} = 5$

For any angle theta: the unit vector is $u = (\cos \theta, \sin \theta)$, because $\cos \theta = \text{adjacent} / \text{hypotenuse}$ and $\sin \theta = \text{opposite} / \text{hypotenuse}$ with hypotenuse = 1.

$$u = (\cos \theta, \sin \theta)$$

$$\text{length} = u^T x = [\cos \theta \quad \sin \theta] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \cos \theta + x_2 \sin \theta = z_1$$

$$\text{coordinates} = \text{length} \times u$$

Assignment from your notes: take the point (5, 7), pick any angle and find its shadow length both ways. With theta = 60 degrees, trigonometry and linear algebra ($u^T x$) give the same answers:

$$\theta = 60^\circ : u = (\cos 60^\circ, \sin 60^\circ) = (0.5, 0.866)$$

$$z_1 = 5 \cos 60^\circ + 7 \sin 60^\circ = 2.5 + 6.06 = 8.56$$

$$z_2 = -5 \sin 60^\circ + 7 \cos 60^\circ = -4.33 + 3.5 = -0.83$$

PCA only works on centred data. Subtract the mean of each variable first, so that the cloud is centred on the origin before rotating.

10. PCA on images (eigenfaces)

Setting: 8 images, 2 per person (4 people). Each image is flattened into one long column. In your notes a small 5 x 2 face image becomes a 10 x 1 column. Put all the columns side by side to get the data matrix A.

Symbol	Meaning	Notes example
n	number of features (pixels per image)	10 features
m	number of training examples (images)	8 images
A	data matrix, one image per column	(10, 8)

- **Mean image:** average each row across the 8 images, $(a_1 + b_1 + \dots + h_1)/8$ and so on. This gives a (10, 1) average image $(m_1 \dots m_{10})$.
- **Centre the data:** subtract the mean image from every face. The centred matrix is A (10, 8).
- **Covariance matrices:** there are two ways to form it, shown below.

$$C = A^T A \quad (m \times m)$$

$$L = A A^T \quad (n \times n)$$

	$C = A^T A$	$L = A A^T$
Shape (notes example)	(8, 8)	(10, 10)
64 x 64 images, 100 people (A = 4096 x 100)	C = (100, 100), eigenvalues: 100, eigenvectors $V_C = (100, 100)$	L = (4096, 4096), eigenvalues: 4096, eigenvectors $V_L = (4096, 4096)$

The trick: finding eigenvectors of the huge L (4096 x 4096) is expensive. Instead find the eigenvectors of the small C (100 x 100) and convert them. Start from the eigen equation of C and multiply by A on the left:

$$C V_C = \lambda_C V_C$$

$$(A^T A) V_C = \lambda_C V_C$$

multiply by A on the left : $(A A^T)(A V_C) = \lambda_C (A V_C)$

$$L V_L = \lambda_C V_L, \quad V_L = A V_C$$

So $A V_C$ is an eigenvector of L with the **same eigenvalue**. In your notes' small example, $V_L = A V_C$ has shape (10, 8).

Weights (projection of faces onto the eigenfaces):

$$\Omega = V_L^T \cdot a_m \quad (\text{weights of one face})$$

$$\Omega = V_L^T \cdot A \quad (\text{weights of all training faces})$$

In the example the shapes are $(8, 10) \times (10, 1) = (8, 1)$ for one face, and $(8, 10) \times (10, 8) = (8, 8)$ for all training faces.

Another example from your notes: 500 images, each 28×28 (784 pixels). The small covariance matrix is 500×500 , so you get 500 eigenvalues and the eigenvector matrix V has shape (784, 500).

Part B: Questions and Answers

Marks are shown on each question. Write the answer to about this length in the exam: 2 marks gets about 3 to 4 sentences, 3 marks gets the idea, the formula and one example.

Q1. What is PCA and why is it used?

[2 marks]

PCA (Principal Component Analysis) creates new variables called principal components. They are linear combinations of the original variables and are uncorrelated with each other. It is used to handle multicollinearity and to reduce the number of variables while keeping most of the variability.

It is used widely in business analytics roles such as marketing, HR and fintech.

Q2. Which problem does PCA solve? Explain multicollinearity.

[2 marks]

Regression $y = \alpha + \beta_1 x_1 + \dots + \beta_n x_n$ assumes the predictors are not correlated with each other. When they are correlated, that is multicollinearity. The statistical fix is to drop one correlated variable, but in industry that does not work because each variable carries information. PCA replaces them with uncorrelated principal components, so the regression becomes $y = \alpha + \beta_1 PC_1 + \beta_2 PC_2 + \dots$

Q3. Explain PCA using the hedge fund example.

