Linear Regression & Regularization

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

Every result derived; every step justified; worked examples throughout

Abstract

Linear regression is the foundation on which most of supervised learning is built, and the cleanest setting in which to understand the ideas — estimation, the bias–variance tradeoff, regularization, inference — that recur everywhere. This document derives ordinary least squares from three independent viewpoints (geometry, calculus, and maximum likelihood) and shows they coincide; states and interprets every Gauss–Markov assumption with its diagnostic and remedy; and develops ridge, lasso, and elastic net to the point where their sparsity and shrinkage behavior is derived rather than asserted, including the spectral (SVD) view of ridge and the subgradient/KKT view of lasso. Worked numerical examples and quant-specific cautions throughout.

Contents

1 The Model and Three Ways to Read It

We observe n pairs $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, stacked into a design matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ (one row per observation, usually with a leading column of ones for the intercept) and response $\mathbf{y} \in \mathbb{R}^n$. The linear model asserts

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

with coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^p$ and error $\boldsymbol{\varepsilon} \in \mathbb{R}^n$. Three readings of this equation give three routes to the same estimator, and seeing all three is what makes the subject solid rather than memorized.

- **Geometric/algebraic:** find the $\boldsymbol{\beta}$ making $\mathbf{X}\boldsymbol{\beta}$ as close to $\mathbf{y}$ as possible in Euclidean distance. Pure linear algebra; no probability needed. (Section 2.)
- **Calculus:** minimize a differentiable loss by setting its gradient to zero. (Section 3.)
- **Probabilistic:** assume Gaussian noise and maximize the likelihood. This is what *justifies* squared error as the loss and connects regression to the rest of statistics. (Section 4.)

Intuition

The most important conceptual point in this whole document: the OLS *point estimate* requires *none* of the statistical assumptions — it is just a projection, definable for any data. The Gauss–Markov assumptions (Section 5) are needed only to claim the estimate is *good* (unbiased, minimum-variance) and to do *inference* (confidence intervals, tests). Conflating "computing $\hat{\boldsymbol{\beta}}$" with "$\hat{\boldsymbol{\beta}}$ is trustworthy" is the single most common error in applied regression. Keep them separate.

2 Ordinary Least Squares I: the Geometric Derivation

We want $\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$. As $\boldsymbol{\beta}$ ranges over $\mathbb{R}^p$, the vector $\mathbf{X}\boldsymbol{\beta}$ ranges over the column space $\mathcal{C}(\mathbf{X})$ — the set of all achievable predictions. So the problem is: *which point in the subspace $\mathcal{C}(\mathbf{X})$ is closest to $\mathbf{y}$?* The answer, from the linear-algebra note, is the orthogonal projection of $\mathbf{y}$ onto $\mathcal{C}(\mathbf{X})$, characterized by the residual being perpendicular to the subspace.

Theorem 1 (Normal equations from orthogonality)**.** *$\hat{\boldsymbol{\beta}}$ minimizes $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ iff the residual $\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}$ is orthogonal to every column of $\mathbf{X}$, i.e. $\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}$, equivalently $\mathbf{X}^\top\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^\top\mathbf{y}$.*

Proof. Let $\boldsymbol{p} = \mathbf{X}\hat{\boldsymbol{\beta}}$ be the orthogonal projection (residual $\perp$ subspace) and $\boldsymbol{q} = \mathbf{X}\boldsymbol{\beta}$ any other point in $\mathcal{C}(\mathbf{X})$. Decompose $\mathbf{y} - \boldsymbol{q} = (\mathbf{y} - \boldsymbol{p}) + (\boldsymbol{p} - \boldsymbol{q})$. The first summand is orthogonal to the subspace; the second lies in it; so they are orthogonal to each other, and Pythagoras gives $\|\mathbf{y} - \boldsymbol{q}\|^2 = \|\mathbf{y} - \boldsymbol{p}\|^2 + \|\boldsymbol{p} - \boldsymbol{q}\|^2 \geq \|\mathbf{y} - \boldsymbol{p}\|^2$, with equality only when $\boldsymbol{q} = \boldsymbol{p}$. Hence the projection minimizes the distance. The orthogonality "$\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}} \perp$ each column" is written $\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}) = \mathbf{0}$. □

When $\mathbf{X}$ has full column rank, $\mathbf{X}^\top\mathbf{X}$ is invertible (it is the Gram matrix of independent columns) and

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

Every symbol is now explained: $\mathbf{X}^\top$ enforces orthogonality to the columns; $\mathbf{X}^\top\mathbf{X}$ records all pairwise feature inner products and is invertible exactly when features are independent; its inverse "undoes" the feature correlations to isolate each coefficient.

Worked Example

Fit $y = \beta_0 + \beta_1 x$ to $(0,1),(1,2),(2,2)$. Then $\mathbf{X} = \left(\begin{smallmatrix}1&0\\1&1\\1&2\end{smallmatrix}\right)$, $\mathbf{y} = (1,2,2)^\top$. Compute $\mathbf{X}^\top\mathbf{X} = \left(\begin{smallmatrix}3&3\\3&5\end{smallmatrix}\right)$, $\mathbf{X}^\top\mathbf{y} = \left(\begin{smallmatrix}5\\6\end{smallmatrix}\right)$. Invert: $(\mathbf{X}^\top\mathbf{X})^{-1} = \frac{1}{6}\left(\begin{smallmatrix}5&-3\\-3&3\end{smallmatrix}\right)$, so $\hat{\boldsymbol{\beta}} = \frac{1}{6}\left(\begin{smallmatrix}5\cdot5-3\cdot6\\-3\cdot5+3\cdot6\end{smallmatrix}\right) = \left(\begin{smallmatrix}7/6\\1/2\end{smallmatrix}\right)$. Residuals $(-1/6, 1/3, -1/6)$ sum to zero (orthogonal to the ones column) and dot $(0,1,2)$ to zero (orthogonal to x). The geometry is exact.

