# Linear Algebra for Machine Learning

A Complete, Step-by-Step Treatment for the Quant / ML Researcher

Every result derived; every step justified; worked examples throughout

**Abstract**

This document develops the linear algebra a machine-learning researcher and quantitative analyst actually uses, from first principles. The philosophy is relentless: *nothing is asserted without explanation.* Every theorem is proved or carefully justified, every definition is motivated by the problem it solves, and almost every concept is accompanied by a fully worked numerical example and an explicit connection to a machine-learning or finance application. The aim is that after reading you should not merely *know* that $\mathbf{X}^\top\mathbf{X}$ appears in least squares — you should understand, at the level of being able to re-derive it on a whiteboard, exactly *why* every symbol is where it is.

## Contents

**1 Orientation: Why Linear Algebra Is the Language of ML** **4**

**2 Vectors, Vector Spaces, and Subspaces** **4**

2.1 What a vector space is, and why the axioms are what they are . . . . . 4

2.2 Linear combinations, span, and why span is always a subspace . . . . . 5

2.3 Linear independence: the precise meaning of "no redundancy" . . . . . 5

2.4 Basis and dimension, and why dimension is well-defined . . . . . 6

**3 Matrices as Linear Maps** **6**

3.1 The definition that makes everything else inevitable . . . . . 6

3.2 Matrix multiplication is composition — and that explains its rule . . . . . 7

3.3 Transpose and the adjoint identity . . . . . 8

3.4 The trace and its cyclic property . . . . . 8

**4 The Four Fundamental Subspaces** **8**

4.1 Column space and null space, and proofs that they are subspaces . . . . . 8

4.2 Rank and the Rank–Nullity Theorem . . . . . 9

4.3 Orthogonal complements and the Fundamental Theorem . . . . . 9

**5 Solving Linear Systems and the Geometry of Least Squares** **10**

5.1 When does $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ have a solution, and is it unique? . . . . . 10

5.2 The least-squares problem and why we project . . . . . 10

5.3 The hat matrix as a projector . . . . . 11

**6 Norms, Inner Products, and Angles** **11**
6.1 Norms: measuring size, and why different norms matter . . . 11
6.2 Inner products, Cauchy–Schwarz, and the meaning of angle . . . 12
6.3 Orthonormal bases and orthogonal matrices . . . 12

**7 Eigenvalues and Eigenvectors** **13**
7.1 Definition and the characteristic equation . . . 13
7.2 Diagonalization: changing to the eigenbasis . . . 13
7.3 Trace and determinant via eigenvalues . . . 14
7.4 The Spectral Theorem — the result you use most . . . 14

**8 Positive Definiteness and Quadratic Forms** **14**
8.1 Quadratic forms and what their sign means . . . 14
8.2 Why this concept is everywhere . . . 15
8.3 Tests for positive definiteness . . . 15

**9 The Singular Value Decomposition** **15**
9.1 Statement and derivation . . . 15
9.2 SVD and the four subspaces . . . 16
9.3 Eckart–Young: best low-rank approximation . . . 16
9.4 The pseudoinverse and minimum-norm least squares . . . 17

**10 Condition Number and Numerical Stability** **17**

**11 Essential Decompositions in Practice** **18**

**12 Matrix Calculus: the Identities and Their Derivations** **18**
12.1 The core gradient identities, derived . . . 18
12.2 The least-squares gradient, in one line . . . 18
12.3 Matrix-derivative identities for likelihoods . . . 19

**13 Capstone: Tying It All to Machine Learning** **19**

**14 Gram–Schmidt and the Construction of QR** **19**
14.1 The Gram–Schmidt process . . . 20
14.2 Why this is QR . . . 20

**15 Determinants, Properly Understood** **20**
15.1 The determinant as signed volume . . . 20

**16 Change of Basis: the Same Object in Different Coordinates** **21**

**17 Worked Computations: Eigendecomposition and SVD by Hand** **21**
17.1 A complete eigendecomposition . . . 21

17.2 A complete SVD . . . 22
17.3 Power iteration: how eigenvectors are actually found . . . 23

**18 Least Squares, Worked Three Ways** **24**
18.1 The problem . . . 24
18.2 Route 1 — the normal equations . . . 24
18.3 Route 2 — the projection picture . . . 24
18.4 Route 3 — QR (sketch) . . . 24

**19 Worked Gram–Schmidt and the QR Factorization** **25**
19.1 The process on two vectors . . . 25
19.2 Reading off $\mathbf{R}$ . . . 25

**20 Positive Definiteness, Worked and Tested** **25**
20.1 Three equivalent tests . . . 26
20.2 A semidefinite and an indefinite case . . . 26

**21 Matrix Calculus: the Identities You Will Reuse Forever** **26**
21.1 The core gradients, derived . . . 26
21.2 The least-squares gradient in one line . . . 27
21.3 Identities for likelihoods and the log-determinant . . . 27

**22 Vectorization and the Kronecker Product** **27**

**23 Determinants as Volume, Worked** **28**
23.1 The $2 \times 2$ case . . . 28
23.2 Why the multiplicative property is obvious geometrically . . . 28

**24 Change of Basis, Worked** **28**
24.1 A vector in two bases . . . 28
24.2 An operator in the eigenbasis . . . 29

**25 Additional Worked Exercises** **29**

**26 Consolidated Exercises** **30**

# 1 Orientation: Why Linear Algebra Is the Language of ML

Before any formalism, it is worth being concrete about *why* this subject is unavoidable. Almost every object in machine learning is a vector, a matrix, or a function of them.

- A single data point with $p$ measured features (a stock's return, volatility, volume, P/E ratio, ...) is a vector $\mathbf{x} \in \mathbb{R}^p$. The entire dataset of $n$ such points is a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, one row per observation.
- A linear model's prediction $\hat{y} = \mathbf{x}^\top \boldsymbol{\beta}$ is an *inner product*. Fitting the model means solving a system involving the matrix $\mathbf{X}^\top \mathbf{X}$.
- Principal component analysis, the covariance structure of a portfolio, the stability of an optimizer, the capacity of a model — all are governed by the *eigenvalues* and *singular values* of matrices.
- Neural networks are compositions of affine maps (matrix multiplications plus a bias) interleaved with nonlinearities; their training is the differentiation of these compositions.

The recurring theme is that **geometry and computation are two views of the same object**. A matrix is simultaneously (i) a rectangular array of numbers you can compute with, and (ii) a linear transformation that stretches, rotates, and projects space. Fluency means being able to switch between these views at will: when an algorithm is numerically unstable, you reach for the geometric picture (near-collinear directions); when a geometric statement needs to be verified, you reach for the algebra.

**Intuition**

Keep one mental image throughout: a matrix $\mathbf{A}$ takes the unit sphere and squashes/stretches it into an ellipsoid. The directions of the ellipsoid's axes and their lengths *are* the eigen/singular structure. Every theorem about conditioning, stability, PCA, and least squares is, at bottom, a statement about that ellipsoid.

# 2 Vectors, Vector Spaces, and Subspaces

## 2.1 What a vector space is, and why the axioms are what they are

**Definition 1** (Vector space)**.** A real vector space is a set $V$ together with two operations — vector addition $+ : V \times V \to V$ and scalar multiplication $\cdot : \mathbb{R} \times V \to V$ — satisfying, for all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and $a, b \in \mathbb{R}$:

(i) commutativity $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$;

(ii) associativity $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$;

(iii) an additive identity $\mathbf{0}$ with $\mathbf{v} + \mathbf{0} = \mathbf{v}$;

(iv) additive inverses $-\mathbf{v}$ with $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$;

(v) $1 \cdot \mathbf{v} = \mathbf{v}$;

(vi) $a(b\mathbf{v}) = (ab)\mathbf{v}$;

(vii) $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$;

(viii) $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$.

Why insist on these eight rules rather than just working with arrays of numbers? Because they are *exactly* the properties that make linear reasoning valid, and they hold for many objects that are not literally arrays — spaces of functions, of random variables, of polynomials. By proving things from the axioms, every theorem applies to all such spaces at once. For instance, the set of all square-integrable random variables (those with finite variance) forms a vector space, and the entire geometric machinery below — projections, orthogonality, "angles" — transfers to give us conditional expectation and the bias–variance decomposition. That transfer is not a metaphor; it is the same theorem applied to a different vector space.

**Intuition**

The axioms say: "you can add things and scale them, and these operations don't surprise you." Distributivity (vii)–(viii) is the load-bearing pair — it is what lets a linear map be determined entirely by its action on a basis, which is the single most useful fact in the subject.

## 2.2 Linear combinations, span, and why span is always a subspace

Given vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in V$ and scalars $c_1, \dots, c_k \in \mathbb{R}$, the vector $\sum_{i=1}^{k} c_i \mathbf{v}_i$ is a **linear combination**. The set of *all* linear combinations is the **span**:

$$\operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_k\} = \Big\{ \textstyle\sum_{i=1}^{k} c_i \mathbf{v}_i : c_i \in \mathbb{R} \Big\}.$$

**Proposition 1.** $\operatorname{span}\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is a subspace — it is closed under addition and scalar multiplication and contains $\mathbf{0}$.

*Proof.* Take two elements $\mathbf{x} = \sum_i a_i \mathbf{v}_i$ and $\mathbf{y} = \sum_i b_i \mathbf{v}_i$ of the span. Their sum is $\mathbf{x} + \mathbf{y} = \sum_i (a_i + b_i)\mathbf{v}_i$, again a linear combination of the same vectors, hence in the span. For a scalar $t$, $t\mathbf{x} = \sum_i (t a_i)\mathbf{v}_i$ is in the span. Finally $\mathbf{0} = \sum_i 0 \cdot \mathbf{v}_i$ is the combination with all coefficients zero. All three closure conditions hold. □

This little proof is the prototype for almost every "… is a subspace" argument: write two general elements, add and scale, observe you stay inside. We will reuse it for column spaces, null spaces, and eigenspaces.

**Intuition**

The span of a set of vectors is the smallest flat (through the origin) containing them. One nonzero vector spans a line; two non-parallel vectors span a plane; in general the span is the set of all destinations reachable by combining the given "directions." In ML, the span of your feature columns is precisely the set of predictions a linear model can possibly produce — a fact we will lean on heavily in least squares.

## 2.3 Linear independence: the precise meaning of "no redundancy"

**Definition 2** (Linear independence)**.** Vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ are linearly independent if the only solution to

$$c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \cdots + c_k \mathbf{v}_k = \mathbf{0}$$

is $c_1 = c_2 = \cdots = c_k = 0$. If any nontrivial solution exists, they are linearly *dependent*.

Read the definition operationally. Dependence means *some* vector can be written in terms of the others: if $\sum_i c_i \mathbf{v}_i = \mathbf{0}$ with, say, $c_1 \neq 0$, then $\mathbf{v}_1 = -\frac{1}{c_1} \sum_{i \geq 2} c_i \mathbf{v}_i$. So a dependent set carries redundant information — one direction is already expressible from the rest. Independence is the formalization of "each vector adds something genuinely new."

**Worked Example**

Consider $\mathbf{v}_1 = (1, 2, 3)$, $\mathbf{v}_2 = (2, 4, 6)$, $\mathbf{v}_3 = (1, 0, 1)$ in $\mathbb{R}^3$. Are they independent? Set $c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 = \mathbf{0}$. Immediately $\mathbf{v}_2 = 2\mathbf{v}_1$, so $2\mathbf{v}_1 - \mathbf{v}_2 = \mathbf{0}$ is a nontrivial relation (take $c_1 = 2, c_2 = -1, c_3 = 0$). They are *dependent*. The redundancy is visible: $\mathbf{v}_2$ points in the same direction as $\mathbf{v}_1$, just twice as far. Geometrically, all three vectors actually lie... well, $\mathbf{v}_1$ and $\mathbf{v}_2$ lie on one line, and $\mathbf{v}_3$ is off it, so the span is only a 2D plane, not all of $\mathbb{R}^3$.

**Worked Example**

In a quant setting, suppose three "factors" are: market return $\mathbf{v}_1$, a sector return $\mathbf{v}_2$, and a third series $\mathbf{v}_3 = 0.5\mathbf{v}_1 + 0.5\mathbf{v}_2$ constructed (perhaps unknowingly) as their average. Then $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is linearly dependent. If you regress returns on all three, the design matrix loses rank and the regression coefficients become non-unique — the textbook cause of "my factor loadings exploded." Detecting this is exactly detecting linear dependence among feature columns.

### 2.4 Basis and dimension, and why dimension is well-defined

**Definition 3** (Basis)**.** A basis of $V$ is a linearly independent set that spans $V$. Equivalently, every vector in $V$ has a *unique* representation as a linear combination of basis vectors.

The uniqueness is worth proving because it is the reason coordinates make sense. Suppose $\mathbf{x} = \sum_i a_i\mathbf{v}_i = \sum_i b_i\mathbf{v}_i$ in a basis. Subtracting, $\sum_i (a_i - b_i)\mathbf{v}_i = \mathbf{0}$, and independence forces $a_i = b_i$ for all $i$. So the coefficients — the *coordinates* of $\mathbf{x}$ in this basis — are unique. This is precisely why we can represent any vector by its list of numbers without ambiguity once a basis is fixed.

**Theorem 1** (Dimension is well-defined)**.** *Any two bases of a finite-dimensional vector space have the same number of elements. That number is the* ***dimension*** $\dim V$.

*Proof sketch via the Steinitz exchange lemma.* The key lemma: if $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ spans $V$ and $\{\mathbf{w}_1, \ldots, \mathbf{w}_n\}$ is independent, then $n \le m$ (an independent set can never be larger than a spanning set). The lemma is proved by "exchanging" one $\mathbf{u}$ for each $\mathbf{w}$, one at a time, keeping a spanning set throughout; if we ever ran out of $\mathbf{u}$'s we would contradict independence of the $\mathbf{w}$'s. Now if $B_1$ and $B_2$ are both bases, each is independent and each spans, so applying the lemma both ways gives $|B_1| \le |B_2|$ and $|B_2| \le |B_1|$, hence equality. □

**Intuition**

Dimension counts genuine degrees of freedom. The reason it is well-defined — that you cannot "cheat" and describe a plane with a basis of three vectors — is what makes statements like "this dataset really lives on a 3-dimensional manifold" meaningful, and what PCA exploits when it says the data's variance is concentrated in $k \ll p$ directions.

## 3 Matrices as Linear Maps

### 3.1 The definition that makes everything else inevitable

**Definition 4** (Linear map)**.** A function $T : \mathbb{R}^n \to \mathbb{R}^m$ is linear if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $a \in \mathbb{R}$:

$$T(\mathbf{x} + \mathbf{y}) = T(\mathbf{x}) + T(\mathbf{y}) \qquad \text{and} \qquad T(a\mathbf{x}) = a\,T(\mathbf{x}).$$

These two conditions (additivity and homogeneity) combine into the single statement $T(a\mathbf{x} + b\mathbf{y}) = aT(\mathbf{x}) + bT(\mathbf{y})$: a linear map commutes with linear combinations. This is the entire content of "linear," and it has an enormous consequence.