[3 marks]

Take 3 highly correlated stocks such as HDFC, ICICI and Axis. When news hits they can move in three ways: a **parallel shift** (all move in the same direction), a **curvature change** (two move one way and one moves little) and a **slope change** (two move one way and the third moves opposite).

These three movements correspond to PC1, PC2 and PC3. Each one is a different weighted combination of the stocks A, B and C. If PC1 is dominated by stock C, then stock C has the most influence. So instead of tracking all 3 stocks and guessing which pattern dominates, you only check which PC is active.

Q4. Explain the geometric intuition of PCA.

[2 marks]

Take a survey where customers rate comfort (x_1) and look (x_2). The points form a tilted oval. None of the original axes line up with the real spread, so we rotate the coordinate system. PC1 (z_1) is along the direction of maximum spread and PC2 (z_2) is perpendicular to it. PC1 alone preserves most of the story and tells the company the dimension along which opinions genuinely differ, which is worth investing R&D into.

Q5. When should PCA be used and when does it not help?

[2 marks]

PCA should be used when the variables are highly correlated (the notes use $\text{cov}(x_1, x_2) > 0.80$), because then the scatter plot points in one dominant direction. If they are not correlated, the plot is a circular blob. Rotating the axes gives nothing useful, since there is no best direction, so PCA has no benefit.

Q6. How is PCA used in hedging? (gold and silver)

[2 marks]

Hedging means protecting yourself from loss by betting on both sides. Gold and silver are correlated: if gold goes up so does silver. If you do not know future movement, you buy gold and sell silver. Hedge funds need to know what ratio of gold to silver to hold to balance this properly. PCA is used to find that ratio.

Q7. What is the covariance matrix and what is total variance?

[2 marks]

Covariance shows how much multiple variables vary together. The covariance matrix has $\text{Var}(x_1)$ and $\text{Var}(x_2)$ on the diagonal and $\text{Cov}(x_1, x_2)$ in the other two places. Total variance (total risk) = $\text{Var}(x_1) + \text{Var}(x_2)$. In the Excel example in the notes it is 24.57.

$$\text{Covariance matrix} = \begin{bmatrix} \text{Var}(x_1) & \text{Cov}(x_1, x_2) \\ \text{Cov}(x_1, x_2) & \text{Var}(x_2) \end{bmatrix}$$

$$\text{Total variance (total risk)} = \text{Var}(x_1) + \text{Var}(x_2)$$

Q8. What are eigenvectors and eigenvalues? Explain the intuition.

[3 marks]

Multiplying a matrix with a vector rotates it or stretches it. If you multiply the covariance matrix with a vector and keep repeating with the new vectors, they keep drifting toward the direction of maximum variability, the long dominant axis. You reach a vector that is no longer rotated, only stretched. That vector is the **eigenvector** (a line through the origin) and the stretching factor is the **eigenvalue**:

$$A \cdot V = \lambda \cdot V$$

A = covariance matrix, V = eigenvector, λ = eigenvalue

In PCA the eigenvectors of the covariance matrix are the principal component directions, and each eigenvalue is the variance along its direction.

Q9. What is a unit vector? Why is it needed and how do you find it?

[2 marks]

A unit vector has length exactly 1. It is needed because $A \cdot V = \lambda \cdot V$ holds for any scalar multiple of V (2V, 3V, 0.5V), so we standardise the vector. To find it, divide the vector by its length, where length = $\sqrt{a^2 + b^2}$. Example: for (3, 5) the length is $\sqrt{34}$ and the unit vector is $(3/\sqrt{34}, 5/\sqrt{34}) = (0.51, 0.86)$.

Q10. Show that rotation does not change total variance. How does this lead to dimensionality reduction?

[3 marks]

Total variability in the system never changes. Rotation only redistributes how it is split across the axes:

$$\text{Var}(x_1) + \text{Var}(x_2) = \text{Var}(z_1) + \text{Var}(z_2) = \lambda_1 + \lambda_2$$

For example, before rotation the spread is 50% and 50% across x_1 and x_2 . After rotating to align with the eigenvector directions, z_1 captures 96% and z_2 captures 4%. So we throw away z_2 and keep z_1 , which is dimensionality reduction. With 100 variables you get 100 eigenvalues, and often just the top 5 capture 90% of the variability.

Q11. Derive z_1 and z_2 by projecting x_1 and x_2 on the rotated axes using trigonometry.

[3 marks]

Rotate the axes by angle theta. The new coordinates are the shadows of x_1 and x_2 on the new axes.