2.1 The hat matrix and leverage

The fitted values are $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{H}\mathbf{y}$ with $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$, the orthogonal projector onto $\mathcal{C}(\mathbf{X})$. From the linear-algebra note, $\mathbf{H}$ is symmetric, idempotent ($\mathbf{H}^2 = \mathbf{H}$), has $\mathrm{Tr}(\mathbf{H}) = p$ (the model degrees of freedom), and eigenvalues in $\{0,1\}$. Its diagonal entry $h_{ii} = \partial\hat{y}_i/\partial y_i$ is the **leverage** of point i — how strongly its own observation determines its fit. Since $\sum_i h_{ii} = p$, leverage is a budget of p units across n points; a point with h_{ii} near 1 single-handedly controls its fitted value and can tilt the whole regression. **Cook's distance** $D_i = \frac{\hat{\varepsilon}_i^2}{p\hat{\sigma}^2}\frac{h_{ii}}{(1-h_{ii})^2}$ combines residual size and leverage to flag influential points — the formal tool behind "one crisis observation is driving my factor loadings."

3 Ordinary Least Squares II: the Calculus Derivation

The same estimator drops out of setting a gradient to zero, which also reveals convexity. Expand the loss:

$$J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = \mathbf{y}^\top\mathbf{y} - 2\boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}.$$

Using the matrix-calculus identities ($\nabla(\mathbf{a}^\top\boldsymbol{\beta}) = \mathbf{a}$, $\nabla(\boldsymbol{\beta}^\top\mathbf{M}\boldsymbol{\beta}) = 2\mathbf{M}\boldsymbol{\beta}$ for symmetric $\mathbf{M}$):

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

Setting $\nabla J = \mathbf{0}$ gives the normal equations again — the calculus and geometry routes agree, as they must. The Hessian $\nabla^2 J = 2\mathbf{X}^\top\mathbf{X} \succeq 0$ is PSD, so J is convex and the stationary point is a global minimum; it is *strictly* convex (unique minimum) iff $\mathbf{X}^\top\mathbf{X} \succ 0$, i.e. full column rank. When rank-deficient, the minimum is a flat valley along the null space of $\mathbf{X}$ — infinitely many equally-good $\hat{\boldsymbol{\beta}}$, the loss of identifiability we keep meeting.

4 Ordinary Least Squares III: the Maximum-Likelihood Derivation

Now the probabilistic reading that justifies the squared-error loss. Assume $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, i.e. $y_i \mid \mathbf{x}_i \sim \mathcal{N}(\mathbf{x}_i^\top\boldsymbol{\beta}, \sigma^2)$ independently. The log-likelihood is

$$\ell(\boldsymbol{\beta}, \sigma^2) = -\tfrac{n}{2}\log(2\pi\sigma^2) - \tfrac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2.$$

Maximizing over $\boldsymbol{\beta}$ depends only on the last term, and maximizing $-\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$ is minimizing $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$ — **OLS is the Gaussian MLE**. This is why we square errors: squared error is the negative log-likelihood of Gaussian noise. Choose a different noise model and you get a different loss (Laplace noise → absolute error / quantile regression; heavy-tailed noise → robust losses). The MLE viewpoint also seeds regularization: a prior on $\boldsymbol{\beta}$ turns MLE into MAP, and Gaussian/Laplace priors produce ridge/lasso (Section 7).

Intuition

The three derivations are not redundant — each carries different information. Geometry gives the projection picture and leverage. Calculus gives convexity and the optimization route used when

no closed form exists (logistic regression, neural nets). Maximum likelihood gives the statistical interpretation, the noise-model-to-loss correspondence, and the bridge to regularization-as-priors and to inference. A researcher who holds all three can move fluidly between "compute it," "optimize it," and "interpret it."

5 The Gauss–Markov Assumptions, Each One Justified

These assumptions are what upgrade "$\hat{\boldsymbol{\beta}}$ exists" to "$\hat{\boldsymbol{\beta}}$ is good and inference is valid." We state each, explain what it buys, what its violation costs, and how to detect and fix it.

A1. Linearity in parameters: $\mathbb{E}[\mathbf{y} \mid \mathbf{X}] = \mathbf{X}\boldsymbol{\beta}$. The model is linear in $\boldsymbol{\beta}$ — but not necessarily in the raw covariates, since polynomial, spline, and interaction terms are allowed (they are just additional columns). Violation means the functional form is wrong, biasing every coefficient.

A2. Strict exogeneity: $\mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}$. The errors have zero mean given the regressors. This is stronger than zero correlation; it forbids any systematic relationship between the error and the features. Its violation (*endogeneity*) — from omitted variables, simultaneity, or measurement error — causes **bias** that more data cannot cure. This is the assumption causal inference fights hardest to defend.

A3. Homoskedasticity: $\mathrm{Var}(\varepsilon_i \mid \mathbf{X}) = \sigma^2$, constant across observations.

A4. No autocorrelation: $\mathrm{Cov}(\varepsilon_i, \varepsilon_j \mid \mathbf{X}) = 0$ for $i \neq j$. A3 and A4 together say $\mathrm{Var}(\boldsymbol{\varepsilon} \mid \mathbf{X}) = \sigma^2\mathbf{I}$.

A5. No perfect multicollinearity: $\mathrm{rank}(\mathbf{X}) = p$. Required for $\mathbf{X}^\top\mathbf{X}$ to be invertible and $\hat{\boldsymbol{\beta}}$ unique.

A6. (For inference) Normality: $\boldsymbol{\varepsilon} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Needed for exact finite-sample t and F distributions; the CLT makes it asymptotically dispensable.

Theorem 2 (Gauss–Markov)**.** *Under A1–A5, OLS is BLUE: among all estimators that are* linear *in* $\mathbf{y}$ *and* unbiased*, it has the smallest variance (in the matrix sense* $\mathrm{Var}(\tilde{\boldsymbol{\beta}}) - \mathrm{Var}(\hat{\boldsymbol{\beta}}) \succeq 0$*).*

The proof writes any other linear unbiased estimator as $\hat{\boldsymbol{\beta}} + \boldsymbol{D}\mathbf{y}$ for some $\boldsymbol{D}$ with $\boldsymbol{D}\mathbf{X} = \mathbf{0}$ (forced by unbiasedness), and shows its variance exceeds OLS's by a PSD matrix $\sigma^2\boldsymbol{D}\boldsymbol{D}^\top$. Note three things the theorem does *not* require or claim: it does not need normality (A6); it ranges only over *linear unbiased* estimators (so a biased estimator like ridge can have lower total MSE — the door through which regularization enters); and "best" is variance, not robustness.