**Theorem 2** (A linear map is determined by its action on a basis)**.** *Let* $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ *be a basis of* $\mathbb{R}^n$*. If we know* $T(\mathbf{e}_1), \dots, T(\mathbf{e}_n)$*, then* $T$ *is completely determined.*

*Proof.* Any $\mathbf{x}$ has a unique expansion $\mathbf{x} = \sum_j x_j \mathbf{e}_j$. By linearity,

$$T(\mathbf{x}) = T\Big(\sum_j x_j \mathbf{e}_j\Big) = \sum_j x_j \, T(\mathbf{e}_j).$$

So $T(\mathbf{x})$ is computed from the known vectors $T(\mathbf{e}_j)$ and the coordinates $x_j$. □

Now specialize to the standard basis $\mathbf{e}_j = (0, \dots, 1, \dots, 0)$ (a 1 in slot $j$). Stack the images $T(\mathbf{e}_j) \in \mathbb{R}^m$ as the columns of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. Then the formula above reads

$$T(\mathbf{x}) = \sum_{j=1}^{n} x_j \, (j\text{-th column of } \mathbf{A}) = \mathbf{A}\mathbf{x}.$$

**This is where the matrix–vector product comes from.** It is not an arbitrary rule; it is forced on us the moment we decide to represent a linear map by what it does to basis vectors. The product $\mathbf{Ax}$ is, by construction, "the linear combination of $\mathbf{A}$'s columns weighted by the entries of $\mathbf{x}$."

**Intuition**

There are two ways to read $\mathbf{Ax}$, and you should hold both. **Column picture**: $\mathbf{Ax} = \sum_j x_j \, \mathbf{a}_j$ is a weighted sum of columns — the output lives in the span of the columns. **Row picture**: the $i$-th entry of $\mathbf{Ax}$ is the inner product of row $i$ with $\mathbf{x}$ — each output coordinate "measures" $\mathbf{x}$ against one row. The column picture explains *which outputs are reachable* (least squares); the row picture explains *what each output coordinate means* (each row is a linear functional / a feature detector).

## 3.2 Matrix multiplication is composition — and that explains its rule

Why is matrix multiplication defined by that strange "row times column" rule, and why is it not commutative? Because matrices represent linear maps, and *multiplication must represent composition*. If $\mathbf{B} : \mathbb{R}^n \to \mathbb{R}^k$ and $\mathbf{A} : \mathbb{R}^k \to \mathbb{R}^m$, then doing $\mathbf{B}$ first and $\mathbf{A}$ second is the map $\mathbf{x} \mapsto \mathbf{A}(\mathbf{Bx})$. We want a matrix $\mathbf{C}$ with $\mathbf{Cx} = \mathbf{A}(\mathbf{Bx})$ for all $\mathbf{x}$.

Compute the $j$-th column of $\mathbf{C}$: it is $\mathbf{Ce}_j = \mathbf{A}(\mathbf{Be}_j) = \mathbf{Ab}_j$ where $\mathbf{b}_j$ is the $j$-th column of $\mathbf{B}$. And $\mathbf{Ab}_j = \sum_\ell (\mathbf{b}_j)_\ell \, \mathbf{a}_\ell$. Writing out the $(i, j)$ entry:

$$C_{ij} = \sum_{\ell=1}^{k} A_{i\ell} B_{\ell j}.$$

That is exactly the familiar formula — *derived*, not posited. And non-commutativity is now obvious: doing $\mathbf{A}$ then $\mathbf{B}$ is generally a different map from doing $\mathbf{B}$ then $\mathbf{A}$ (rotate-then-project $\neq$ project-then-rotate), so $\mathbf{AB} \neq \mathbf{BA}$ in general.

**Worked Example**

Let $\mathbf{A} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ (rotation by 90°) and $\mathbf{B} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$ (projection onto the $x$-axis). Take $\mathbf{x} = (0, 1)$.

- $\mathbf{ABx}$: first project $(0, 1) \mapsto (0, 0)$, then rotate $\mapsto (0, 0)$.
- $\mathbf{BAx}$: first rotate $(0, 1) \mapsto (-1, 0)$, then project $\mapsto (-1, 0)$.

Different answers from the same inputs — a concrete witness that $\mathbf{AB} \neq \mathbf{BA}$. The geometric reason (projection destroys the information rotation would have used) is more memorable than the algebra.

### 3.3 Transpose and the adjoint identity

The transpose $\mathbf{A}^\top$ has $(\mathbf{A}^\top)_{ij} = A_{ji}$. Mechanically it flips the array, but its meaning is the **adjoint relationship**:

$$\langle \mathbf{Ax}, \mathbf{y} \rangle = (\mathbf{Ax})^\top \mathbf{y} = \mathbf{x}^\top \mathbf{A}^\top \mathbf{y} = \langle \mathbf{x}, \mathbf{A}^\top \mathbf{y} \rangle.$$

This identity — "you may move $\mathbf{A}$ to the other side of an inner product by transposing it" — is used constantly: it is how the normal equations arise, how backpropagation moves gradients backward through a linear layer (the backward pass of $\mathbf{A}$ is multiplication by $\mathbf{A}^\top$), and how we derive that $\mathbf{X}^\top\mathbf{X}$ is symmetric. The algebraic rules $(\mathbf{AB})^\top = \mathbf{B}^\top\mathbf{A}^\top$ and $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ both have the same flavor: reversing order, because composition is being undone from the outside in (to take off your socks and shoes, reverse the order you put them on).

### 3.4 The trace and its cyclic property

The trace $\operatorname{Tr}(\mathbf{A}) = \sum_i A_{ii}$ sums the diagonal. Its one indispensable property is cyclicity: $\operatorname{Tr}(\mathbf{AB}) = \operatorname{Tr}(\mathbf{BA})$ whenever both products are defined. Proof:

$$\operatorname{Tr}(\mathbf{AB}) = \sum_i (\mathbf{AB})_{ii} = \sum_i \sum_j A_{ij} B_{ji} = \sum_j \sum_i B_{ji} A_{ij} = \sum_j (\mathbf{BA})_{jj} = \operatorname{Tr}(\mathbf{BA}).$$

Just a reordering of a double sum. Cyclicity extends to $\operatorname{Tr}(\mathbf{ABC}) = \operatorname{Tr}(\mathbf{CAB})$ and is the workhorse behind "trace tricks" in matrix calculus (e.g. differentiating $\log\det$, computing expected quadratic forms $\mathbb{E}[\mathbf{x}^\top\mathbf{Ax}] = \operatorname{Tr}(\mathbf{A}\operatorname{Cov}(\mathbf{x})) + \boldsymbol{\mu}^\top\mathbf{A}\boldsymbol{\mu}$, which appears all over risk and variance computations).

**Worked Example**

Why is $\mathbb{E}[\mathbf{x}^\top\mathbf{Ax}] = \operatorname{Tr}(\mathbf{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\top\mathbf{A}\boldsymbol{\mu}$ for $\mathbf{x}$ with mean $\boldsymbol{\mu}$, covariance $\boldsymbol{\Sigma}$? The scalar $\mathbf{x}^\top\mathbf{Ax}$ equals its own trace, so $\mathbb{E}[\mathbf{x}^\top\mathbf{Ax}] = \mathbb{E}[\operatorname{Tr}(\mathbf{Axx}^\top)] = \operatorname{Tr}(\mathbf{A}\,\mathbb{E}[\mathbf{xx}^\top])$. Since $\mathbb{E}[\mathbf{xx}^\top] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top$, we get $\operatorname{Tr}(\mathbf{A}\boldsymbol{\Sigma}) + \operatorname{Tr}(\mathbf{A}\boldsymbol{\mu}\boldsymbol{\mu}^\top) = \operatorname{Tr}(\mathbf{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}^\top\mathbf{A}\boldsymbol{\mu}$. This single identity computes the expected value of *any* quadratic form — e.g. expected portfolio variance under parameter uncertainty.

## 4 The Four Fundamental Subspaces

This section is the structural heart of the subject. Every matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ carries four subspaces, and understanding how they fit together explains least squares, rank deficiency, and the geometry of solving $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$.

### 4.1 Column space and null space, and proofs that they are subspaces

**Definition 5.** The **column space** $\mathcal{C}(\mathbf{X}) = \{\mathbf{Xv} : \mathbf{v} \in \mathbb{R}^n\} \subseteq \mathbb{R}^m$ is the set of all reachable outputs — the span of the columns. The **null space** $\mathcal{N}(\mathbf{X}) = \{\mathbf{v} \in \mathbb{R}^n : \mathbf{Xv} = \mathbf{0}\} \subseteq \mathbb{R}^n$ is the set of inputs the map annihilates.

$\mathcal{C}(\mathbf{X})$ is a subspace by the same add-and-scale argument as for span. $\mathcal{N}(\mathbf{X})$ is a subspace too: if $\mathbf{Xu} = \mathbf{0}$ and $\mathbf{Xv} = \mathbf{0}$ then $\mathbf{X}(a\mathbf{u} + b\mathbf{v}) = a\mathbf{Xu} + b\mathbf{Xv} = \mathbf{0}$ (using *linearity of* $\mathbf{X}$), and $\mathbf{X0} = \mathbf{0}$. Notice the two spaces live in *different* ambient spaces: outputs in $\mathbb{R}^m$, inputs in $\mathbb{R}^n$.

**Intuition**

$\mathcal{C}(\mathbf{X})$ answers "what can this matrix produce?"; $\mathcal{N}(\mathbf{X})$ answers "what does this matrix ignore?" In regression, $\mathcal{C}(\mathbf{X})$ is the set of all fitted-value vectors the model can produce — the predictions you can possibly make. $\mathcal{N}(\mathbf{X})$ being nontrivial means there are coefficient directions that change

nothing about the predictions: the textbook signature of perfect multicollinearity and non-identifiability.

## 4.2 Rank and the Rank–Nullity Theorem

The **rank** $r = \operatorname{rank}(\mathbf{X})$ is the dimension of the column space, equivalently the maximal number of linearly independent columns. A foundational fact, often surprising on first encounter, is that the row rank equals the column rank — the number of independent columns equals the number of independent rows. (One clean proof: row operations preserve row space dimension and column dependence relations simultaneously; reducing to row echelon form, the number of pivots counts both.)

**Theorem 3** (Rank–Nullity). *For* $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\quad \operatorname{rank}(\mathbf{X}) + \dim \mathcal{N}(\mathbf{X}) = n.$

*Proof.* Choose a basis $\{\mathbf{z}_1, \dots, \mathbf{z}_d\}$ of $\mathcal{N}(\mathbf{X})$ where $d = \dim \mathcal{N}(\mathbf{X})$. Extend it to a basis $\{\mathbf{z}_1, \dots, \mathbf{z}_d, \mathbf{w}_1, \dots, \mathbf{w}_{n-d}\}$ of all of $\mathbb{R}^n$ (always possible: keep adding independent vectors until you span). Claim: $\{\mathbf{X}\mathbf{w}_1, \dots, \mathbf{X}\mathbf{w}_{n-d}\}$ is a basis of $\mathcal{C}(\mathbf{X})$, which would give rank $= n - d$, i.e. the theorem.

*They span $\mathcal{C}(\mathbf{X})$:* any output is $\mathbf{X}\mathbf{v}$ for some $\mathbf{v} = \sum_i a_i \mathbf{z}_i + \sum_j b_j \mathbf{w}_j$; applying $\mathbf{X}$ kills the $\mathbf{z}$ terms (they are in the null space), leaving $\mathbf{X}\mathbf{v} = \sum_j b_j \mathbf{X}\mathbf{w}_j$.

*They are independent:* suppose $\sum_j c_j \mathbf{X}\mathbf{w}_j = \mathbf{0}$. Then $\mathbf{X}(\sum_j c_j \mathbf{w}_j) = \mathbf{0}$, so $\sum_j c_j \mathbf{w}_j \in \mathcal{N}(\mathbf{X})$, hence equals some combination $\sum_i a_i \mathbf{z}_i$. But $\{\mathbf{z}_i\} \cup \{\mathbf{w}_j\}$ is a basis (independent), so all $c_j$ (and $a_i$) vanish. □

This is the conservation law of linear maps: the $n$ input dimensions are partitioned into those that get "used" (mapped to independent outputs, counted by rank) and those that get "lost" (collapsed to zero, counted by nullity). Nothing else can happen.

**Worked Example**

Take $\mathbf{X} = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{smallmatrix}\right)$. Row 2 is twice row 1, so there is one independent row: rank $= 1$. Rank–nullity predicts $\dim \mathcal{N}(\mathbf{X}) = n - r = 3 - 1 = 2$. Check: $\mathbf{X}\mathbf{v} = \mathbf{0}$ means $v_1 + 2v_2 + 3v_3 = 0$ (the second equation is redundant), one equation in three unknowns, whose solution set is a 2-dimensional plane. Confirmed. In a regression with these three collinear features, two whole dimensions of coefficient space leave predictions unchanged — the coefficients are determined only up to that 2D null space.

## 4.3 Orthogonal complements and the Fundamental Theorem

**Definition 6.** The orthogonal complement of a subspace $S \subseteq \mathbb{R}^k$ is $S^\perp = \{\mathbf{v} : \langle \mathbf{v}, \mathbf{s} \rangle = 0 \ \forall \mathbf{s} \in S\}$, the set of vectors perpendicular to everything in $S$.

**Theorem 4** (Fundamental Theorem of Linear Algebra, orthogonality form). *For* $\mathbf{X} \in \mathbb{R}^{m\times n}$*:*

$$\mathcal{N}(\mathbf{X}) = \mathcal{C}(\mathbf{X}^\top)^\perp \quad \text{in } \mathbb{R}^n, \qquad \mathcal{N}(\mathbf{X}^\top) = \mathcal{C}(\mathbf{X})^\perp \quad \text{in } \mathbb{R}^m.$$

*Consequently* $\mathbb{R}^n = \mathcal{C}(\mathbf{X}^\top) \oplus \mathcal{N}(\mathbf{X})$ *and* $\mathbb{R}^m = \mathcal{C}(\mathbf{X}) \oplus \mathcal{N}(\mathbf{X}^\top)$ *(orthogonal direct sums).*

*Proof.* We prove the first statement; the second is the same applied to $\mathbf{X}^\top$. A vector $\mathbf{v} \in \mathcal{N}(\mathbf{X})$ means $\mathbf{X}\mathbf{v} = \mathbf{0}$, i.e. every row of $\mathbf{X}$ dotted with $\mathbf{v}$ is zero. The rows of $\mathbf{X}$ are the columns of $\mathbf{X}^\top$, and they span $\mathcal{C}(\mathbf{X}^\top)$. So $\mathbf{v} \perp$ every spanning vector of the row space $\iff \mathbf{v} \perp \mathcal{C}(\mathbf{X}^\top) \iff \mathbf{v} \in \mathcal{C}(\mathbf{X}^\top)^\perp$. The dimensions add up by rank–nullity ($r + (n - r) = n$), confirming the direct sum. □

**Intuition**

Picture $\mathbb{R}^n$ split cleanly into two perpendicular pieces: the row space (directions that "matter," on which $\mathbf{X}$ acts invertibly) and the null space (directions that get crushed to zero). Any input decomposes uniquely into a part in each. This is the skeleton on which least squares hangs: we will project the data onto $\mathcal{C}(\mathbf{X})$ and the leftover residual will land in $\mathcal{N}(\mathbf{X}^\top) = \mathcal{C}(\mathbf{X})^\perp$, which is exactly why the residual is orthogonal to every column of $\mathbf{X}$.