- On z_1 : shadow of $x_1 = x_1 \cos(\theta)$ ($\cos \theta = \text{adjacent/hypotenuse}$). Shadow of $x_2 = x_2 \sin(\theta)$ (angle is $90 - \theta$).
- On z_2 : shadow of $x_1 = -x_1 \sin(\theta)$ (angle is $90 + \theta$, $\cos(90 + \theta) = -\sin \theta$). Shadow of $x_2 = x_2 \cos(\theta)$.

$$z_1 = x_1 \cos \theta + x_2 \sin \theta$$

$$z_2 = -x_1 \sin \theta + x_2 \cos \theta$$

$$x_1^2 + x_2^2 = z_1^2 + z_2^2$$

Q12. Write the rotation in matrix form and show that $A^T A = I$.

[2 marks]

In matrix form $Z = A^T X$, where the columns of A are the unit vectors along z_1 and z_2 .

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$Z = A^T X$$

$$\begin{aligned} A^T A &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \\ &= \begin{bmatrix} \cos^2 \theta + \sin^2 \theta & 0 \\ 0 & \cos^2 \theta + \sin^2 \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \end{aligned}$$

$A^T A = I$ shows the columns are unit length and perpendicular to each other, so the new axes form a valid (orthonormal) set of axes.

Q13. How do you find the length of the projection of a point on an axis? Find the projection of (4, 3) on the axis along (3, 4).

[3 marks]

Take the unit vector u along the axis. Length of projection = $u^T x$. To get the coordinates, multiply the length by the unit vector.

$$u = \left(\frac{3}{\sqrt{9+16}}, \frac{4}{\sqrt{9+16}} \right) = (0.6, 0.8)$$

$$\text{length} = [0.6 \ 0.8] \cdot \begin{bmatrix} 4 \\ 3 \end{bmatrix} = 0.6(4) + 0.8(3) = 4.8$$

$$\text{coordinates} = 4.8 \times (0.6, 0.8) = (2.88, 3.84)$$

Q14. Why must the axis vector be a unit vector?**[2 marks]**

If you multiply with (3, 4) directly you get 24, which is 5 times larger than the true length 4.8. The 5 is the length of (3, 4). A unit vector has length 1, so it does not inflate the answer and the number you get is the true length of the shadow.

$$\begin{bmatrix} 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 3 \end{bmatrix} = 24 = 5 \times 4.8$$

Using (3, 4) directly inflates the answer by its own length $\sqrt{3^2 + 4^2} = 5$

Q15. Numerical: take the point (5, 7), pick theta = 60 degrees, and find z_1 and z_2 .**[2 marks]**

$$\theta = 60^\circ : u = (\cos 60^\circ, \sin 60^\circ) = (0.5, 0.866)$$

$$z_1 = 5\cos 60^\circ + 7\sin 60^\circ = 2.5 + 6.06 = 8.56$$

$$z_2 = -5\sin 60^\circ + 7\cos 60^\circ = -4.33 + 3.5 = -0.83$$

Both methods (trigonometry and linear algebra with $u = (\cos 60, \sin 60) = (0.5, 0.866)$) give the same result: $z_1 = 8.56$ and $z_2 = -0.83$.

Q16. Why does PCA only work on centred data?**[1 mark]**

The rotation is about the origin. Subtracting the mean of each variable places the centre of the cloud at the origin, so the new axes pass through the centre of the data and the variance is measured correctly.

Q17. What does the projection of (3, 2) on the x-axis give?**[2 marks]**

The unit vector along the x-axis is (1, 0). Multiplying [1 0] with (3, 2) gives 3, which is the length of the projection (the shadow on the x-axis).

Q18. Explain how PCA is applied to a set of face images (eigenfaces).**[3 marks]**

- **Flatten** each image into one column (a 5 x 2 image becomes 10 x 1). Stack all images as columns to form A, with n features and m images.
- **Mean image**: average each row over all m images. **Subtract** it from every image to centre the data.
- Form the covariance matrix. Find its eigenvalues and eigenvectors. The eigenvectors are the eigenfaces.
- **Project** each face on the eigenfaces to get its weights: $\Omega = V_L^T a_m$.
- Keep only the top eigenfaces that hold enough of the variance (for example 90%).

$$\Omega = V_L^T \cdot a_m \quad (\text{weights of one face})$$

$$\Omega = V_L^T \cdot A \quad (\text{weights of all training faces})$$

Q19. Why do we use $C = A^T A$ instead of $L = A A^T$ for images? Prove the relation between their eigenvectors.

[3 marks]

For a 64 x 64 image of 100 people, A is 4096 x 100. $L = A A^T$ is 4096 x 4096 with 4096 eigenvalues, which is very expensive. $C = A^T A$ is only 100 x 100 with 100 eigenvalues. So we find the eigenvectors V_C of C and convert them: multiply the eigen equation of C by A on the left.

$$C V_C = \lambda_C V_C$$

$$(A^T A) V_C = \lambda_C V_C$$

multiply by A on the left : $(A A^T)(A V_C) = \lambda_C (A V_C)$

$$L V_L = \lambda_C V_L, \quad V_L = A V_C$$

So $V_L = A V_C$ is an eigenvector of L with the same eigenvalue.