5.1 Sampling distribution and inference

Under A1–A6, $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1})$ exactly. The variance formula is worth dwelling on: $\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$. Writing $\mathbf{X}^\top\mathbf{X}$ via its eigendecomposition, $(\mathbf{X}^\top\mathbf{X})^{-1}$ has eigenvalues $1/\lambda_i$ — so a near-collinear feature direction (tiny λ_i, i.e. small singular value) gives an enormous coefficient variance. **This is exactly why multicollinearity makes coefficients unstable and sign-flipping.** With $\hat{\sigma}^2 = \frac{1}{n-p}\|\hat{\boldsymbol{\varepsilon}}\|^2$ (the $n-p$ divisor making it unbiased — we "used up" p degrees of freedom fitting $\boldsymbol{\beta}$), the t-statistic $\hat{\beta}_j / \widehat{\mathrm{se}}(\hat{\beta}_j) \sim t_{n-p}$ gives tests and intervals. Without normality, the CLT gives the same asymptotically.

5.2 Diagnostics and remedies (compact reference)

Violation	Detect	Consequence & fix
Heteroskedasticity	Residual-vs-fit funnel; Breusch–Pagan, White	Estimates still unbiased, but SEs wrong ⇒ invalid inference. Use White/Huber robust (sandwich) SEs; or WLS/FGLS.
Autocorrelation	Durbin–Watson; residual ACF; Ljung–Box	Endemic in time series. Newey–West (HAC) SEs; or model the dynamics (ARMA errors, GLS).
Non-normality	Q–Q plot; Jarque–Bera	Hurts only finite-sample inference; CLT rescues large n. Robust regression; transform y.
Non-linearity	Residual structure; RESET	Bias. Add polynomial/spline/interaction terms.
Multicollinearity	$\mathrm{VIF}_j = 1/(1-R_j^2)$; condition number	Inflated variance, unstable signs. Drop/combine features; PCA; **ridge**.
Endogeneity	Theory; Hausman test	Bias & inconsistency. Instrumental variables (2SLS); controls.
Influential points	Leverage h_{ii}; Cook's D_i	Distorted fit. Robust loss (Huber); winsorize; investigate.

5.3 Goodness of fit and its traps

$R^2 = 1 - \mathrm{RSS}/\mathrm{TSS}$ is the fraction of variance explained, but it *never decreases* when you add a regressor (more columns can only shrink the projection residual), so it cannot guide model selection. Adjusted $R^2 = 1 - \frac{(1-R^2)(n-1)}{n-p}$ penalizes added parameters. For genuine selection use AIC ($2p - 2\ell$, predictive, tends to over-select), BIC ($p\log n - 2\ell$, consistent for the true model, penalizes complexity harder), or cross-validated error. In finance, beware R^2 inflated by look-ahead leakage and overlapping windows.

6 Regularization: Ridge, Lasso, Elastic Net

The penalized objective ($\lambda \geq 0$):

$$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \ \tfrac{1}{2}\,\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda P(\boldsymbol{\beta}).$$

The motivation is the bias–variance MSE tradeoff: OLS is unbiased but, under multicollinearity or large p, has huge variance. Deliberately introducing bias (shrinking coefficients) can slash variance and lower total error.

Standardize first — non-negotiable

Penalties are not scale-invariant: a feature in basis points is penalized differently than the same effect in percent. Center and scale every feature to unit variance before fitting, and **never penalize the intercept** (it sets the baseline level, which scaling does not touch). Transform coefficients back to original units afterward.

6.1 Ridge (L2): $P = \frac{1}{2}\,\|\boldsymbol{\beta}\|_2^2$

The penalized normal equations are $(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\hat{\boldsymbol{\beta}} = \mathbf{X}^\top\mathbf{y}$ (differentiate $\frac{1}{2}\,\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{\lambda}{2}\,\|\boldsymbol{\beta}\|^2$ and set to zero), giving the **closed form**

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}.$$

The $+\lambda\mathbf{I}$ lifts every eigenvalue of $\mathbf{X}^\top\mathbf{X}$ by λ, guaranteeing invertibility *even under perfect collinearity or* $p > n$. This is ridge's defining virtue and the reason it is the canonical multicollinearity remedy.

Spectral (SVD) view — why ridge shrinks the way it does. Write $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Then the fitted values become

$$\hat{\mathbf{y}}_{\text{ridge}} = \sum_i \mathbf{u}_i \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \mathbf{u}_i^\top \mathbf{y}.$$

Ridge applies a smooth **shrinkage filter** $\frac{\sigma_i^2}{\sigma_i^2+\lambda} \in (0,1)$ to each principal direction. High-variance directions (large σ_i) are barely touched (≈ 1); low-variance directions (small σ_i, where the data are uninformative and noise dominates) are shrunk hard (≈ 0). OLS uses filter 1 everywhere and thus amplifies noise in low-variance directions; ridge suppresses it. The **effective degrees of freedom** are $\text{df}(\lambda) = \sum_i \sigma_i^2/(\sigma_i^2 + \lambda)$, decreasing from p to 0 as λ grows — a continuous dial on model complexity.

Worked Example

Suppose two features are nearly identical, giving singular values $\sigma_1 = 10$, $\sigma_2 = 0.1$. OLS weights both directions equally. Ridge with $\lambda = 1$ filters them by $\frac{100}{101} \approx 0.99$ and $\frac{0.01}{1.01} \approx 0.01$ — it trusts the well-determined direction and almost entirely discards the noisy near-collinear one. The coefficient variance, which scaled like $1/\sigma_i^2 = 100$ for the bad direction under OLS, is tamed. This is regularization as noise control, made quantitative.

Bias–variance and the existence theorem. Ridge is biased: $\mathbb{E}[\hat{\boldsymbol{\beta}}_{\text{ridge}}] = (\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} \neq \boldsymbol{\beta}$. Yet Hoerl–Kennard proved **there always exists $\lambda > 0$ giving strictly lower MSE than OLS**: the variance reduction outpaces the squared bias for small λ. This is the concrete realization of "a biased estimator can beat BLUE." Bayesian reading: ridge is the MAP estimate under a Gaussian prior $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$ with $\lambda = \sigma^2/\tau^2$.

6.2 Lasso (L1): $P = \|\boldsymbol{\beta}\|_1$

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \tfrac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 .$$

The ℓ_1 term is convex but non-differentiable at zero, which has no closed form — and is precisely what produces **exact sparsity** (variable selection).