# 5 Solving Linear Systems and the Geometry of Least Squares

## 5.1 When does $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ have a solution, and is it unique?

The system $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ asks: can $\mathbf{y}$ be written as a combination of the columns of $\mathbf{X}$? By definition that is possible *iff* $\mathbf{y} \in \mathcal{C}(\mathbf{X})$. Two regimes:

- **Existence** fails when $\mathbf{y} \notin \mathcal{C}(\mathbf{X})$ — there is no exact solution. This is the *generic* situation in data analysis: with $n$ noisy observations and $p < n$ features, $\mathbf{y}$ almost never lies exactly in the $p$-dimensional column space. We then seek the best approximate solution: least squares.
- **Uniqueness** fails when $\mathcal{N}(\mathbf{X}) \neq \{\mathbf{0}\}$: if $\boldsymbol{\beta}$ solves the system and $\boldsymbol{\delta} \in \mathcal{N}(\mathbf{X})$, then $\mathbf{X}(\boldsymbol{\beta} + \boldsymbol{\delta}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{0} = \mathbf{y}$, so $\boldsymbol{\beta} + \boldsymbol{\delta}$ is also a solution. The solution set is an entire affine copy of the null space.

**Intuition**

These two failure modes are the two diseases of regression. Non-existence is benign and expected — it is why we minimize squared error instead of demanding exactness. Non-uniqueness (a nontrivial null space, i.e. collinear features or $p > n$) is the dangerous one: the data alone cannot pick a coefficient vector, and we must add information (regularization) to break the tie.

## 5.2 The least-squares problem and why we project

When $\mathbf{y} \notin \mathcal{C}(\mathbf{X})$ we instead minimize the squared Euclidean distance:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 .$$

Geometrically, $\mathbf{X}\boldsymbol{\beta}$ ranges over all of $\mathcal{C}(\mathbf{X})$ as $\boldsymbol{\beta}$ varies, so we are asking: *which point in the subspace $\mathcal{C}(\mathbf{X})$ is closest to $\mathbf{y}$?* The answer, intuitively and provably, is the orthogonal projection of $\mathbf{y}$ onto $\mathcal{C}(\mathbf{X})$.

**Theorem 5** (Orthogonality principle)**.** *$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ minimizes $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2$ over the subspace if and only if the residual $\mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to the subspace, i.e. $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}$.*

*Proof.* Let $\boldsymbol{p} = \hat{\mathbf{y}}$ be the orthogonal projection and let $\boldsymbol{q} = \mathbf{X}\boldsymbol{\beta}$ be any other point in $\mathcal{C}(\mathbf{X})$. Write $\mathbf{y} - \boldsymbol{q} = (\mathbf{y} - \boldsymbol{p}) + (\boldsymbol{p} - \boldsymbol{q})$. The first term is orthogonal to the subspace (by assumption), and the second term lies *in* the subspace, so the two are orthogonal to each other. Pythagoras then gives

$$\|\mathbf{y} - \boldsymbol{q}\|^2 = \|\mathbf{y} - \boldsymbol{p}\|^2 + \|\boldsymbol{p} - \boldsymbol{q}\|^2 \geq \|\mathbf{y} - \boldsymbol{p}\|^2 ,$$

with equality iff $\boldsymbol{p} = \boldsymbol{q}$. So the projection is the unique minimizer. Conversely, if the residual were *not* orthogonal, we could move slightly within the subspace to decrease the distance, so a minimizer must satisfy orthogonality. The condition "residual $\perp$ every column" is exactly $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}$. □

Rearranging the orthogonality condition gives the **normal equations**

$$\mathbf{X}^\top\mathbf{X}\,\hat{\boldsymbol{\beta}} = \mathbf{X}^\top\mathbf{y}.$$

**Every symbol now has a reason.** $\mathbf{X}^\top$ appears because orthogonality to the columns of $\mathbf{X}$ is expressed by hitting the residual with $\mathbf{X}^\top$ (the adjoint identity again). $\mathbf{X}^\top\mathbf{X}$ is the Gram matrix of the columns — it records all pairwise inner products of features — and it is invertible exactly when the columns are independent (no null space), which is exactly the uniqueness condition. When it is invertible,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}.$$

**Worked Example**

Fit a line $y = \beta_0 + \beta_1 x$ to the three points $(x, y) = (0,1), (1,2), (2,2)$. The design matrix and response are

$$\mathbf{X} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}.$$

Then $\mathbf{X}^\top\mathbf{X} = \left(\begin{smallmatrix} 3 & 3 \\ 3 & 5 \end{smallmatrix}\right)$ and $\mathbf{X}^\top\mathbf{y} = \left(\begin{smallmatrix} 5 \\ 6 \end{smallmatrix}\right)$. Solving $\left(\begin{smallmatrix} 3 & 3 \\ 3 & 5 \end{smallmatrix}\right)\hat{\boldsymbol{\beta}} = \left(\begin{smallmatrix} 5 \\ 6 \end{smallmatrix}\right)$: the inverse of $\mathbf{X}^\top\mathbf{X}$ is $\frac{1}{3\cdot5-9}\left(\begin{smallmatrix} 5 & -3 \\ -3 & 3 \end{smallmatrix}\right) = \frac{1}{6}\left(\begin{smallmatrix} 5 & -3 \\ -3 & 3 \end{smallmatrix}\right)$, giving $\hat{\boldsymbol{\beta}} = \frac{1}{6}\left(\begin{smallmatrix} 5\cdot5-3\cdot6 \\ -3\cdot5+3\cdot6 \end{smallmatrix}\right) = \frac{1}{6}\left(\begin{smallmatrix} 7 \\ 3 \end{smallmatrix}\right) = \begin{pmatrix} 7/6 \\ 1/2 \end{pmatrix}$. So the fitted line is $y = 7/6 + x/2$. Sanity check the orthogonality: residuals are $\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} = (1,2,2) - (7/6, 5/3, 13/6) = (-1/6, 1/3, -1/6)$, which sum to 0 (orthogonal to the intercept column of ones) and dot with $(0,1,2)$ to $0 + 1/3 - 1/3 = 0$ (orthogonal to the $x$ column). The geometry checks out exactly.

### 5.3 The hat matrix as a projector

Substituting back, the fitted values are $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} =: \mathbf{H}\mathbf{y}$. The matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ is the **projection ("hat") matrix**. Its defining algebraic properties each have a geometric meaning:

- **Symmetric** $\mathbf{H}^\top = \mathbf{H}$: orthogonal projectors are self-adjoint. Verify: $\mathbf{H}^\top = \mathbf{X}((\mathbf{X}^\top\mathbf{X})^{-1})^\top\mathbf{X}^\top = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top = \mathbf{H}$ since $\mathbf{X}^\top\mathbf{X}$ is symmetric.
- **Idempotent** $\mathbf{H}^2 = \mathbf{H}$: projecting twice is the same as projecting once (you are already in the subspace after the first projection). Verify: $\mathbf{H}^2 = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\underbrace{\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}}_{=\mathbf{I}}\mathbf{X}^\top = \mathbf{H}$.
- **Trace equals rank** $\mathrm{Tr}(\mathbf{H}) = p$: using cyclicity, $\mathrm{Tr}(\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top) = \mathrm{Tr}((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}) = \mathrm{Tr}(\mathbf{I}_p) = p$. This counts the model's degrees of freedom — the dimension of the space it projects onto.
- **Eigenvalues in** $\{0,1\}$: from $\mathbf{H}^2 = \mathbf{H}$, any eigenvalue $\lambda$ satisfies $\lambda^2 = \lambda$, so $\lambda \in \{0,1\}$. Eigenvalue 1 for directions inside $\mathcal{C}(\mathbf{X})$ (left unchanged), 0 for directions orthogonal to it (annihilated).

The complementary matrix $\mathbf{I} - \mathbf{H}$ projects onto the orthogonal complement $\mathcal{C}(\mathbf{X})^\perp = \mathcal{N}(\mathbf{X}^\top)$, and it produces the residuals $\hat{\boldsymbol{\varepsilon}} = (\mathbf{I}-\mathbf{H})\mathbf{y}$. That the residual lives in $\mathcal{N}(\mathbf{X}^\top)$ is just the Fundamental Theorem in action.

**Intuition**

The diagonal entry $h_{ii}$ of $\mathbf{H}$ is the *leverage* of observation $i$: it equals $\partial\hat{y}_i/\partial y_i$, how much the $i$-th fitted value responds to its own observation. High leverage means an observation sits far out in feature space and can single-handedly tilt the fit — the formal object behind "this one stress-period data point is driving my whole regression." Since $\mathrm{Tr}(\mathbf{H}) = p = \sum_i h_{ii}$ and each $h_{ii} \in [0,1]$, leverage is a budget of $p$ units shared among $n$ observations.

## 6 Norms, Inner Products, and Angles

### 6.1 Norms: measuring size, and why different norms matter

**Definition 7** (Norm)**.** A norm on $\mathbb{R}^n$ is a function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ with: (i) $\|\mathbf{x}\| \geq 0$ with equality iff $\mathbf{x} = \mathbf{0}$ (positivity); (ii) $\|a\mathbf{x}\| = |a|\,\|\mathbf{x}\|$ (homogeneity); (iii) $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ (triangle inequality).

The $\ell_p$ family is $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$ for $p \geq 1$, with $\|\mathbf{x}\|_\infty = \max_i |x_i|$. The three that dominate ML:

- $\ell_2$ (Euclidean): the geometric length; rotation-invariant; the only $\ell_p$ norm coming from an inner product. Smooth everywhere, which is why squared-$\ell_2$ objectives have clean closed-form gradients.
- $\ell_1$ (Manhattan / taxicab): sum of absolute values. Has "corners" on the axes, which is the entire reason $\ell_1$ regularization produces *sparse* solutions (coefficients exactly zero) — a fact we make precise below.
- $\ell_\infty$: the largest coordinate; used in worst-case/robustness bounds.

**Worked Example**

For $\mathbf{x} = (3, -4)$: $\|\mathbf{x}\|_1 = 3+4 = 7$, $\|\mathbf{x}\|_2 = \sqrt{9+16} = 5$, $\|\mathbf{x}\|_\infty = 4$. Notice $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$ — a general ordering. The unit balls nest accordingly: the $\ell_1$ ball (diamond) sits inside the $\ell_2$ ball (circle) sits inside the $\ell_\infty$ ball (square).

**Intuition: why $\ell_1$ gives sparsity**

Imagine minimizing a smooth loss subject to a "budget" $\|\boldsymbol{\beta}\| \leq t$. The optimum is where the loss's elliptical contours first touch the budget ball. The $\ell_2$ ball is round, so the touch point is generically at a smooth spot with all coordinates nonzero. The $\ell_1$ ball is a diamond whose sharp corners poke out *along the axes* — where some coordinates are exactly zero. Elliptical contours expanding outward tend to hit a corner first. So $\ell_1$ "wants" to set coordinates to zero; $\ell_2$ merely shrinks them. This picture is the geometric soul of lasso vs. ridge.

## 6.2 Inner products, Cauchy–Schwarz, and the meaning of angle

The standard inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_i x_i y_i$ induces the $\ell_2$ norm via $\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. It lets us define angle, and the legitimacy of that definition rests on:

**Theorem 6** (Cauchy–Schwarz). $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \, \|\mathbf{y}\|_2$, *with equality iff* $\mathbf{x}, \mathbf{y}$ *are parallel.*

*Proof.* If $\mathbf{y} = \mathbf{0}$ both sides are 0. Otherwise consider the quadratic in $t$: $0 \leq \|\mathbf{x} - t\mathbf{y}\|^2 = \|\mathbf{x}\|^2 - 2t\langle \mathbf{x}, \mathbf{y} \rangle + t^2 \|\mathbf{y}\|^2$. A quadratic $at^2 + bt + c \geq 0$ for all $t$ must have non-positive discriminant $b^2 - 4ac \leq 0$. Here that reads $4\langle \mathbf{x}, \mathbf{y} \rangle^2 - 4\|\mathbf{y}\|^2 \|\mathbf{x}\|^2 \leq 0$, i.e. $\langle \mathbf{x}, \mathbf{y} \rangle^2 \leq \|\mathbf{x}\|^2 \|\mathbf{y}\|^2$. Taking square roots gives the inequality; equality requires the quadratic to have a real root $t^\star$, i.e. $\mathbf{x} = t^\star \mathbf{y}$ (parallel). □

Because $|\langle \mathbf{x}, \mathbf{y} \rangle| / (\|\mathbf{x}\| \, \|\mathbf{y}\|) \leq 1$, we may *define* the angle by $\cos\theta = \langle \mathbf{x}, \mathbf{y} \rangle / (\|\mathbf{x}\| \, \|\mathbf{y}\|)$ and be sure it lands in $[-1, 1]$. This quantity is the **cosine similarity** ubiquitous in ML (document similarity, embedding retrieval). Orthogonality $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ means $\theta = 90°$: no shared component.

**Worked Example**

In a quant context, the (sample) correlation between two demeaned return series is exactly the cosine of the angle between them as vectors in $\mathbb{R}^n$: $\rho = \langle \mathbf{x}, \mathbf{y} \rangle / (\|\mathbf{x}\| \, \|\mathbf{y}\|)$. Correlation $\pm 1$ means the vectors are parallel/antiparallel (perfectly collinear returns); correlation 0 means orthogonal. So a correlation matrix is a matrix of pairwise cosines, and "diversification" is literally seeking assets at large angles to one another.

## 6.3 Orthonormal bases and orthogonal matrices

A set $\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$ is orthonormal if $\langle \mathbf{q}_i, \mathbf{q}_j \rangle = \delta_{ij}$ (unit length, mutually perpendicular). Orthonormal bases are computationally golden because coordinates are just inner products: if $\mathbf{x} = \sum_j c_j \mathbf{q}_j$

then $c_j = \langle \mathbf{x}, \mathbf{q}_j \rangle$ (dot both sides with $\mathbf{q}_j$ and use orthonormality). No system to solve.