Q20. Give the shapes in the example with 8 images of 10 pixels each.

[2 marks]

Item	Shape
A (centred data)	(10, 8)
$C = A^T A$	(8, 8)
$L = A A^T$	(10, 10)
$V_L = A V_C$	(10, 8)
Omega for one face ($V_L^T a_m$)	(8, 1)
Omega for all training faces	(8, 8)

Q21. Shapes for 64 x 64 images of 100 people.

[2 marks]

Item	Shape
A	(4096, 100)
$C = A^T A$	(100, 100), 100 eigenvalues, $V_C = (100, 100)$
$L = A A^T$	(4096, 4096), 4096 eigenvalues, $V_L = (4096, 4096)$

Q22. 500 images, each 28 x 28. What are the shapes of the eigenvalue and eigenvector matrices (small covariance method)?

[2 marks]

Each image has $28 \times 28 = 784$ pixels. The small covariance matrix is 500 x 500, so the eigenvalues have shape (500, 500) and the eigenvector matrix V has shape (784, 500).

Q23. What is the mean image and why do we subtract it?

[2 marks]

The mean image is the average of all training images, taken pixel by pixel: $(a_1 + b_1 + \dots + h_1)/8$ for the first pixel and so on, giving one (10, 1) average face. Subtracting it from every face centres the data, which PCA needs, so the eigenfaces capture how each face differs from the average face.

Q24. Compare PCA and LDA.**[3 marks]**

	PCA	LDA
Type	Unsupervised, ignores labels	Supervised, uses class labels
Goal	Direction of maximum variance	Direction that best separates classes
Matrix used	Covariance matrix	$S_W^{-1} S_B$
Eigenvectors give	Principal components	Discriminant direction
Max components	Up to number of features	Number of classes minus 1

Q25. Set B Q1B: Analyse how eigenvectors and eigenvalues are used in PCA and LDA.**[2 marks]**

PCA: the eigenvectors of the covariance matrix are the principal component directions (the new rotated axes). The eigenvalue of each eigenvector is the variance along it, so the largest eigenvalue gives PC1. Variance explained = $\lambda_k / \text{sum of all } \lambda$. We keep the top eigenvectors and drop the rest for dimensionality reduction.

LDA: the eigenvectors of $S_W^{-1} S_B$ are the discriminant directions. The eigenvalue measures how well the classes are separated along that direction. The eigenvector with the largest eigenvalue is the optimal discriminant direction. In your LDA notebook, for 2 classes only one eigenvalue was non-zero (15.657, the other was about 0), and its eigenvector [0.9196, 0.3930] was the direction used to project the data.

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$S_W^{-1} S_B v = \lambda v$$

Q26. How do you decide how many principal components to keep?**[2 marks]**

Compute the variance explained by each component, λ_k divided by the sum of all eigenvalues, and keep the smallest number of components whose cumulative variance reaches the target. For example 90%. With 100 variables, often the top 5 capture 90%.

$$\text{variance explained by PC}_k = \frac{\lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_n}$$

$$\text{keep the smallest } k \text{ where } \frac{\lambda_1 + \dots + \lambda_k}{\lambda_1 + \dots + \lambda_n} \geq 0.90$$

In code, `PCA(0.90)` keeps just enough components to preserve 90% of the variance, the same way `PCA(0.95)` and `PCA(0.94)` are used in your notebooks.

Quick revision sheet

Idea	One line
PCA	Rotates axes to the direction of maximum variance. New variables are uncorrelated linear combinations.
When	Variables highly correlated (cov above 0.80 in your notes). Circular blob means PCA is useless.
Eigen equation	$A.V = \lambda.V$. V stays on its line and is only stretched by λ .
Unit vector	Length 1. $u = v / \text{length}$. Needed because any multiple of V satisfies the equation.
Variance	Total variance is unchanged by rotation. $\text{Var}(x_1) + \text{Var}(x_2) = \text{Var}(z_1) + \text{Var}(z_2) = \lambda_1 + \lambda_2$.
Rotation	$z_1 = x_1 \cos(\theta) + x_2 \sin(\theta)$, $z_2 = -x_1 \sin(\theta) + x_2 \cos(\theta)$. $Z = A^T X$, $A^T A = I$.
Projection	Length = $u^T x$. Coordinates = length times u .
Images	Flatten, subtract mean image, use small $C = A^T A$ ($m \times m$), then $V_L = A.V_C$. Weights $\Omega = V_L^T a$.
Centering	PCA only works on centred data.

Check against your notes: page 21 of your scan writes the unit vector of (3, 5) as (0.51, 0.85). The exact value of $5/\sqrt{34}$ is 0.857, which rounds to 0.86, so this PDF uses 0.86.