Why corners give sparsity (subgradient/KKT derivation). At an optimum, $\mathbf{0}$ must lie in the subdifferential: $\mathbf{0} \in -\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + \lambda\partial\left\|\hat{\boldsymbol{\beta}}\right\|_1$, where $\partial|\beta_j| = \text{sign}(\beta_j)$ if $\beta_j \neq 0$ and is the interval $[-1, 1]$ if $\beta_j = 0$. Let $c_j = \mathbf{x}_{(j)}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$ be feature j's correlation with the residual. The optimality conditions are

$$\hat{\beta}_j \neq 0 \Rightarrow c_j = \lambda\,\text{sign}(\hat{\beta}_j), \qquad \hat{\beta}_j = 0 \Rightarrow |c_j| \leq \lambda.$$

So any feature whose residual-correlation is below the threshold λ is set *exactly* to zero — not merely small, but zero. Geometrically, the ℓ_1 ball is a diamond with vertices on the axes (where coordinates vanish), and the elliptical loss contours generically first touch a vertex.

The orthonormal case makes it transparent. When $\mathbf{X}^\top\mathbf{X} = \mathbf{I}$, the three estimators have closed forms in terms of the OLS coefficient $\hat{\beta}_j^{\text{ols}}$:

$$\text{ridge: } \frac{\hat{\beta}_j^{\text{ols}}}{1+\lambda}; \quad \text{lasso: } \text{sign}(\hat{\beta}_j^{\text{ols}})(|\hat{\beta}_j^{\text{ols}}| - \lambda)_+; \quad \text{best subset: } \hat{\beta}_j^{\text{ols}}\mathbf{1}\{|\hat{\beta}_j^{\text{ols}}| > \sqrt{2\lambda}\}.$$

Ridge *scales* uniformly; lasso *soft-thresholds* (shrink toward zero, then clip to zero once small enough); subset selection *hard-thresholds* (keep or kill). The soft-threshold operator is the source of lasso's simultaneous shrinkage and selection.

Worked Example

With $\hat{\beta}^{\text{ols}} = (3, 0.4, -0.2)$ and $\lambda = 0.5$ (orthonormal case): lasso gives $\text{sign}(3)(3 - 0.5)_+ = 2.5$, $(0.4 - 0.5)_+ = 0$, $(0.2 - 0.5)_+ = 0$ (with sign) $= 0$. Two coefficients vanish — automatic feature selection — while the strong one survives, shrunk. Ridge with the same λ would give $3/1.5 = 2$, $0.4/1.5 = 0.27$, $-0.2/1.5 = -0.13$: all nonzero, merely scaled. The contrast is the whole story.

Computation and pathologies. Lasso is solved by coordinate descent (cyclically soft-thresholding one coordinate at a time) or LARS (which traces the entire piecewise-linear solution path in λ). Bayesian reading: lasso is MAP under a Laplace prior. Known weaknesses: with correlated features it arbitrarily picks one and zeros the rest (unstable selection); with $p > n$ it selects at most n features; and its constant shrinkage biases large coefficients (motivating adaptive lasso, SCAD, MCP).

6.3 Elastic Net: $P = \alpha \|\boldsymbol{\beta}\|_1 + \frac{1-\alpha}{2} \|\boldsymbol{\beta}\|_2^2$

The convex combination ($\alpha \in [0, 1]$) inherits lasso's sparsity and ridge's stability. The ℓ_2 part induces a **grouping effect**: strongly correlated predictors receive similar coefficients and are selected or dropped together, curing lasso's instability under collinearity — valuable when factor exposures are correlated. Two hyperparameters (λ, α) tuned jointly by cross-validation.

6.4 Choosing λ

Use k-fold cross-validation on a log-spaced grid. Two standard choices: $\lambda_{\min}$ (minimizes CV error) and the more parsimonious one-standard-error rule $\lambda_{1\text{se}}$ (largest λ within one CV standard error of the minimum — prefers simpler models, safer against overfitting). **For time series, never shuffle:** use forward-chaining and purged/embargoed CV to prevent leakage from overlapping labels and serial correlation, or the cross-validated λ will be optimistically biased and the live performance will disappoint.

7 Practical Checklist for the Quant

1. Clean and de-bias the data: stationarity, outliers, look-ahead and survivorship bias.
2. Standardize features; decide intercept treatment.
3. Fit OLS as a baseline; inspect residual diagnostics, VIF, and the condition number.
4. If multicollinear or p large: ridge for stability, lasso/elastic-net for selection.
5. Tune λ by purged time-series CV; report both $\lambda_{\min}$ and $\lambda_{1\text{se}}$.
6. Use robust/HAC standard errors for any inference on financial data.
7. Validate out-of-sample on a strictly later period; account for how many specifications you tried (multiple-testing/backtest-overfitting).

8 A Fully Worked OLS, With and Without Collinearity

8.1 A clean fit

Fit $y = \beta_0 + \beta_1 x$ to four points $(1, 2), (2, 2), (3, 4), (4, 5)$. Center-free normal equations: $\mathbf{X}^\top \mathbf{X} = \begin{pmatrix} n & \sum x \\ \sum x & \sum x^2 \end{pmatrix} = \left(\begin{smallmatrix} 4 & 10 \\ 10 & 30 \end{smallmatrix}\right)$, $\mathbf{X}^\top \mathbf{y} = \begin{pmatrix} \sum y \\ \sum xy \end{pmatrix} = \left(\begin{smallmatrix} 13 \\ 39 \end{smallmatrix}\right)$ (since $\sum xy = 2 + 4 + 12 + 20 = 38$... recompute:

$1\cdot 2+2\cdot 2+3\cdot 4+4\cdot 5 = 2+4+12+20 = 38$, and $\sum y = 2+2+4+5 = 13$). So $\mathbf{X}^\top\mathbf{y} = (13, 38)^\top$. Determinant of $\mathbf{X}^\top\mathbf{X}$ is $4\cdot 30 - 10\cdot 10 = 120 - 100 = 20$. Inverse $\frac{1}{20}\left(\begin{smallmatrix} 30 & -10 \\ -10 & 4 \end{smallmatrix}\right)$. Then

$$\hat{\boldsymbol{\beta}} = \tfrac{1}{20}\begin{pmatrix} 30 & -10 \\ -10 & 4 \end{pmatrix}\begin{pmatrix} 13 \\ 38 \end{pmatrix} = \tfrac{1}{20}\begin{pmatrix} 390-380 \\ -130+152 \end{pmatrix} = \tfrac{1}{20}\begin{pmatrix} 10 \\ 22 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1.1 \end{pmatrix}.$$

So $\hat{y} = 0.5 + 1.1x$. The determinant 20 is comfortably far from zero — the features (constant and x) are well separated, and the fit is stable.