Assemble orthonormal columns into a square matrix $\mathbf{Q}$. Then $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$ (entry $(i,j)$ is $\langle \mathbf{q}_i, \mathbf{q}_j \rangle = \delta_{ij}$), so $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ — the inverse is free. Such $\mathbf{Q}$ is **orthogonal** and represents a rigid motion: $\|\mathbf{Q}\mathbf{x}\|^2 = \mathbf{x}^\top \mathbf{Q}^\top \mathbf{Q}\mathbf{x} = \mathbf{x}^\top \mathbf{x} = \|\mathbf{x}\|^2$, so it preserves all lengths and (by polarization) all angles. Rotations and reflections are exactly the orthogonal matrices. They are the numerically safest transformations because they neither amplify nor shrink errors — a property we will exploit in the SVD and QR.

# 7 Eigenvalues and Eigenvectors

## 7.1 Definition and the characteristic equation

**Definition 8** (Eigenpair)**.** A nonzero vector $\mathbf{v}$ is an eigenvector of a square matrix $\mathbf{A}$ with eigenvalue $\lambda$ if $\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$: the map $\mathbf{A}$ acts on $\mathbf{v}$ purely by scaling, not rotating.

Eigenvectors are the special directions an operator leaves invariant up to scale. To find them, rewrite $\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$ as $(\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = \mathbf{0}$. For a *nonzero* $\mathbf{v}$ to satisfy this, $\mathbf{A} - \lambda \mathbf{I}$ must have a nontrivial null space, i.e. be singular, i.e.

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0.$$

This is the **characteristic polynomial**, a degree-$n$ polynomial in $\lambda$ whose roots are the eigenvalues. Once an eigenvalue is known, the eigenvectors are the null space of $\mathbf{A} - \lambda \mathbf{I}$.

**Worked Example**

$\mathbf{A} = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$. Characteristic polynomial: $\det\left(\begin{smallmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{smallmatrix}\right) = (2-\lambda)^2 - 1 = \lambda^2 - 4\lambda + 3 = (\lambda - 1)(\lambda - 3)$. Eigenvalues $\lambda = 1, 3$. For $\lambda = 3$: $(\mathbf{A} - 3\mathbf{I})\mathbf{v} = \left(\begin{smallmatrix} -1 & 1 \\ 1 & -1 \end{smallmatrix}\right)\mathbf{v} = \mathbf{0} \Rightarrow v_1 = v_2$, eigenvector $(1,1)/\sqrt{2}$. For $\lambda = 1$: $\left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)\mathbf{v} = \mathbf{0} \Rightarrow v_1 = -v_2$, eigenvector $(1,-1)/\sqrt{2}$. Note the eigenvectors are orthogonal — not a coincidence, because $\mathbf{A}$ is symmetric (the spectral theorem below).

## 7.2 Diagonalization: changing to the eigenbasis

Suppose $\mathbf{A} \in \mathbb{R}^{n \times n}$ has $n$ linearly independent eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ with eigenvalues $\lambda_1, \ldots, \lambda_n$. Put the eigenvectors as columns of $\mathbf{V}$ and the eigenvalues on the diagonal of $\mathbf{\Lambda}$. Then $\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}$ (column $j$ reads $\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j$), and since $\mathbf{V}$ is invertible,

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}.$$

This says: to apply $\mathbf{A}$, change coordinates to the eigenbasis ($\mathbf{V}^{-1}$), scale each axis by its eigenvalue ($\mathbf{\Lambda}$), and change back ($\mathbf{V}$). In the right coordinate system, $\mathbf{A}$ is just independent scalings.

The payoff is computing functions of $\mathbf{A}$ cheaply: $\mathbf{A}^k = \mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1}$ (the middle factor telescopes because $\mathbf{V}^{-1}\mathbf{V} = \mathbf{I}$ between every pair). Powers of $\mathbf{A}$ are governed entirely by powers of its eigenvalues.

**Worked example: a Markov chain's long-run behavior**

Let a regime-switching model have transition matrix $\mathbf{P} = \left(\begin{smallmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{smallmatrix}\right)$ (columns sum to 1). Its eigenvalues are 1 and 0.7. The eigenvalue-1 eigenvector (normalized to sum to 1) is the stationary distribution $\boldsymbol{\pi}$; the eigenvalue 0.7 controls how fast the chain forgets its start: after $k$ steps the deviation from stationarity scales like $0.7^k$. So $\mathbf{P}^k \to$ (rank-one stationary matrix) geometrically at rate 0.7. The *second-largest* eigenvalue magnitude is the mixing rate — a fact used everywhere from PageRank to MCMC convergence diagnostics.

### 7.3 Trace and determinant via eigenvalues

Two identities tie the "summary statistics" of a matrix to its spectrum:

$$\operatorname{Tr}(\mathbf{A}) = \sum_i \lambda_i, \qquad \det(\mathbf{A}) = \prod_i \lambda_i.$$

The trace identity follows from the characteristic polynomial's coefficients (the coefficient of $\lambda^{n-1}$ is both $-\operatorname{Tr}(\mathbf{A})$ and $-\sum \lambda_i$ by Vieta). The determinant identity is the constant term ($\det(\mathbf{A} - 0 \cdot \mathbf{I}) = \det \mathbf{A} = \prod \lambda_i$). Geometrically $\det \mathbf{A} = \prod \lambda_i$ is the factor by which $\mathbf{A}$ scales volume: each eigendirection is stretched by $\lambda_i$, so the product is the total volume scaling. $\det \mathbf{A} = 0$ iff some $\lambda_i = 0$ iff $\mathbf{A}$ collapses a dimension iff $\mathbf{A}$ is singular.

### 7.4 The Spectral Theorem — the result you use most

Symmetric matrices ($\mathbf{A} = \mathbf{A}^\top$) are special and pervasive: covariance matrices, Gram matrices $\mathbf{X}^\top\mathbf{X}$, kernel matrices, and Hessians are all symmetric. They enjoy the cleanest possible eigenstructure.

**Theorem 7** (Spectral Theorem for real symmetric matrices)**.** *If $\mathbf{A} = \mathbf{A}^\top \in \mathbb{R}^{n\times n}$, then $\mathbf{A}$ has $n$ real eigenvalues and an* orthonormal *basis of eigenvectors. Equivalently $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ with $\mathbf{Q}$ orthogonal ($\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$) and $\mathbf{\Lambda}$ real diagonal. Writing columns out, $\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{q}_i\mathbf{q}_i^\top$.*

*Proof of the two key parts. Real eigenvalues:* Let $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ with possibly complex $\mathbf{v}$. Take the conjugate-transpose inner product with $\mathbf{v}$: $\bar{\mathbf{v}}^\top\mathbf{A}\mathbf{v} = \lambda\,\bar{\mathbf{v}}^\top\mathbf{v} = \lambda\,\|\mathbf{v}\|^2$. But $\bar{\mathbf{v}}^\top\mathbf{A}\mathbf{v}$ is its own conjugate transpose (since $\mathbf{A}$ is real symmetric, $\overline{\bar{\mathbf{v}}^\top\mathbf{A}\mathbf{v}} = \mathbf{v}^\top\mathbf{A}\bar{\mathbf{v}} = \bar{\mathbf{v}}^\top\mathbf{A}\mathbf{v}$ after transposing the scalar), hence real. A real number divided by the positive real $\|\mathbf{v}\|^2$ is real, so $\lambda \in \mathbb{R}$.

*Orthogonal eigenvectors for distinct eigenvalues:* Let $\mathbf{A}\mathbf{u} = \alpha\mathbf{u}$, $\mathbf{A}\mathbf{v} = \beta\mathbf{v}$, $\alpha \neq \beta$. Then $\alpha\langle\mathbf{u},\mathbf{v}\rangle = \langle\mathbf{A}\mathbf{u},\mathbf{v}\rangle = \langle\mathbf{u},\mathbf{A}\mathbf{v}\rangle = \beta\langle\mathbf{u},\mathbf{v}\rangle$ (using symmetry to move $\mathbf{A}$ across). So $(\alpha-\beta)\langle\mathbf{u},\mathbf{v}\rangle = 0$, and since $\alpha \neq \beta$, $\langle\mathbf{u},\mathbf{v}\rangle = 0$. (Repeated eigenvalues require choosing an orthonormal basis within each eigenspace, always possible.) □

**Intuition**

The spectral theorem says every symmetric matrix is, in the right orthonormal coordinates, a pure diagonal scaling — it stretches space along $n$ mutually perpendicular axes. The expansion $\mathbf{A} = \sum_i \lambda_i \mathbf{q}_i\mathbf{q}_i^\top$ writes $\mathbf{A}$ as a weighted sum of rank-one projectors $\mathbf{q}_i\mathbf{q}_i^\top$, each projecting onto one eigen-axis and scaling by $\lambda_i$. PCA is nothing but this decomposition applied to a covariance matrix: the $\mathbf{q}_i$ are principal directions, the $\lambda_i$ the variances along them.

## 8 Positive Definiteness and Quadratic Forms

### 8.1 Quadratic forms and what their sign means

A **quadratic form** is $q(\mathbf{x}) = \mathbf{x}^\top\mathbf{A}\mathbf{x}$ for symmetric $\mathbf{A}$ (any non-symmetric part cancels, so we assume symmetry WLOG). Expanding, $q(\mathbf{x}) = \sum_{i,j} A_{ij}x_ix_j$. This is the multivariate generalization of $ax^2$, and its "curvature" is governed by $\mathbf{A}$.

**Definition 9.** $\mathbf{A}$ is **positive definite** ($\mathbf{A} \succ 0$) if $\mathbf{x}^\top\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$; **positive semidefinite** ($\mathbf{A} \succeq 0$) if $\mathbf{x}^\top\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x}$.

**Theorem 8** (Definiteness $\iff$ eigenvalue signs)**.** *For symmetric $\mathbf{A}$: $\mathbf{A} \succ 0 \iff$ all eigenvalues $\lambda_i > 0$; $\mathbf{A} \succeq 0 \iff$ all $\lambda_i \geq 0$.*

*Proof.* Diagonalize $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ and substitute $\mathbf{y} = \mathbf{Q}^\top\mathbf{x}$ (an invertible change of variables, $\mathbf{y} \neq \mathbf{0} \iff \mathbf{x} \neq \mathbf{0}$). Then $\mathbf{x}^\top\mathbf{A}\mathbf{x} = \mathbf{x}^\top\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top\mathbf{x} = \mathbf{y}^\top\mathbf{\Lambda}\mathbf{y} = \sum_i \lambda_i y_i^2$. This is positive for all $\mathbf{y} \neq \mathbf{0}$ exactly when every $\lambda_i > 0$ (otherwise pick $\mathbf{y}$ along the offending axis to make it $\leq 0$). The PSD case is identical with $\geq$. □

So definiteness is just "all eigenvalues positive," read through the eigenbasis where the quadratic form becomes a sum of squares $\sum \lambda_i y_i^2$.

### 8.2 Why this concept is everywhere

- **Covariance matrices are PSD.** For any $\mathbf{x}$, $\mathbf{x}^\top\mathbf{\Sigma}\mathbf{x} = \mathbf{x}^\top\mathbb{E}[(\mathbf{z}-\boldsymbol{\mu})(\mathbf{z}-\boldsymbol{\mu})^\top]\mathbf{x} = \mathbb{E}[(\mathbf{x}^\top(\mathbf{z}-\boldsymbol{\mu}))^2] = \mathrm{Var}(\mathbf{x}^\top\mathbf{z}) \geq 0$. A variance cannot be negative — so the matrix that produces variances must be PSD. A negative eigenvalue would mean a portfolio with negative variance, an impossibility; when estimated covariance matrices come out non-PSD (from noise or missing data), risk systems must "repair" them.
- **Convexity.** A twice-differentiable $f$ is convex iff its Hessian $\nabla^2 f \succeq 0$ everywhere. The quadratic form $\mathbf{x}^\top(\nabla^2 f)\mathbf{x}$ being nonnegative means the function curves upward in every direction — no saddles, every local min is global. This is why PD Hessians guarantee unique optima.
- **Valid kernels.** A function $K$ is a valid (Mercer) kernel iff every Gram matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$ is PSD — the condition that makes the SVM dual a convex problem and guarantees an underlying feature space exists.

### 8.3 Tests for positive definiteness

Three practical tests, each with its use:

1. **Eigenvalues.** Compute them; check all $> 0$. Most informative, most expensive.
2. **Sylvester's criterion.** All leading principal minors (determinants of top-left $k \times k$ blocks) are positive. Cheap for small matrices and by hand.
3. **Cholesky.** Attempt $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ with $\mathbf{L}$ lower-triangular, positive diagonal. It succeeds iff $\mathbf{A} \succ 0$. This is the *numerical* test of choice: fast ($\sim n^3/3$ flops, half of LU) and it either completes or fails on a non-positive pivot.

**Worked Example**

Is $\mathbf{A} = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ positive definite? *Sylvester:* leading minors are $2 > 0$ and $\det = 3 > 0$. Yes. *Eigenvalues:* we found $1, 3 > 0$ earlier. Yes. *Geometric reading:* $\mathbf{x}^\top\mathbf{A}\mathbf{x} = 2x_1^2 + 2x_1x_2 + 2x_2^2$; its level sets $\{q(\mathbf{x}) = c\}$ are ellipses, with axes along the eigenvectors $(1,1), (1,-1)$ and semi-axis lengths $\propto 1/\sqrt{\lambda}$, so the long axis is along $(1,-1)$ (smaller eigenvalue 1) and the short axis along $(1,1)$ (larger eigenvalue 3).

## 9 The Singular Value Decomposition

The SVD is the most important matrix factorization in applied mathematics. Unlike eigendecomposition, it exists for *every* matrix, rectangular or not, and it directly exposes rank, the four subspaces, the best low-rank approximation, and the conditioning of a least-squares problem.