8.2 What collinearity does to the same machinery

Now add a second feature $x_2 = 2x_1 +$ tiny noise — nearly a copy of x_1. Then the two predictor columns are almost linearly dependent, so $\mathbf{X}^\top\mathbf{X}$ is nearly singular: its determinant approaches zero and its smallest eigenvalue $\lambda_{\min} \to 0$. Since $\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$ has eigenvalues σ^2/λ_i, the variance along the near-dependent direction explodes like $\sigma^2/\lambda_{\min} \to \infty$. Concretely, if $\lambda_{\min} = 0.001$, that coefficient's variance is $1000\sigma^2$ — the estimate is essentially noise, and its sign can flip between samples. This is the precise mechanism behind "my factor loadings are unstable": two correlated factors create a near-singular $\mathbf{X}^\top\mathbf{X}$, and OLS amplifies noise along their difference.

Worked Example

VIF made concrete. The variance inflation factor for feature j is $\mathrm{VIF}_j = 1/(1-R_j^2)$, where R_j^2 is from regressing x_j on the other features. If x_2 is 95% explained by x_1 ($R_j^2 = 0.95$), then $\mathrm{VIF} = 1/0.05 = 20$ — the coefficient's variance is 20× what it would be with uncorrelated features, i.e. its standard error is $\sqrt{20} \approx 4.5\times$ larger. A VIF above 5–10 is the standard red flag. The cure is to drop or combine the redundant feature, or to apply ridge, which we now compute.

9 A Fully Worked Ridge Computation

9.1 Ridge on the clean example

Take the clean fit above and add $\lambda = 10$. The ridge normal equations use $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I} = \left(\begin{smallmatrix} 4+10 & 10 \\ 10 & 30+10 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 14 & 10 \\ 10 & 40 \end{smallmatrix}\right)$ (penalizing both coefficients here for illustration; in practice the intercept is not penalized). Determinant $14\cdot 40 - 100 = 560 - 100 = 460$. Inverse $\frac{1}{460}\left(\begin{smallmatrix} 40 & -10 \\ -10 & 14 \end{smallmatrix}\right)$. Then

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \tfrac{1}{460}\begin{pmatrix} 40 & -10 \\ -10 & 14 \end{pmatrix}\begin{pmatrix} 13 \\ 38 \end{pmatrix} = \tfrac{1}{460}\begin{pmatrix} 520-380 \\ -130+532 \end{pmatrix} = \tfrac{1}{460}\begin{pmatrix} 140 \\ 402 \end{pmatrix} = \begin{pmatrix} 0.304 \\ 0.874 \end{pmatrix}.$$

Both coefficients shrank toward zero (from $0.5, 1.1$ to $0.30, 0.87$) — ridge's hallmark. Crucially, the determinant rose from 20 to 460: **ridge conditioned the problem**, so even if the original had been near-singular, the $+\lambda\mathbf{I}$ would have made it safely invertible.

9.2 The shrinkage as λ varies

At $\lambda = 0$ we recover OLS $(0.5, 1.1)$; at $\lambda = 10$, $(0.30, 0.87)$; as $\lambda \to \infty$, both $\to 0$. The path is smooth and monotone — ridge never zeros a coefficient exactly, it only scales it down. The effective degrees of freedom $\sum_i \sigma_i^2/(\sigma_i^2 + \lambda)$ likewise slide smoothly from p (here 2) toward 0. Choosing λ by cross-validation picks the point on this path with the best bias–variance balance.

10 A Fully Worked Lasso Soft-Threshold

10.1 The orthonormal case

When $\mathbf{X}^\top\mathbf{X} = \mathbf{I}$, lasso decouples into per-coordinate soft-thresholding: $\hat\beta_j^{\text{lasso}} = \text{sign}(\hat\beta_j^{\text{ols}})(|\hat\beta_j^{\text{ols}}|-\lambda)_+$. Take OLS coefficients $\hat{\boldsymbol\beta}^{\text{ols}} = (2.5,\ 0.6,\ -0.3,\ 1.8)$ and $\lambda = 0.7$. Apply the operator coordinate-wise:

$$\begin{aligned}
\beta_1:&\ \text{sign}(2.5)(2.5-0.7)_+ = +1.8,\\
\beta_2:&\ \text{sign}(0.6)(0.6-0.7)_+ = 0 \quad (\text{since } 0.6-0.7<0),\\
\beta_3:&\ \text{sign}(-0.3)(0.3-0.7)_+ = 0,\\
\beta_4:&\ \text{sign}(1.8)(1.8-0.7)_+ = +1.1.
\end{aligned}$$

Result: $\hat{\boldsymbol\beta}^{\text{lasso}} = (1.8,\ 0,\ 0,\ 1.1)$. Two coefficients are set *exactly* to zero — automatic feature selection — while the two strong ones survive, each shrunk by exactly $\lambda = 0.7$. Compare ridge with matching λ, which would give $(2.5, 0.6, -0.3, 1.8)/(1+\lambda)$, all four nonzero. The contrast — selection versus uniform scaling — is the entire practical difference between the two penalties.

Intuition

The soft-threshold has two regimes visible in the arithmetic: coefficients below λ in magnitude are *killed* (the selection effect), and those above are *shrunk by a constant* λ (the bias effect). The constant shrinkage is lasso's known weakness — it biases large, genuine coefficients by the same λ used to kill noise — which motivates the adaptive lasso (smaller penalty on large coefficients) and relaxed lasso (refit OLS on the selected support to undo the shrinkage bias). Knowing the operator's two regimes tells you exactly what lasso does right and what it does wrong.

11 Deeper Theory I: The Frisch–Waugh–Lovell Theorem

One of the most useful structural results about least squares explains what a regression coefficient *means* when other variables are present, and it underlies everything from control variables to factor-neutralization in quant strategies.