### 9.1 Statement and derivation

**Theorem 9** (SVD). *Every $\mathbf{X} \in \mathbb{R}^{m\times n}$ factors as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthogonal, and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is "diagonal" with nonnegative entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 = \cdots = 0$, where $r = \mathrm{rank}(\mathbf{X})$.*

*Constructive derivation.* The matrix $\mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{n\times n}$ is symmetric and PSD (it is a Gram matrix: $\mathbf{v}^\top\mathbf{X}^\top\mathbf{X}\mathbf{v} = \|\mathbf{X}\mathbf{v}\|^2 \geq 0$). By the spectral theorem it has an orthonormal eigenbasis $\mathbf{v}_1, \ldots, \mathbf{v}_n$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. Define $\sigma_i = \sqrt{\lambda_i}$ (the *singular values*). For the $r$ indices with $\sigma_i > 0$, set $\mathbf{u}_i = \mathbf{X}\mathbf{v}_i/\sigma_i$. These $\mathbf{u}_i$ are orthonormal:

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \frac{1}{\sigma_i\sigma_j}\mathbf{v}_i^\top\mathbf{X}^\top\mathbf{X}\mathbf{v}_j = \frac{\lambda_j}{\sigma_i\sigma_j}\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \frac{\lambda_j}{\sigma_i\sigma_j}\delta_{ij} = \delta_{ij}.$$

By construction $\mathbf{X}\mathbf{v}_i = \sigma_i\mathbf{u}_i$ for all $i$ (for $i > r$, $\mathbf{X}\mathbf{v}_i = \mathbf{0}$ since $\|\mathbf{X}\mathbf{v}_i\|^2 = \lambda_i = 0$). Stacking these relations as columns gives $\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$, and right-multiplying by $\mathbf{V}^\top$ (orthogonal) yields $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. □

So the right singular vectors $\mathbf{v}_i$ are eigenvectors of $\mathbf{X}^\top\mathbf{X}$, the left singular vectors $\mathbf{u}_i$ are eigenvectors of $\mathbf{X}\mathbf{X}^\top$, and the singular values are the square roots of the shared nonzero eigenvalues. The relation $\mathbf{X}\mathbf{v}_i = \sigma_i\mathbf{u}_i$ is the soul of the SVD: $\mathbf{X}$ *maps an orthonormal input frame* $\{\mathbf{v}_i\}$ *to an orthogonal output frame* $\{\sigma_i\mathbf{u}_i\}$. Every linear map, in suitable orthonormal coordinates, is a diagonal scaling.

**Intuition**

Recall the unit-sphere-to-ellipsoid image from the start. The $\mathbf{v}_i$ are the pre-images of the ellipsoid's axes; the $\mathbf{u}_i$ are the axis directions; the $\sigma_i$ are the semi-axis lengths. $\sigma_{\max}$ is the largest stretch the matrix can apply to any unit vector, $\sigma_{\min}$ the smallest. This is why $\|\mathbf{X}\|_2 = \sigma_{\max}$ (the operator norm is the maximum stretch) and why the condition number is $\sigma_{\max}/\sigma_{\min}$ (worst-to-best stretch ratio).

## 9.2 SVD and the four subspaces

The SVD hands you orthonormal bases for all four fundamental subspaces at once: $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ span $\mathcal{C}(\mathbf{X})$; $\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_m\}$ span $\mathcal{N}(\mathbf{X}^\top)$; $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ span $\mathcal{C}(\mathbf{X}^\top)$ (row space); $\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n\}$ span $\mathcal{N}(\mathbf{X})$. The rank is simply the count of nonzero singular values — and, crucially, this count is *numerically robust*: instead of asking "is this determinant exactly zero?" (hopeless in floating point) we ask "how many singular values exceed a tolerance?".

## 9.3 Eckart–Young: best low-rank approximation

**Theorem 10** (Eckart–Young–Mirsky)**.** *The truncated SVD* $\mathbf{X}_k = \sum_{i=1}^{k}\sigma_i\mathbf{u}_i\mathbf{v}_i^\top$ *is the closest rank-k matrix to* $\mathbf{X}$ *in Frobenius (and spectral) norm:*

$$\min_{\operatorname{rank}(\mathbf{B})\leq k} \|\mathbf{X} - \mathbf{B}\|_F = \|\mathbf{X} - \mathbf{X}_k\|_F = \sqrt{\sum_{i>k}\sigma_i^2}.$$

The proof idea: in the orthonormal frame, $\|\mathbf{X} - \mathbf{B}\|_F^2$ decomposes over singular directions, and you minimize it by keeping the $k$ largest $\sigma_i$ and zeroing the rest. The reconstruction error is the energy in the discarded singular values.

**Intuition**

This single theorem powers an astonishing range of methods: PCA (best low-rank approximation of the data/covariance), image and data compression (store $k$ singular triples instead of the full matrix), latent semantic analysis, noise reduction (small singular values are often noise), and recommender systems / matrix completion (approximate a sparse ratings matrix by a low-rank one). Whenever you hear "low-rank structure," Eckart–Young is the justification that truncating

the SVD is optimal.

**Worked Example**

Suppose a $1000 \times 50$ matrix of daily returns for 50 assets has singular values that drop off sharply after the first 3. Eckart–Young says the best rank-3 summary captures $\sum_{i \le 3} \sigma_i^2 / \sum_i \sigma_i^2$ of the total variance. If that ratio is 0.92, three latent factors explain 92% of the comovement — the empirical basis of factor models (market, size, value, ...). The discarded directions, with tiny $\sigma_i$, are idiosyncratic noise.

### 9.4 The pseudoinverse and minimum-norm least squares

When $\mathbf{X}^\top \mathbf{X}$ is singular, the normal equations have infinitely many solutions. The SVD resolves this cleanly. Define the **Moore–Penrose pseudoinverse** $\mathbf{X}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^\top$, where $\mathbf{\Sigma}^+$ inverts each nonzero singular value $(1/\sigma_i)$ and transposes the shape. Then $\hat{\boldsymbol{\beta}} = \mathbf{X}^+\mathbf{y}$ is the least-squares solution of *minimum $\ell_2$ norm* — among all coefficient vectors achieving the smallest residual, it picks the shortest one. This is the principled default when features are collinear or $p > n$, and it is the $\lambda \to 0^+$ limit of ridge regression, tying directly to regularization.

## 10 Condition Number and Numerical Stability

The **condition number** of $\mathbf{X}$ is $\kappa(\mathbf{X}) = \sigma_{\max}/\sigma_{\min}$. It quantifies how much a relative error in the input can be amplified in the output when solving a linear system. If $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ and $\mathbf{y}$ is perturbed by $\delta\mathbf{y}$, the relative change in the solution is bounded by

$$\frac{\|\delta\boldsymbol{\beta}\|}{\|\boldsymbol{\beta}\|} \le \kappa(\mathbf{X}) \, \frac{\|\delta\mathbf{y}\|}{\|\mathbf{y}\|}.$$

A large $\kappa$ (an *ill-conditioned* matrix) means tiny data perturbations cause huge swings in the answer. The reason is geometric: an ill-conditioned $\mathbf{X}$ has a very small $\sigma_{\min}$, i.e. it nearly collapses some direction, and inverting that near-collapse enormously amplifies any error component along it.

**Never form the normal equations when ill-conditioned**

Solving least squares via $\mathbf{X}^\top\mathbf{X}$ *squares* the condition number: $\kappa(\mathbf{X}^\top\mathbf{X}) = \kappa(\mathbf{X})^2$ (because the singular values of $\mathbf{X}^\top\mathbf{X}$ are $\sigma_i^2$). If $\mathbf{X}$ has $\kappa = 10^6$ (not unusual with correlated features), then $\mathbf{X}^\top\mathbf{X}$ has $\kappa = 10^{12}$, which annihilates roughly 12 digits of precision — catastrophic in double precision (which has $\sim 16$). The fix is to never square: solve via $\mathbf{QR}$ ($\mathbf{X} = \mathbf{QR}$, then $\mathbf{R}\boldsymbol{\beta} = \mathbf{Q}^\top\mathbf{y}$) or **SVD**, both of which work with $\kappa(\mathbf{X})$ directly.

**Multicollinearity is ill-conditioning**

Two nearly-identical features make two columns of $\mathbf{X}$ nearly parallel, so one singular value is tiny: $\sigma_{\min} \approx 0$, $\kappa$ huge. The least-squares coefficients then have enormous variance (recall $\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$, and $(\mathbf{X}^\top\mathbf{X})^{-1}$ has eigenvalues $1/\sigma_i^2$, exploding for small $\sigma_i$). This is *exactly* why correlated factors give unstable, sign-flipping loadings. Ridge regression adds $\lambda\mathbf{I}$, lifting every $\sigma_i^2$ to $\sigma_i^2 + \lambda$, bounding the inverse and taming the condition number — regularization as numerical conditioning. In portfolio optimization the same disease afflicts $\mathbf{\Sigma}^{-1}$; Ledoit–Wolf shrinkage is the analogue of ridge for covariance matrices.

# 11 Essential Decompositions in Practice

- **LU** ($\mathbf{A} = \mathbf{PLU}$, with pivoting): Gaussian elimination, packaged. The general-purpose solver for square systems; $\sim 2n^3/3$ flops. Once computed, solving for many right-hand sides is cheap (forward/back substitution).
- **Cholesky** ($\mathbf{A} = \mathbf{L}\mathbf{L}^\top$, $\mathbf{A} \succ 0$): the specialized, twice-as-fast solver for symmetric positive-definite systems. Used to sample correlated Gaussians ($\boldsymbol{\mu} + \mathbf{L}\mathbf{z}$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ has covariance $\mathbf{L}\mathbf{L}^\top = \boldsymbol{\Sigma}$), in Gaussian-process regression, and in Kalman filtering. It also *is* the PD test.
- **QR** ($\mathbf{X} = \mathbf{QR}$, $\mathbf{Q}$ orthogonal, $\mathbf{R}$ upper-triangular): the numerically stable route to least squares. Because $\mathbf{Q}$ is orthogonal it preserves norms, so $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\| = \|\mathbf{Q}^\top\mathbf{y} - \mathbf{R}\boldsymbol{\beta}\|$, and the triangular system $\mathbf{R}\boldsymbol{\beta} = \mathbf{Q}^\top\mathbf{y}$ is solved by back-substitution — all at condition number $\kappa(\mathbf{X})$, not its square.
- **Sherman–Morrison–Woodbury**: the inverse of a low-rank update,

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1}.$$

  It lets you update an inverse after a rank-$k$ change in $O(k \cdot n^2)$ instead of recomputing in $O(n^3)$. This is the engine of recursive least squares, the Kalman filter's update step, and efficient leave-one-out cross-validation (each LOO fit is a rank-one downdate, so all $n$ refits cost barely more than one).

**Why LOOCV is nearly free for linear models**

Leave-one-out CV naively requires $n$ refits. But removing one row is a rank-one change to $\mathbf{X}^\top\mathbf{X}$, so Woodbury gives each refit's prediction in closed form. The result is the famous shortcut $\text{LOOCV} = \frac{1}{n}\sum_i \left(\frac{\hat{\varepsilon}_i}{1-h_{ii}}\right)^2$, where $\hat{\varepsilon}_i$ is the ordinary residual and $h_{ii}$ the leverage — computed from a *single* fit. The whole cross-validation collapses to one regression plus the diagonal of the hat matrix.

# 12 Matrix Calculus: the Identities and Their Derivations

We will need derivatives of scalar functions with respect to vectors and matrices. The convention: $\nabla_{\mathbf{x}} f$ is the vector of partials, same shape as $\mathbf{x}$.

## 12.1 The core gradient identities, derived

- $\nabla_{\mathbf{x}}(\mathbf{a}^\top\mathbf{x}) = \mathbf{a}$. Because $\mathbf{a}^\top\mathbf{x} = \sum_j a_j x_j$, so $\partial/\partial x_k = a_k$. The gradient of a linear function is its coefficient vector.
- $\nabla_{\mathbf{x}}(\mathbf{x}^\top\mathbf{A}\mathbf{x}) = (\mathbf{A}+\mathbf{A}^\top)\mathbf{x}$, which is $2\mathbf{A}\mathbf{x}$ when $\mathbf{A}$ is symmetric. Derivation: $\mathbf{x}^\top\mathbf{A}\mathbf{x} = \sum_{i,j} A_{ij}x_i x_j$; differentiating w.r.t. $x_k$, the terms with $i = k$ contribute $\sum_j A_{kj}x_j$ and those with $j = k$ contribute $\sum_i A_{ik}x_i$, summing to $(\mathbf{A}\mathbf{x})_k + (\mathbf{A}^\top\mathbf{x})_k$.
- $\nabla_{\mathbf{x}} \|\mathbf{x}\|_2^2 = 2\mathbf{x}$. Special case of the above with $\mathbf{A} = \mathbf{I}$.

## 12.2 The least-squares gradient, in one line

Now the payoff. Let $J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{y}^\top\mathbf{y} - 2\boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}$. Differentiate term by term using the identities: the constant $\mathbf{y}^\top\mathbf{y}$ gives $\mathbf{0}$; the linear term $-2\boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{y})$ gives $-2\mathbf{X}^\top\mathbf{y}$; the quadratic term $\boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X})\boldsymbol{\beta}$ gives $2\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}$ (since $\mathbf{X}^\top\mathbf{X}$ is symmetric). Hence

$$\nabla J = -2\mathbf{X}^\top\mathbf{y} + 2\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Setting $\nabla J = \mathbf{0}$ reproduces the normal equations $\mathbf{X}^\top \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^\top \mathbf{y}$ — now derived a second way, by calculus rather than geometry, and the two routes agree. The Hessian $\nabla^2 J = 2\mathbf{X}^\top \mathbf{X} \succeq 0$ confirms $J$ is convex, so this stationary point is the global minimum.

### 12.3 Matrix-derivative identities for likelihoods

For Gaussian likelihoods and beyond:

$$\frac{\partial}{\partial \mathbf{A}} \operatorname{Tr}(\mathbf{AB}) = \mathbf{B}^\top, \qquad \frac{\partial}{\partial \mathbf{A}} \log\det \mathbf{A} = \mathbf{A}^{-\top}, \qquad \frac{\partial}{\partial \mathbf{A}} \operatorname{Tr}(\mathbf{A}^{-1}\mathbf{B}) = -\mathbf{A}^{-\top}\mathbf{B}^\top \mathbf{A}^{-\top}.$$

The log det identity is the one that appears when maximizing a multivariate Gaussian log-likelihood over a covariance matrix (the log-likelihood contains $-\frac{1}{2}\log\det\boldsymbol{\Sigma}$); setting its derivative to zero is how you derive that the MLE of the covariance is the sample covariance.

## 13 Capstone: Tying It All to Machine Learning

Every thread above converges on a handful of ML workhorses. To make the unity explicit:

- **Linear/ridge regression** is orthogonal projection (hat matrix) plus, in the ridge case, a spectral shrinkage $\sigma_i^2/(\sigma_i^2 + \lambda)$ of the SVD — conditioning, projection, and regularization in one object.
- **PCA** is the spectral theorem applied to the covariance matrix (or equivalently the SVD of the centered data): principal components are eigenvectors, explained variances are eigenvalues, and the optimality of keeping the top $k$ is Eckart–Young.
- **Gaussian models** (GMM, GP, Kalman) rely on the multivariate Gaussian, whose every operation — sampling, conditioning, marginalizing — is Cholesky factorization and Schur complements of covariance matrices.
- **SVMs and kernel methods** live or die by positive semidefiniteness (Mercer's condition), which guarantees a feature space exists and the optimization is convex.
- **Optimization** of all these is governed by the Hessian's eigenvalues: the condition number sets gradient-descent speed, and positive definiteness guarantees a unique global optimum.
- **Neural networks** are compositions of affine maps; their forward pass is matrix multiplication, their backward pass is multiplication by transposes (the adjoint identity), and their training stability is again an eigenvalue/conditioning story.

The recurring moral: *eigenvalues and singular values are the universal currency.* They measure variance (PCA), curvature (optimization), stretch (conditioning), and importance (low-rank approximation). Mastering how a matrix acts on its eigen/singular directions is mastering the linear algebra of machine learning.

## 14 Gram–Schmidt and the Construction of QR

We have used orthonormal bases freely; here is how to *build* one from any basis, and why the construction yields the QR factorization.

### 14.1 The Gram–Schmidt process

Given independent vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$, we produce orthonormal $\mathbf{q}_1, \dots, \mathbf{q}_n$ with the same span at every stage. The idea: take each new vector and subtract off its components along the directions already fixed, leaving only the genuinely new part, then normalize.

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{a}_1, & \mathbf{q}_1 &= \mathbf{u}_1 / \|\mathbf{u}_1\|; \\ \mathbf{u}_2 &= \mathbf{a}_2 - \langle \mathbf{a}_2, \mathbf{q}_1 \rangle \mathbf{q}_1, & \mathbf{q}_2 &= \mathbf{u}_2 / \|\mathbf{u}_2\|; \\ \mathbf{u}_k &= \mathbf{a}_k - \sum_{j<k} \langle \mathbf{a}_k, \mathbf{q}_j \rangle \mathbf{q}_j, & \mathbf{q}_k &= \mathbf{u}_k / \|\mathbf{u}_k\|. \end{aligned}$$

The subtracted sum $\sum_{j<k} \langle \mathbf{a}_k, \mathbf{q}_j \rangle \mathbf{q}_j$ is exactly the projection of $\mathbf{a}_k$ onto the space already built; removing it guarantees $\mathbf{u}_k \perp \mathbf{q}_j$ for all $j < k$ (check: $\langle \mathbf{u}_k, \mathbf{q}_i \rangle = \langle \mathbf{a}_k, \mathbf{q}_i \rangle - \langle \mathbf{a}_k, \mathbf{q}_i \rangle = 0$). This is the orthogonality principle from least squares, applied recursively.

**Worked Example**

Orthonormalize $\mathbf{a}_1 = (1,1,0)$, $\mathbf{a}_2 = (1,0,1)$. Step 1: $\mathbf{q}_1 = (1,1,0)/\sqrt{2}$. Step 2: $\langle \mathbf{a}_2, \mathbf{q}_1 \rangle = (1+0+0)/\sqrt{2} = 1/\sqrt{2}$, so $\mathbf{u}_2 = (1,0,1) - \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}(1,1,0) = (1,0,1) - (\tfrac{1}{2}, \tfrac{1}{2}, 0) = (\tfrac{1}{2}, -\tfrac{1}{2}, 1)$. Normalize: $\|\mathbf{u}_2\| = \sqrt{\frac{1}{4} + \frac{1}{4} + 1} = \sqrt{3/2}$, so $\mathbf{q}_2 = (\tfrac{1}{2}, -\tfrac{1}{2}, 1)/\sqrt{3/2}$. Verify $\langle \mathbf{q}_1, \mathbf{q}_2 \rangle = 0$: $(1)(\frac{1}{2}) + (1)(-\frac{1}{2}) + (0)(1) = 0$. Orthogonal, as designed.

### 14.2 Why this is QR

Rearrange the Gram–Schmidt relations to express each $\mathbf{a}_k$ in terms of the $\mathbf{q}_j$: $\mathbf{a}_k = \sum_{j \le k} r_{jk} \mathbf{q}_j$ where $r_{jk} = \langle \mathbf{a}_k, \mathbf{q}_j \rangle$ for $j < k$ and $r_{kk} = \|\mathbf{u}_k\|$. In matrix form, with $\mathbf{a}_k$ as columns of $\mathbf{X}$ and $\mathbf{q}_j$ as columns of $\mathbf{Q}$, this is $\mathbf{X} = \mathbf{QR}$ with $\mathbf{R}$ upper-triangular (because $\mathbf{a}_k$ uses only $\mathbf{q}_1, \dots, \mathbf{q}_k$). That triangularity is the whole point: it makes the least-squares system $\mathbf{R}\boldsymbol{\beta} = \mathbf{Q}^\top \mathbf{y}$ solvable by back-substitution, stably, at condition number $\kappa(\mathbf{X})$. In practice one uses the numerically superior *Householder* reflections rather than classical Gram–Schmidt (which loses orthogonality under rounding), but the factorization is the same.

## 15 Determinants, Properly Understood

The determinant is often taught as a cofactor-expansion recipe; the useful definition is geometric.

### 15.1 The determinant as signed volume

$\det \mathbf{A}$ is the signed volume of the parallelepiped spanned by the columns of $\mathbf{A}$ (equivalently, the factor by which $\mathbf{A}$ scales all volumes). This single idea explains every property:

- $\det \mathbf{A} = 0 \iff$ **$\mathbf{A}$ singular**: the columns are dependent, so the parallelepiped is flat (zero volume), so $\mathbf{A}$ collapses a dimension.
- $\det(\mathbf{AB}) = \det \mathbf{A} \det \mathbf{B}$: composing two maps multiplies their volume-scaling factors.
- $\det(\mathbf{A}^{-1}) = 1/\det \mathbf{A}$: undoing a map inverts its scaling.
- **Row swap flips the sign**: orientation reverses (the "signed" in signed volume).
- $\det = \prod \lambda_i$: each eigen-direction stretches volume by $\lambda_i$; the total is the product.

**Intuition**

In probability, the determinant is the Jacobian factor in the change-of-variables formula: when you transform a random vector by $\mathbf{A}$, densities scale by $1/|\det \mathbf{A}|$ because volume scales by

$|\det \mathbf{A}|$. This is why the multivariate Gaussian's normalizing constant contains $(\det \boldsymbol{\Sigma})^{-1/2}$, and why normalizing flows track $\log|\det \boldsymbol{J}|$ of their transformations. The "volume" interpretation is not an analogy — it is literally the probability mass bookkeeping.

## 16 Change of Basis: the Same Object in Different Coordinates

A vector exists independently of coordinates; its coordinate *list* depends on the chosen basis. If $\mathbf{P}$ has the new basis vectors as columns, then old coordinates $\mathbf{x}$ and new coordinates $\mathbf{x}'$ relate by $\mathbf{x} = \mathbf{P}\mathbf{x}'$, i.e. $\mathbf{x}' = \mathbf{P}^{-1}\mathbf{x}$. A linear map represented by $\mathbf{A}$ in the old basis is represented by $\mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ in the new one — a **similarity transformation**. Diagonalization $\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$ is precisely the statement "in the eigenbasis ($\mathbf{P} = \mathbf{V}$), $\mathbf{A}$ looks diagonal." The spectral theorem is the special case where the change of basis is orthogonal ($\mathbf{P} = \mathbf{Q}$, $\mathbf{P}^{-1} = \mathbf{Q}^\top$), so it preserves lengths and angles — the most benign possible change of coordinates.

**Intuition**

Eigenvalues, trace, determinant, and rank are *invariant* under change of basis (similarity) — they are intrinsic properties of the linear map, not artifacts of coordinates. This is why they are the quantities that show up in every theorem: they describe what the transformation *does*, independent of how you choose to write it down. PCA can be seen as choosing the coordinate system (the eigenbasis of the covariance) in which the data's second-order structure is diagonal — uncorrelated, axis-aligned.

## 17 Worked Computations: Eigendecomposition and SVD by Hand

The theorems become real only when you grind through the arithmetic once. This section computes a full eigendecomposition and a full SVD on small matrices, by hand, so every abstract quantity — eigenvalues, eigenvectors, singular values, the four subspaces — becomes a concrete number.

### 17.1 A complete eigendecomposition

Take the symmetric matrix

$$\mathbf{A} = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}.$$

**Step 1 — characteristic polynomial.** We solve $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$:

$$\det \begin{pmatrix} 4-\lambda & 1 \\ 1 & 4-\lambda \end{pmatrix} = (4-\lambda)^2 - 1 = \lambda^2 - 8\lambda + 15 = (\lambda - 3)(\lambda - 5) = 0.$$

So the eigenvalues are $\lambda_1 = 5$ and $\lambda_2 = 3$. Both are positive, so $\mathbf{A}$ is positive definite (a fact we could also check by Sylvester's criterion: leading minors $4 > 0$ and $\det \mathbf{A} = 15 > 0$).

**Step 2 — eigenvectors.** For $\lambda_1 = 5$, solve $(\mathbf{A} - 5\mathbf{I})\mathbf{v} = \mathbf{0}$:

$$\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \mathbf{v} = \mathbf{0} \;\Rightarrow\; v_1 = v_2 \;\Rightarrow\; \mathbf{v}_1 = \tfrac{1}{\sqrt{2}}(1, 1).$$

For $\lambda_2 = 3$, solve $(\mathbf{A} - 3\mathbf{I})\mathbf{v} = \mathbf{0}$:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \mathbf{v} = \mathbf{0} \;\Rightarrow\; v_1 = -v_2 \;\Rightarrow\; \mathbf{v}_2 = \tfrac{1}{\sqrt{2}}(1, -1).$$

The two eigenvectors are orthogonal ($\mathbf{v}_1^\top \mathbf{v}_2 = \frac{1}{2}(1-1) = 0$), exactly as the Spectral Theorem guarantees for a symmetric matrix. We have normalized them to unit length.

**Step 3 — assemble the decomposition.** With $\mathbf{Q} = [\mathbf{v}_1\ \mathbf{v}_2]$ and $\mathbf{\Lambda} = \mathrm{diag}(5, 3)$,

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top = \tfrac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 5 & 0 \\ 0 & 3 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Multiplying back out verifies the result: the $(1,1)$ entry is $\frac{1}{2}(5 \cdot 1 + 3 \cdot 1) = 4$ and the $(1,2)$ entry is $\frac{1}{2}(5 \cdot 1 + 3 \cdot (-1)) = 1$, recovering $\mathbf{A}$. The eigenbasis $\{\mathbf{v}_1, \mathbf{v}_2\}$ is the 45°-rotated coordinate system in which $\mathbf{A}$ acts by pure scaling: stretch by 5 along the $(1,1)$ diagonal, by 3 along the $(1,-1)$ anti-diagonal.

**Intuition**

The worked example exposes what diagonalization *means*: in the standard basis $\mathbf{A}$ mixes the coordinates (the off-diagonal 1's couple $x_1$ and $x_2$), but in the eigenbasis it decouples into independent scalings. The eigenvectors are the directions $\mathbf{A}$ does not rotate, only stretches, and the eigenvalues are the stretch factors. For a covariance matrix this is the principal-component decomposition; for the Hessian of a loss it is the set of independent curvature directions that govern how gradient descent converges (the conditioning story of the optimization note). Once you have done the arithmetic once, the abstract statement "$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$" carries the full geometric picture.

## 17.2 A complete SVD

Now a non-square, non-symmetric matrix:

$$\mathbf{X} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (3 \times 2).$$

The SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is computed via the eigendecomposition of $\mathbf{X}^\top\mathbf{X}$ (a $2 \times 2$ matrix, easier than the $3 \times 3$ $\mathbf{X}\mathbf{X}^\top$).

**Step 1 — form $\mathbf{X}^\top\mathbf{X}$.**

$$\mathbf{X}^\top\mathbf{X} = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix}.$$

This is already diagonal, so its eigenvalues are 5 and 1 with eigenvectors $\mathbf{e}_1 = (1,0)$ and $\mathbf{e}_2 = (0,1)$.

**Step 2 — singular values.** The singular values are the square roots of these eigenvalues: $\sigma_1 = \sqrt{5} \approx 2.236$, $\sigma_2 = \sqrt{1} = 1$. The right singular vectors are the eigenvectors of $\mathbf{X}^\top\mathbf{X}$: $\mathbf{v}_1 = (1,0)$, $\mathbf{v}_2 = (0,1)$, so $\mathbf{V} = \mathbf{I}$.

**Step 3 — left singular vectors.** Use $\mathbf{u}_i = \frac{1}{\sigma_i}\mathbf{X}\mathbf{v}_i$:

$$\mathbf{u}_1 = \tfrac{1}{\sqrt{5}}\mathbf{X}\mathbf{e}_1 = \tfrac{1}{\sqrt{5}}(2,0,1), \qquad \mathbf{u}_2 = \tfrac{1}{1}\mathbf{X}\mathbf{e}_2 = (0,1,0).$$

Check $\|\mathbf{u}_1\| = \frac{1}{\sqrt{5}}\sqrt{4+0+1} = 1$ ✓, and $\mathbf{u}_1^\top\mathbf{u}_2 = 0$ ✓. The decomposition is

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad \mathbf{U} = \begin{pmatrix} 2/\sqrt{5} & 0 \\ 0 & 1 \\ 1/\sqrt{5} & 0 \end{pmatrix}, \ \mathbf{\Sigma} = \begin{pmatrix} \sqrt{5} & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \ \mathbf{V} = \mathbf{I}.$$

**Step 4 — read off the four subspaces.** The column space of $\mathbf{X}$ is spanned by $\mathbf{u}_1, \mathbf{u}_2$ (the left singular vectors with nonzero $\sigma$), a 2-dimensional plane in $\mathbb{R}^3$. The left null space is spanned by

the remaining direction orthogonal to both, $\mathbf{u}_3 \propto (1, 0, -2)$ (check: orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$). The row space is all of $\mathbb{R}^2$ (both $\sigma > 0$), and the null space is $\{\mathbf{0}\}$ — so $\mathbf{X}$ has full column rank 2 and the least-squares solution is unique. The SVD has handed us the rank, all four subspaces, and the conditioning ($\kappa = \sigma_1/\sigma_2 = \sqrt{5} \approx 2.24$, well-conditioned) in one computation.

**Intuition**

The SVD is the master decomposition because it works for *any* matrix — rectangular, rank-deficient, non-symmetric — and lays bare everything at once: singular values (the intrinsic "gains" of the map), the orthonormal input directions $\mathbf{v}_i$ and their images $\mathbf{u}_i$, the rank (count of nonzero $\sigma$), the four fundamental subspaces (spanned by appropriate $\mathbf{u}$'s and $\mathbf{v}$'s), and the condition number $\sigma_{\max}/\sigma_{\min}$. In ML it is the engine behind PCA (the $\mathbf{v}_i$ are principal directions, $\sigma_i^2/(n-1)$ the variances), behind least-squares solving (the pseudoinverse), behind low-rank approximation and compression (Eckart–Young), and behind the numerical diagnosis of ill-conditioned regressions. Doing one by hand demystifies all of these at once.