11.1 Statement

Partition the design matrix into two blocks $\mathbf{X} = [\mathbf{X}_1\ \mathbf{X}_2]$ with coefficient blocks $\boldsymbol\beta_1, \boldsymbol\beta_2$. The FWL theorem states that the OLS estimate $\hat{\boldsymbol\beta}_2$ from the full regression $\mathbf{y} = \mathbf{X}_1\boldsymbol\beta_1 + \mathbf{X}_2\boldsymbol\beta_2 + \boldsymbol\varepsilon$ is *identical* to the estimate from the following three-step "partialling-out" procedure:

1. Regress $\mathbf{y}$ on $\mathbf{X}_1$ alone and take the residuals $\tilde{\mathbf{y}} = \mathbf{M}_1\mathbf{y}$, where $\mathbf{M}_1 = \mathbf{I} - \mathbf{X}_1(\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top$ is the residual-maker (annihilator) matrix for $\mathbf{X}_1$.
2. Regress each column of $\mathbf{X}_2$ on $\mathbf{X}_1$ and take residuals $\tilde{\mathbf{X}}_2 = \mathbf{M}_1\mathbf{X}_2$.
3. Regress $\tilde{\mathbf{y}}$ on $\tilde{\mathbf{X}}_2$: the resulting coefficient equals $\hat{\boldsymbol\beta}_2$.

11.2 Proof

Write the normal equations for the full model in block form:

$$\begin{pmatrix}\mathbf{X}_1^\top\mathbf{X}_1 & \mathbf{X}_1^\top\mathbf{X}_2\\ \mathbf{X}_2^\top\mathbf{X}_1 & \mathbf{X}_2^\top\mathbf{X}_2\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol\beta}_1\\ \hat{\boldsymbol\beta}_2\end{pmatrix} = \begin{pmatrix}\mathbf{X}_1^\top\mathbf{y}\\ \mathbf{X}_2^\top\mathbf{y}\end{pmatrix}.$$

From the top block, $\hat{\boldsymbol\beta}_1 = (\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top(\mathbf{y} - \mathbf{X}_2\hat{\boldsymbol\beta}_2)$. Substitute into the bottom block:

$$\mathbf{X}_2^\top\mathbf{X}_1(\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top(\mathbf{y} - \mathbf{X}_2\hat{\boldsymbol\beta}_2) + \mathbf{X}_2^\top\mathbf{X}_2\hat{\boldsymbol\beta}_2 = \mathbf{X}_2^\top\mathbf{y}.$$

Group terms using $\mathbf{P}_1 = \mathbf{X}_1(\mathbf{X}_1^\top\mathbf{X}_1)^{-1}\mathbf{X}_1^\top$ (the projector) and $\mathbf{M}_1 = \mathbf{I} - \mathbf{P}_1$:

$$\mathbf{X}_2^\top\mathbf{P}_1\mathbf{y} - \mathbf{X}_2^\top\mathbf{P}_1\mathbf{X}_2\hat{\boldsymbol{\beta}}_2 + \mathbf{X}_2^\top\mathbf{X}_2\hat{\boldsymbol{\beta}}_2 = \mathbf{X}_2^\top\mathbf{y} \implies \mathbf{X}_2^\top\mathbf{M}_1\mathbf{X}_2\,\hat{\boldsymbol{\beta}}_2 = \mathbf{X}_2^\top\mathbf{M}_1\mathbf{y}.$$

Because $\mathbf{M}_1$ is symmetric and idempotent ($\mathbf{M}_1 = \mathbf{M}_1^\top = \mathbf{M}_1^2$), we have $\mathbf{X}_2^\top\mathbf{M}_1\mathbf{X}_2 = (\mathbf{M}_1\mathbf{X}_2)^\top(\mathbf{M}_1\mathbf{X}_2) = \tilde{\mathbf{X}}_2^\top\tilde{\mathbf{X}}_2$ and $\mathbf{X}_2^\top\mathbf{M}_1\mathbf{y} = \tilde{\mathbf{X}}_2^\top\tilde{\mathbf{y}}$. Thus $\hat{\boldsymbol{\beta}}_2 = (\tilde{\mathbf{X}}_2^\top\tilde{\mathbf{X}}_2)^{-1}\tilde{\mathbf{X}}_2^\top\tilde{\mathbf{y}}$ — exactly the coefficient of regressing the residualized $\tilde{\mathbf{y}}$ on the residualized $\tilde{\mathbf{X}}_2$. □

Intuition

FWL says a multiple-regression coefficient is a *partial* effect: $\hat{\boldsymbol{\beta}}_2$ measures the relationship between $\mathbf{y}$ and $\mathbf{X}_2$ *after removing the part of both that* $\mathbf{X}_1$ *can explain*. This is why "controlling for" a variable works, and it is exactly what quants do when they neutralize a signal against known factors: regress both the signal and forward returns on the factor exposures, then relate the residuals. The alpha that survives is the FWL coefficient — the predictive power orthogonal to the factors. The theorem guarantees this two-step neutralization gives the identical coefficient to throwing everything into one regression.

Worked Example

Suppose forward return y is regressed on a momentum signal x_2 and the market beta x_1. A naive univariate regression of y on x_2 might show strong predictive power — but if momentum stocks are simply high-beta, that "power" is just market exposure. FWL prescribes: residualize y against beta (the part of returns not explained by the market), residualize momentum against beta (the part of momentum orthogonal to beta), and regress the residuals. If the coefficient collapses, the signal was beta in disguise; if it survives, it is genuine beta-neutral alpha. This is the daily bread of factor research, and it is one theorem.

12 Deeper Theory II: The Geometry of Projections, Fully Developed

12.1 Projectors and their spectral structure

The hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ and its complement $\mathbf{M} = \mathbf{I} - \mathbf{H}$ are the two fundamental projectors of least squares. We collect their properties with proofs, because the entire inferential theory rests on them.

Proposition 1 (Properties of $\mathbf{H}$ and $\mathbf{M}$)**.** $\mathbf{H}$ and $\mathbf{M}$ are symmetric and idempotent; $\mathbf{HM} = \mathbf{0}$; $\operatorname{rank}(\mathbf{H}) = p$ and $\operatorname{rank}(\mathbf{M}) = n - p$; the eigenvalues of $\mathbf{H}$ are 1 (with multiplicity p) and 0 (with multiplicity $n - p$).