## 17.3 Power iteration: how eigenvectors are actually found

For large matrices we do not solve the characteristic polynomial (a numerically catastrophic approach beyond tiny sizes); we iterate. **Power iteration** computes the dominant eigenvector by repeatedly multiplying and normalizing:

$$\mathbf{x}_{k+1} = \frac{\mathbf{A}\mathbf{x}_k}{\|\mathbf{A}\mathbf{x}_k\|}.$$

Why it works: write the start vector in the eigenbasis, $\mathbf{x}_0 = \sum_i c_i \mathbf{v}_i$. Then $\mathbf{A}^k\mathbf{x}_0 = \sum_i c_i \lambda_i^k \mathbf{v}_i$. Factoring out the largest eigenvalue $\lambda_1$,

$$\mathbf{A}^k\mathbf{x}_0 = \lambda_1^k\Big(c_1\mathbf{v}_1 + \sum_{i\geq 2} c_i(\lambda_i/\lambda_1)^k\mathbf{v}_i\Big).$$

Since $|\lambda_i/\lambda_1| < 1$ for $i \geq 2$, every term except the first decays geometrically, so the iterate aligns with $\mathbf{v}_1$. The convergence rate is governed by the ratio $|\lambda_2/\lambda_1|$ — the *eigengap*: a large gap means fast convergence, a small gap means slow.

**Worked iteration** on $\mathbf{A} = \left(\begin{smallmatrix} 4 & 1 \\ 1 & 4 \end{smallmatrix}\right)$ (eigenvalues $5, 3$, ratio $0.6$), starting from $\mathbf{x}_0 = (1, 0)$:

$$\mathbf{A}\mathbf{x}_0 = (4, 1),\ \text{normalize} \to (0.970, 0.243); \quad \mathbf{A}(0.970, 0.243) = (4.12, 2.18) \to (0.884, 0.468);$$

$$\mathbf{A}(0.884, 0.468) = (4.00, 2.76) \to (0.823, 0.568); \quad \cdots \to (0.707, 0.707) = \mathbf{v}_1.$$

After a handful of steps the iterate converges to $\frac{1}{\sqrt{2}}(1, 1)$, the dominant eigenvector found earlier — and the ratio of successive $\|\mathbf{A}\mathbf{x}_k\|$ values converges to $\lambda_1 = 5$. This is essentially how Google's original PageRank found the dominant eigenvector of the web's link matrix, and how iterative PCA solvers find top components without forming the full eigendecomposition.

**Intuition**

Power iteration explains how eigenvectors are found at scale and why the *eigengap* matters far beyond this algorithm. A large gap $|\lambda_1| \gg |\lambda_2|$ means one direction dominates — a clear top principal component, fast power-iteration convergence, a well-separated cluster structure in spectral clustering (the Fiedler-vector story of the unsupervised note). A small gap means the top directions are nearly tied — ambiguous principal components, slow convergence, no clear number of clusters. The same spectral quantity thus governs an algorithm's speed, a model's identifiability, and the stability of a decomposition. It is one of the most reused ideas in applied linear algebra.

# 18 Least Squares, Worked Three Ways

The normal equations, the projection picture, and the QR route all solve the same least-squares problem; computing one small example through each cements how they relate.

## 18.1 The problem

Fit a line $y = \beta_0 + \beta_1 x$ to the three points $(1,1), (2,2), (3,2)$. The design matrix and target are

$$\mathbf{X} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}.$$

There is no exact solution (the points are not collinear), so we minimize $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$.

## 18.2 Route 1 — the normal equations

Form $\mathbf{X}^\top\mathbf{X}$ and $\mathbf{X}^\top\mathbf{y}$:

$$\mathbf{X}^\top\mathbf{X} = \begin{pmatrix} 3 & 6 \\ 6 & 14 \end{pmatrix}, \qquad \mathbf{X}^\top\mathbf{y} = \begin{pmatrix} 5 \\ 11 \end{pmatrix}.$$

Solve $\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$. The inverse is $\frac{1}{\det}\left(\begin{smallmatrix} 14 & -6 \\ -6 & 3 \end{smallmatrix}\right)$ with $\det = 3 \cdot 14 - 36 = 6$, so

$$\hat{\boldsymbol{\beta}} = \tfrac{1}{6}\begin{pmatrix} 14 & -6 \\ -6 & 3 \end{pmatrix}\begin{pmatrix} 5 \\ 11 \end{pmatrix} = \tfrac{1}{6}\begin{pmatrix} 70 - 66 \\ -30 + 33 \end{pmatrix} = \tfrac{1}{6}\begin{pmatrix} 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 2/3 \\ 1/2 \end{pmatrix}.$$

So the fitted line is $\hat{y} = \frac{2}{3} + \frac{1}{2}x$.

## 18.3 Route 2 — the projection picture

The fitted values are $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$: at $x = 1, 2, 3$ these are $\frac{2}{3} + \frac{1}{2} = \frac{7}{6} \approx 1.17$, $\frac{2}{3} + 1 = \frac{5}{3} \approx 1.67$, $\frac{2}{3} + \frac{3}{2} = \frac{13}{6} \approx 2.17$. The residual vector is $\mathbf{y} - \hat{\mathbf{y}} = (1 - 1.17,\ 2 - 1.67,\ 2 - 2.17) = (-0.17, 0.33, -0.17)$. **Check orthogonality**: the residual must be orthogonal to the column space of $\mathbf{X}$, i.e. to both columns. Against the all-ones column: $-0.17 + 0.33 - 0.17 = -0.01 \approx 0$ ✓ (rounding). Against $(1, 2, 3)$: $-0.17 + 0.67 - 0.50 = 0 \approx 0$ ✓. The residual is orthogonal to the fitted plane — the geometric content of least squares: $\hat{\mathbf{y}}$ is the orthogonal projection of $\mathbf{y}$ onto $\mathrm{col}(\mathbf{X})$, and the normal equations $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{0}$ are precisely the statement that the residual is orthogonal to every column.

## 18.4 Route 3 — QR (sketch)

Gram–Schmidt on the columns of $\mathbf{X}$ produces $\mathbf{X} = \mathbf{QR}$ with $\mathbf{Q}$ orthonormal. Then $\hat{\boldsymbol{\beta}} = \mathbf{R}^{-1}\mathbf{Q}^\top\mathbf{y}$, solved by back-substitution on the triangular $\mathbf{R}$ — no $\mathbf{X}^\top\mathbf{X}$ ever formed. This matters numerically: forming $\mathbf{X}^\top\mathbf{X}$ *squares* the condition number ($\kappa(\mathbf{X}^\top\mathbf{X}) = \kappa(\mathbf{X})^2$), so an already ill-conditioned design becomes far worse in the normal equations. QR works directly on $\mathbf{X}$, preserving conditioning, which is why production least-squares solvers (and `numpy.linalg.lstsq`) use QR or SVD, never the textbook normal equations.

**Intuition**

The three routes are mathematically identical but numerically and conceptually distinct. The normal equations are the cleanest by hand and reveal the algebra. The projection picture reveals the geometry — least squares is dropping a perpendicular onto the column space, and

the orthogonal residual is the defining property. QR (and SVD) are what you actually run, because they avoid squaring the condition number. A practitioner who holds all three in mind reads a regression correctly: the coefficients (algebra), the residual orthogonality and leverage (geometry), and the numerical reliability (conditioning). The same triangle — algebra, geometry, numerics — recurs for every matrix computation in machine learning.

# 19 Worked Gram–Schmidt and the QR Factorization

We run Gram–Schmidt on a concrete matrix, producing the orthonormal basis and the $\mathbf{R}$ factor, so the abstract orthogonalization becomes mechanical.

## 19.1 The process on two vectors

Take the columns $\mathbf{a}_1 = (1,1,0)$ and $\mathbf{a}_2 = (1,0,1)$. Gram–Schmidt orthogonalizes them.

**First vector**: normalize $\mathbf{a}_1$. $\|\mathbf{a}_1\| = \sqrt{2}$, so $\mathbf{q}_1 = \frac{1}{\sqrt{2}}(1,1,0)$.

**Second vector**: subtract the component of $\mathbf{a}_2$ along $\mathbf{q}_1$, then normalize. The projection coefficient is $\mathbf{q}_1^\top \mathbf{a}_2 = \frac{1}{\sqrt{2}}(1+0+0) = \frac{1}{\sqrt{2}}$. The orthogonalized vector is

$$\mathbf{u}_2 = \mathbf{a}_2 - (\mathbf{q}_1^\top \mathbf{a}_2)\mathbf{q}_1 = (1,0,1) - \tfrac{1}{\sqrt{2}} \cdot \tfrac{1}{\sqrt{2}}(1,1,0) = (1,0,1) - (\tfrac{1}{2},\tfrac{1}{2},0) = (\tfrac{1}{2},-\tfrac{1}{2},1).$$

Check orthogonality: $\mathbf{q}_1^\top \mathbf{u}_2 = \frac{1}{\sqrt{2}}(\frac{1}{2} - \frac{1}{2} + 0) = 0$ ✓. Normalize: $\|\mathbf{u}_2\| = \sqrt{\frac{1}{4}+\frac{1}{4}+1} = \sqrt{3/2}$, so $\mathbf{q}_2 = \sqrt{2/3}(\frac{1}{2},-\frac{1}{2},1)$.

## 19.2 Reading off R

The QR factorization $\mathbf{A} = \mathbf{QR}$ has $\mathbf{R}$ upper-triangular with entries $R_{ij} = \mathbf{q}_i^\top \mathbf{a}_j$. Here

$$\mathbf{R} = \begin{pmatrix} \mathbf{q}_1^\top \mathbf{a}_1 & \mathbf{q}_1^\top \mathbf{a}_2 \\ 0 & \mathbf{q}_2^\top \mathbf{a}_2 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & 1/\sqrt{2} \\ 0 & \sqrt{3/2} \end{pmatrix}.$$

The diagonal entries of $\mathbf{R}$ are the lengths of the successive orthogonalized vectors — and if any were zero, that column was linearly dependent on the earlier ones (the test for rank deficiency the algorithm performs automatically). The factorization expresses each original column as a combination of the orthonormal $\mathbf{q}$'s with the triangular coefficients in $\mathbf{R}$.

**Intuition**

QR is Gram–Schmidt bookkept as a matrix product, and it is the numerically stable workhorse for least squares, eigenvalue algorithms (the QR algorithm iterates QR factorizations to find all eigenvalues), and orthogonalization throughout scientific computing. The key insight from the worked example: orthogonalization is repeated "subtract off what you can already explain, keep the remainder." That same move — regress out the explained part, work with the residual — is the Frisch–Waugh–Lovell theorem in regression, the deflation step in PCA, and the conjugate-gradient method in optimization. Gram–Schmidt is where a student first meets it, and recognizing it later in disguise is a mark of fluency.

# 20 Positive Definiteness, Worked and Tested

Positive (semi)definiteness is the single most important matrix property in ML — it certifies that a covariance is valid, a quadratic loss is convex, a kernel is legitimate, and a Newton step is a descent

direction. We work the tests concretely.

### 20.1 Three equivalent tests

A symmetric $\mathbf{A}$ is positive definite if and only if any one of these holds: (1) all eigenvalues are positive; (2) all leading principal minors are positive (Sylvester's criterion); (3) it has a Cholesky factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ with positive diagonal. We test

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

**Eigenvalue test**: $\det(\mathbf{A} - \lambda\mathbf{I}) = (2-\lambda)^2 - 1 = \lambda^2 - 4\lambda + 3 = (\lambda - 1)(\lambda - 3)$, eigenvalues $1, 3$ — both positive, so PD. **Sylvester**: leading minors are $2 > 0$ and $\det \mathbf{A} = 4 - 1 = 3 > 0$ — PD. **Cholesky**: seek $\mathbf{L} = \left(\begin{smallmatrix} \ell_{11} & 0 \\ \ell_{21} & \ell_{22} \end{smallmatrix}\right)$ with $\mathbf{L}\mathbf{L}^\top = \mathbf{A}$. Matching entries: $\ell_{11}^2 = 2 \Rightarrow \ell_{11} = \sqrt{2}$; $\ell_{21}\ell_{11} = 1 \Rightarrow \ell_{21} = 1/\sqrt{2}$; $\ell_{21}^2 + \ell_{22}^2 = 2 \Rightarrow \ell_{22}^2 = 2 - \frac{1}{2} = \frac{3}{2} \Rightarrow \ell_{22} = \sqrt{3/2}$. The positive diagonal confirms PD. All three tests agree, as they must.

### 20.2 A semidefinite and an indefinite case

For $\mathbf{B} = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$: eigenvalues 0 and 2 — positive *semi*definite (one zero eigenvalue, a flat direction $(1, -1)$ along which the quadratic form is zero). This is the signature of a rank-deficient covariance (perfectly collinear data) or a convex-but-not-strictly-convex loss (a flat valley). For $\mathbf{C} = \left(\begin{smallmatrix} 1 & 2 \\ 2 & 1 \end{smallmatrix}\right)$: $\det = 1 - 4 = -3 < 0$, so eigenvalues have opposite signs — indefinite, a saddle. The quadratic form $\mathbf{x}^\top\mathbf{C}\mathbf{x}$ goes up in some directions and down in others, exactly the saddle-point geometry that complicates non-convex optimization.

**Intuition**

The sign of a quadratic form is the hinge between the tractable and the hard. Positive definite means a strictly convex bowl — a unique minimum, a valid covariance, a legitimate kernel, a guaranteed-descent Newton step. Positive semidefinite means a bowl with flat directions — convex but with non-unique minima, the signature of collinearity or redundancy, cured by regularization (which adds $\lambda\mathbf{I}$, lifting the zero eigenvalues to $\lambda > 0$ and restoring strict definiteness). Indefinite means a saddle — the central difficulty of non-convex loss landscapes. Cholesky is the practical test (it succeeds exactly for PD matrices and is cheap), and it doubles as the way to sample correlated Gaussians and to whiten data. One concept, threaded through covariance, convexity, kernels, and optimization.

## 21 Matrix Calculus: the Identities You Will Reuse Forever

Machine learning is gradient-driven, and most gradients are taken with respect to vectors and matrices. A small set of identities, derived once, covers almost everything.

### 21.1 The core gradients, derived

For a constant vector $\mathbf{a}$ and matrix $\mathbf{A}$ (symmetric where noted):

- $\nabla_{\mathbf{x}}(\mathbf{a}^\top\mathbf{x}) = \mathbf{a}$. Each component $\partial(\sum_j a_j x_j)/\partial x_i = a_i$, assembling to $\mathbf{a}$.
- $\nabla_{\mathbf{x}}(\mathbf{x}^\top\mathbf{A}\mathbf{x}) = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}$, which is $2\mathbf{A}\mathbf{x}$ for symmetric $\mathbf{A}$. Derivation: $\mathbf{x}^\top\mathbf{A}\mathbf{x} = \sum_{j,k} A_{jk}x_j x_k$; differentiating w.r.t. $x_i$ gives $\sum_k A_{ik}x_k + \sum_j A_{ji}x_j = (\mathbf{A}\mathbf{x})_i + (\mathbf{A}^\top\mathbf{x})_i$.
- $\nabla_{\mathbf{x}} \|\mathbf{x}\|^2 = 2\mathbf{x}$ (the special case $\mathbf{A} = \mathbf{I}$).