Proof. Symmetry: $\mathbf{H}^\top = (\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top)^\top = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top = \mathbf{H}$ using symmetry of $(\mathbf{X}^\top\mathbf{X})^{-1}$. *Idempotence:* $\mathbf{H}^2 = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top = \mathbf{H}$, the inner $\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}$ collapsing to $\mathbf{I}$. *Orthogonality:* $\mathbf{HM} = \mathbf{H}(\mathbf{I} - \mathbf{H}) = \mathbf{H} - \mathbf{H}^2 = \mathbf{0}$. *Rank and eigenvalues:* an idempotent matrix has eigenvalues in $\{0, 1\}$ (if $\mathbf{Hv} = \lambda\mathbf{v}$ then $\mathbf{H}^2\mathbf{v} = \lambda^2\mathbf{v}$, but $\mathbf{H}^2 = \mathbf{H}$ forces $\lambda^2 = \lambda$). The trace equals the sum of eigenvalues equals the number of unit eigenvalues; $\operatorname{Tr}(\mathbf{H}) = \operatorname{Tr}(\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top) = \operatorname{Tr}((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}) = \operatorname{Tr}(\mathbf{I}_p) = p$ by the cyclic property of trace. So $\mathbf{H}$ has p unit eigenvalues and $n - p$ zero ones. □

12.2 Why the residual sum of squares has $n - p$ degrees of freedom

The fitted values are $\hat{\mathbf{y}} = \mathbf{Hy}$ and the residuals are $\hat{\boldsymbol{\varepsilon}} = \mathbf{My} = \mathbf{M}\boldsymbol{\varepsilon}$ (since $\mathbf{MX} = \mathbf{0}$, the systematic part vanishes). The expected residual sum of squares is

$$\mathbb{E}[\hat{\boldsymbol{\varepsilon}}^\top\hat{\boldsymbol{\varepsilon}}] = \mathbb{E}[\boldsymbol{\varepsilon}^\top\mathbf{M}\boldsymbol{\varepsilon}] = \mathbb{E}[\operatorname{Tr}(\mathbf{M}\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top)] = \operatorname{Tr}(\mathbf{M}\,\sigma^2\mathbf{I}) = \sigma^2\operatorname{Tr}(\mathbf{M}) = \sigma^2(n - p).$$

This is the proof that $\hat{\sigma}^2 = \hat{\boldsymbol{\varepsilon}}^\top \hat{\boldsymbol{\varepsilon}}/(n-p)$ is unbiased — the divisor $n-p$ is not a fudge but the rank of the residual projector, the dimension of the space the residuals are free to live in after p constraints are imposed by fitting. Each fitted parameter removes one dimension.

Intuition

The decomposition $\mathbf{y} = \mathbf{H}\mathbf{y} + \mathbf{M}\mathbf{y}$ splits the data into the part explained by the model (living in the p-dimensional column space) and the residual (living in the orthogonal $(n-p)$-dimensional complement). These two pieces are orthogonal, so Pythagoras gives the ANOVA identity $\|\mathbf{y}\|^2 = \|\hat{\mathbf{y}}\|^2 + \|\hat{\boldsymbol{\varepsilon}}\|^2$. Everything in regression inference — R^2, the F-test, the t-test — is bookkeeping on these two orthogonal subspaces and their dimensions p and $n-p$.

13 Deeper Theory III: Distribution of the Estimator, Proved

13.1 Unbiasedness and variance

Under $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\mathbb{E}[\boldsymbol{\varepsilon}] = \mathbf{0}$, $\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$:

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

Taking expectations, $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$ (unbiased), since the second term has mean zero. For the variance, with $\mathbf{A} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$:

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \mathbf{A}\,\mathrm{Var}(\boldsymbol{\varepsilon})\,\mathbf{A}^\top = \sigma^2\mathbf{A}\mathbf{A}^\top = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}.$$

This is the formula whose eigen-structure explains collinearity's variance inflation.

13.2 The Gauss–Markov theorem, proved in full

Theorem 3. *Among all linear unbiased estimators $\tilde{\boldsymbol{\beta}} = \mathbf{C}\mathbf{y}$, OLS has the smallest variance:* $\mathrm{Var}(\tilde{\boldsymbol{\beta}}) - \mathrm{Var}(\hat{\boldsymbol{\beta}}) \succeq 0$.

Proof. Write $\mathbf{C} = \mathbf{A} + \mathbf{D}$ where $\mathbf{A} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$. Unbiasedness of $\tilde{\boldsymbol{\beta}}$ requires $\mathbb{E}[\mathbf{C}\mathbf{y}] = \mathbf{C}\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}$ for all $\boldsymbol{\beta}$, hence $\mathbf{C}\mathbf{X} = \mathbf{I}$, i.e. $(\mathbf{A}+\mathbf{D})\mathbf{X} = \mathbf{I}$. Since $\mathbf{A}\mathbf{X} = \mathbf{I}$ already, we need $\mathbf{D}\mathbf{X} = \mathbf{0}$. Now

$$\mathrm{Var}(\tilde{\boldsymbol{\beta}}) = \sigma^2\mathbf{C}\mathbf{C}^\top = \sigma^2(\mathbf{A}+\mathbf{D})(\mathbf{A}+\mathbf{D})^\top = \sigma^2(\mathbf{A}\mathbf{A}^\top + \mathbf{A}\mathbf{D}^\top + \mathbf{D}\mathbf{A}^\top + \mathbf{D}\mathbf{D}^\top).$$

The cross terms vanish: $\mathbf{A}\mathbf{D}^\top = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{D}^\top = (\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{D}\mathbf{X})^\top = \mathbf{0}$ since $\mathbf{D}\mathbf{X} = \mathbf{0}$; similarly $\mathbf{D}\mathbf{A}^\top = \mathbf{0}$. Thus $\mathrm{Var}(\tilde{\boldsymbol{\beta}}) = \sigma^2\mathbf{A}\mathbf{A}^\top + \sigma^2\mathbf{D}\mathbf{D}^\top = \mathrm{Var}(\hat{\boldsymbol{\beta}}) + \sigma^2\mathbf{D}\mathbf{D}^\top$. Since $\mathbf{D}\mathbf{D}^\top \succeq 0$ (a Gram matrix), $\mathrm{Var}(\tilde{\boldsymbol{\beta}}) \succeq \mathrm{Var}(\hat{\boldsymbol{\beta}})$, with equality iff $\mathbf{D} = \mathbf{0}$, i.e. $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}$. □

The proof is worth internalizing: unbiasedness pins down $\mathbf{C}\mathbf{X} = \mathbf{I}$, and *any* deviation $\mathbf{D}$ from the OLS choice adds a PSD chunk $\sigma^2\mathbf{D}\mathbf{D}^\top$ to the variance. OLS is optimal precisely because it is the unique linear unbiased estimator with no such excess.