### 21.2 The least-squares gradient in one line

With these, the least-squares gradient falls out immediately. Expand $\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 = \mathbf{y}^\top\mathbf{y} - 2\boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y} + \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X})\boldsymbol{\beta}$. Differentiate term by term using the identities ($\mathbf{X}^\top\mathbf{X}$ is symmetric):

$$\nabla_{\boldsymbol{\beta}} = -2\mathbf{X}^\top\mathbf{y} + 2(\mathbf{X}^\top\mathbf{X})\boldsymbol{\beta}.$$

Setting to zero gives the normal equations $\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$ — the entire derivation of the OLS estimator in three lines, powered by two identities.

### 21.3 Identities for likelihoods and the log-determinant

Gaussian likelihoods need derivatives through a determinant and an inverse:

- $\nabla_{\mathbf{A}} \log\det\mathbf{A} = \mathbf{A}^{-\top}$ (for invertible $\mathbf{A}$); for symmetric $\mathbf{A}$ this is $\mathbf{A}^{-1}$.
- $\nabla_{\mathbf{A}} \operatorname{Tr}(\mathbf{A}^{-1}\mathbf{B}) = -\mathbf{A}^{-\top}\mathbf{B}^\top\mathbf{A}^{-\top}$.

These are exactly what you differentiate when fitting a Gaussian's covariance by maximum likelihood: the $\log\det\boldsymbol{\Sigma}$ term (the normalizer) and the $\operatorname{Tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S})$ term (the data fit) produce, on setting the gradient to zero, the MLE $\hat{\boldsymbol{\Sigma}} = \mathbf{S}$ (the sample covariance) — the multivariate Gaussian result of the probability note, derived through these matrix-calculus identities.

**Intuition**

Matrix calculus is the notation that turns pages of index-juggling into one-line derivations, and the handful of identities above cover the overwhelming majority of ML gradients: linear terms, quadratic forms (every squared loss and every quadratic regularizer), and the log-det/trace pair (every Gaussian likelihood). The derivations all reduce to "differentiate the scalar sum, then reassemble into vector/matrix form." Mastering these is what lets you derive the OLS estimator, the ridge solution, the Gaussian MLE, the Newton step, and the backpropagation equations without fear — and lets you check an autodiff library's output when something looks wrong.

## 22 Vectorization and the Kronecker Product

Some ML computations — covariance of vectorized parameters, certain layer gradients, Lyapunov equations in control — need the algebra of *stacking* a matrix into a vector. The **vectorization** operator $\operatorname{vec}(\mathbf{A})$ stacks columns; the **Kronecker product** $\mathbf{A}\otimes\mathbf{B}$ is the block matrix with blocks $A_{ij}\mathbf{B}$. The key identity linking them is

$$\operatorname{vec}(\mathbf{AXB}) = (\mathbf{B}^\top\otimes\mathbf{A})\,\operatorname{vec}(\mathbf{X}),$$

which converts a "sandwich" matrix product into a single matrix-times-vector — the move that lets you take a gradient with respect to a matrix-shaped parameter by treating it as a long vector. Eigenvalues of $\mathbf{A}\otimes\mathbf{B}$ are all products $\lambda_i(\mathbf{A})\lambda_j(\mathbf{B})$, and $\det(\mathbf{A}\otimes\mathbf{B}) = \det(\mathbf{A})^m\det(\mathbf{B})^n$ for appropriate sizes — properties that make the Kronecker structure computationally cheap to exploit when it appears (e.g. in the Kronecker-factored approximations used to scale second-order optimization to neural networks).

**Intuition**

Vectorization and the Kronecker product are the bridge between matrix-shaped objects and the vector-shaped world that gradients and covariances live in. You meet them when a parameter is naturally a matrix (a weight layer, a covariance) but the optimization or uncertainty calculus wants a vector. The headline identity $\operatorname{vec}(\mathbf{AXB}) = (\mathbf{B}^\top\otimes\mathbf{A})\operatorname{vec}(\mathbf{X})$ is the workhorse, and the cheap eigenvalue/determinant structure of Kronecker products is what makes large-scale

Kronecker-factored methods (like K-FAC for neural-network training) feasible. They are specialized but, when the structure is present, they turn an intractable computation into a routine one.

# 23 Determinants as Volume, Worked

The determinant is best understood as the signed volume-scaling factor of a linear map, and a worked example makes this geometric.

## 23.1 The $2 \times 2$ case

For $\mathbf{A} = \left(\begin{smallmatrix} 3 & 1 \\ 0 & 2 \end{smallmatrix}\right)$, $\det \mathbf{A} = 3 \cdot 2 - 1 \cdot 0 = 6$. Geometrically, $\mathbf{A}$ maps the unit square (corners $\mathbf{0}, (1,0), (0,1), (1,1)$) to the parallelogram with corners $\mathbf{0}, (3,0), (1,2), (4,2)$. The area of this parallelogram is the base–height product: base $(3,0)$ has length 3, and the height (the component of $(1,2)$ perpendicular to the base) is 2, giving area $6 = \det \mathbf{A}$. So the determinant is exactly the factor by which areas scale under $\mathbf{A}$. A negative determinant would mean the map also flips orientation (reflects); a zero determinant means the map collapses the square to a line (or point) of zero area — precisely the singular case where columns are linearly dependent.

## 23.2 Why the multiplicative property is obvious geometrically

$\det(\mathbf{AB}) = \det \mathbf{A} \det \mathbf{B}$ is immediate once you see determinants as volume scaling: applying $\mathbf{B}$ then $\mathbf{A}$ scales volume by $\det \mathbf{B}$ then by $\det \mathbf{A}$, so the composite scales by the product. Likewise $\det(\mathbf{A}^{-1}) = 1/\det \mathbf{A}$ (undoing the scaling), and $\det \mathbf{A} = \prod_i \lambda_i$ (the eigenvalues are the per-eigendirection scalings, and volume scales by their product). The whole algebra of determinants follows from the single geometric idea of volume scaling.

**Intuition**

Seeing the determinant as signed volume scaling unifies its otherwise-arbitrary-looking properties and connects it to ML. The log det that appears in the Gaussian log-likelihood is the log-volume of the covariance ellipsoid — a measure of the distribution's spread, which is why maximizing likelihood trades off fitting the data (the quadratic term) against not letting the covariance balloon (the log det term). The Jacobian determinant in the change-of-variables formula (used in normalizing-flow generative models) is the local volume-scaling of a transformation, correcting densities so probability is conserved. A zero determinant flags singularity — collinear data, a non-invertible map, a degenerate distribution. The geometric view turns a computational nuisance into an interpretable quantity.

# 24 Change of Basis, Worked

The same vector or operator looks different in different coordinate systems, and translating between them is the change-of-basis operation underlying PCA, diagonalization, and feature transforms.

## 24.1 A vector in two bases

The vector $\mathbf{x} = (3,1)$ in the standard basis. Express it in the basis $\mathbf{b}_1 = (1,1), \mathbf{b}_2 = (1,-1)$. We seek coefficients $c_1, c_2$ with $c_1\mathbf{b}_1 + c_2\mathbf{b}_2 = \mathbf{x}$:

$$c_1(1,1) + c_2(1,-1) = (3,1) \;\Rightarrow\; c_1 + c_2 = 3,\; c_1 - c_2 = 1 \;\Rightarrow\; c_1 = 2, c_2 = 1.$$

So in the new basis, $\mathbf{x}$ has coordinates $(2, 1)$ — the *same vector*, different numbers, because the basis vectors are different rulers. The change-of-basis matrix $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2]$ converts new coordinates to standard ($\mathbf{x} = \mathbf{Bc}$), and its inverse converts the other way ($\mathbf{c} = \mathbf{B}^{-1}\mathbf{x}$).

### 24.2 An operator in the eigenbasis

Diagonalization is change of basis applied to an operator: $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ says "in the eigenbasis $\mathbf{Q}$, the operator $\mathbf{A}$ is just the diagonal scaling $\mathbf{\Lambda}$." To apply $\mathbf{A}$ to a vector, you can either multiply directly, or: rotate into the eigenbasis ($\mathbf{Q}^\top\mathbf{x}$), scale each coordinate ($\mathbf{\Lambda}\mathbf{Q}^\top\mathbf{x}$), and rotate back ($\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top\mathbf{x}$). The eigenbasis is the coordinate system in which the operator is simplest. PCA is exactly this: rotate the data into the eigenbasis of the covariance, where the features become uncorrelated (the covariance is diagonal) and ordered by variance.

**Intuition**

Change of basis is the idea that an object's coordinates depend on the ruler you measure with, while the object itself is invariant. Diagonalization, PCA, the Fourier transform, and whitening are all "find the basis in which this is simplest": the eigenbasis (operator becomes diagonal), the principal-component basis (features become uncorrelated), the frequency basis (convolution becomes multiplication). Recognizing a method as "transform to a convenient basis, do something simple, transform back" demystifies a huge swath of applied mathematics — and it is why the eigendecomposition and SVD, which *find* those convenient bases, are the most reused tools in the subject.

## 25 Additional Worked Exercises

**Worked Example**

**Exercise.** Prove the Rank–Nullity theorem's consequence that a square matrix is invertible iff its null space is trivial.

**Solution.** For an $n \times n$ matrix $\mathbf{A}$, Rank–Nullity says $\operatorname{rank}(\mathbf{A}) + \dim\ker(\mathbf{A}) = n$. If the null space is trivial ($\dim\ker = 0$), then rank $= n$, so the columns span $\mathbb{R}^n$ — $\mathbf{A}$ is onto, and being square, also one-to-one, hence invertible. Conversely, if $\mathbf{A}$ is invertible, $\mathbf{Ax} = \mathbf{0}$ implies $\mathbf{x} = \mathbf{A}^{-1}\mathbf{0} = \mathbf{0}$, so the null space is trivial. The two conditions are equivalent. This is why "$\mathbf{X}$ has full column rank" (trivial null space) is exactly the condition for the OLS normal-equations matrix $\mathbf{X}^\top\mathbf{X}$ to be invertible and the least-squares solution to be unique — a collinear design has a nontrivial null space and a non-unique solution, the problem ridge regression repairs.

**Worked Example**

**Exercise.** Show that for any matrix $\mathbf{X}$, $\mathbf{X}^\top\mathbf{X}$ is positive semidefinite, and positive definite iff $\mathbf{X}$ has full column rank.

**Solution.** For any vector $\mathbf{v}$, $\mathbf{v}^\top(\mathbf{X}^\top\mathbf{X})\mathbf{v} = (\mathbf{Xv})^\top(\mathbf{Xv}) = \|\mathbf{Xv}\|^2 \geq 0$, so $\mathbf{X}^\top\mathbf{X}$ is PSD. It is positive *definite* iff $\|\mathbf{Xv}\|^2 > 0$ for all $\mathbf{v} \neq \mathbf{0}$, i.e. iff $\mathbf{Xv} \neq \mathbf{0}$ for all $\mathbf{v} \neq \mathbf{0}$ — exactly the condition that $\mathbf{X}$ has trivial null space, i.e. full column rank. When $\mathbf{X}$ is rank-deficient (collinear features, or $p > n$), $\mathbf{X}^\top\mathbf{X}$ is PSD but singular, with zero eigenvalues along the collinear directions — the OLS solution is non-unique. Ridge adds $\lambda\mathbf{I}$, lifting those zero eigenvalues to $\lambda$ and restoring positive definiteness and uniqueness. This single fact links rank, definiteness, invertibility, and regularization.

**Worked Example**

**Exercise.** The condition number of a matrix is $\kappa = \sigma_{\max}/\sigma_{\min}$. Explain why solving $\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$ is numerically worse than working with $\mathbf{X}$ directly.
**Solution.** The singular values of $\mathbf{X}^\top\mathbf{X}$ are the *squares* of those of $\mathbf{X}$ (since $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$ gives $\mathbf{X}^\top\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^\top$). Therefore $\kappa(\mathbf{X}^\top\mathbf{X}) = \sigma_{\max}^2/\sigma_{\min}^2 = \kappa(\mathbf{X})^2$ — forming the normal equations *squares* the condition number. A design with $\kappa(\mathbf{X}) = 10^4$ (moderately ill-conditioned) yields $\kappa(\mathbf{X}^\top\mathbf{X}) = 10^8$, near the limit of double-precision reliability, so the solved coefficients lose many digits of accuracy. QR and SVD-based solvers work directly on $\mathbf{X}$, keeping the conditioning at $\kappa(\mathbf{X})$, which is why they are the production methods. This is a concrete instance of why the algebraically-cleanest formula (the normal equations) is not the numerically-soundest algorithm — a distinction that matters whenever data are nearly collinear.

## 26 Consolidated Exercises

These reinforce the derivations; work them with the geometric picture in mind.

1. **(Subspaces)** For $\mathbf{X} = \left(\begin{smallmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 1 & 3 & 2 \end{smallmatrix}\right)$, find the rank, a basis for $\mathcal{C}(\mathbf{X})$, and a basis for $\mathcal{N}(\mathbf{X})$. Verify rank–nullity. *Hint: row-reduce; note row 3 = row 1 + row 2.*
2. **(Projection)** Show directly that $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ satisfies $\mathbf{H}\mathbf{X} = \mathbf{X}$ (it fixes the column space) and interpret geometrically.
3. **(Spectral)** For symmetric $\mathbf{A}$, prove $\|\mathbf{A}\|_2 = \max_i |\lambda_i|$ using the eigenbasis. Why does this fail for non-symmetric $\mathbf{A}$ (where instead $\|\mathbf{A}\|_2 = \sigma_{\max}$)?
4. **(PSD)** Prove that if $\mathbf{A} \succeq 0$ and $\mathbf{B} \succeq 0$ then $\mathbf{A} + \mathbf{B} \succeq 0$, but $\mathbf{AB}$ need not even be symmetric.
5. **(SVD)** Given the SVD of $\mathbf{X}$, write the SVD of $\mathbf{X}^+$ and verify $\mathbf{X}\mathbf{X}^+\mathbf{X} = \mathbf{X}$.
6. **(Conditioning)** Construct a $2 \times 2$ matrix with $\kappa = 10^4$ and exhibit a right-hand-side perturbation that changes the solution by a factor $\sim 10^4$ in relative norm.
7. **(Quant)** Show that the sample correlation matrix of $p$ asset returns is PSD, and explain what a zero eigenvalue would imply about the existence of a riskless portfolio combination.
8. **(Calculus)** Derive $\nabla_{\boldsymbol{\beta}}\left[\,\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\,\|\boldsymbol{\beta}\|_2^2\,\right]$ and solve for the ridge estimator, showing the role of $\lambda$ in conditioning.

---

*End of the Linear Algebra prerequisite. The companion documents develop Calculus & Optimization and Probability & Statistics to the same standard, and the algorithm series applies all three.*