13.3 Normality and the sampling distributions

Adding $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, $\hat{\boldsymbol{\beta}}$ is a linear transform of a Gaussian, hence Gaussian: $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1})$. Independently, $(n-p)\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n-p}$ (a quadratic form in the residual Gaussian, with degrees of freedom $= \mathrm{rank}(\mathbf{M}) = n-p$), and $\hat{\boldsymbol{\beta}} \perp \hat{\sigma}^2$ (they are functions of $\mathbf{H}\boldsymbol{\varepsilon}$ and $\mathbf{M}\boldsymbol{\varepsilon}$ respectively, which are independent because $\mathbf{H}\mathbf{M} = \mathbf{0}$ for jointly Gaussian vectors). The t-statistic $\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{[(\mathbf{X}^\top\mathbf{X})^{-1}]_{jj}}}$ is therefore a standard normal over the square root of an independent scaled χ^2 — the definition of a t_{n-p} distribution. This is the complete, rigorous foundation of the t-tests reported by every regression package.

14 Worked Example: Gradient Descent for OLS, Run to Convergence

The closed form $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ is unavailable at scale (inverting a $p \times p$ matrix is $O(p^3)$), so large regressions use gradient descent. We run it by hand on a tiny problem to see the convergence the optimization note predicts.

14.1 Setup

Minimize $J(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$ for the centered one-feature data $\mathbf{X} = (-1, 0, 1)^\top$ (single feature, no intercept for clarity), $\mathbf{y} = (-1, 0, 3)^\top$. The closed-form solution: $\mathbf{X}^\top\mathbf{X} = 2$, $\mathbf{X}^\top\mathbf{y} = (-1)(-1) + 0 + 1 \cdot 3 = 4$, so $\hat{\beta} = 4/2 = 2$. The gradient is $\nabla J = \mathbf{X}^\top(\mathbf{X}\beta - \mathbf{y}) = 2\beta - 4$. The curvature (Hessian) is $\mathbf{X}^\top\mathbf{X} = 2$, so $L = 2$ and the optimal step from the descent lemma is $\eta = 1/L = 0.5$.

14.2 Iterations with $\eta = 0.5$

Update rule $\beta_{t+1} = \beta_t - \eta(2\beta_t - 4) = \beta_t - 0.5(2\beta_t - 4) = \beta_t - \beta_t + 2 = 2$. **One step from any start lands exactly on $\hat{\beta} = 2$** — because with $\eta = 1/L$ on a quadratic, gradient descent is exact in one step (it jumps to the vertex of the parabola, which is also what Newton's method does). This is the best possible case: a perfectly conditioned 1-D quadratic.

14.3 A poorly-chosen step size

Now take $\eta = 0.1$ (too small) from $\beta_0 = 0$: $\beta_{t+1} = \beta_t - 0.1(2\beta_t - 4) = 0.8\beta_t + 0.4$. Iterate: $\beta_1 = 0.4$, $\beta_2 = 0.72$, $\beta_3 = 0.976$, $\beta_4 = 1.18$, … converging geometrically to 2 with ratio 0.8 per step (the factor $1 - \eta L'$… here the contraction is $|1 - 2\eta| = 0.8$). To get within 0.01 of 2 takes $\log(0.01)/\log(0.8) \approx 21$ steps. And with $\eta = 1.1 > 1/L \cdot 2 = 1$? Then $\beta_{t+1} = \beta_t - 1.1(2\beta_t - 4) = -1.2\beta_t + 4.4$, whose multiplier $|-1.2| > 1$ *diverges*: $\beta_1 = 4.4$, $\beta_2 = -0.88$, $\beta_3 = 5.46, \ldots$ oscillating outward. This is the $\eta > 2/L$ divergence the optimization note proved, now in arithmetic.

Worked Example

The three regimes seen here are universal: $\eta = 1/L$ is optimal (one step on a quadratic); $\eta < 1/L$ converges geometrically but slowly (the small-step case, 21 iterations); $\eta > 2/L$ diverges (the oscillating case). On a multi-feature problem $L = \lambda_{\max}(\mathbf{X}^\top\mathbf{X})$ and the convergence rate is governed by the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$, so a stretched, collinear design (large κ) makes even the optimal fixed step slow — the motivation for feature standardization and preconditioning. The 1-D example is the clean case where $\kappa = 1$.

15 Generalized and Weighted Least Squares

The Gauss–Markov assumption $\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$ fails constantly in finance: volatility varies across observations (heteroskedasticity) and errors are serially correlated (autocorrelation). Generalized least squares handles both.

15.1 The GLS estimator

Suppose $\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2\boldsymbol{\Omega}$ for a known positive-definite $\boldsymbol{\Omega} \neq \mathbf{I}$. The efficient estimator is no longer OLS; it is

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

Derivation by whitening. Factor $\boldsymbol{\Omega}^{-1} = \mathbf{P}^\top \mathbf{P}$ (Cholesky). Premultiply the model by $\mathbf{P}$: $\mathbf{Py} = \mathbf{PX}\boldsymbol{\beta} + \mathbf{P}\boldsymbol{\varepsilon}$. The transformed error $\mathbf{P}\boldsymbol{\varepsilon}$ has variance $\mathbf{P}(\sigma^2\boldsymbol{\Omega})\mathbf{P}^\top = \sigma^2\mathbf{P}\boldsymbol{\Omega}\mathbf{P}^\top = \sigma^2\mathbf{I}$ (since $\mathbf{P}\boldsymbol{\Omega}\mathbf{P}^\top = \mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-1}\mathbf{P}^\top = \mathbf{I}$). So the transformed model satisfies Gauss–Markov, and OLS on it — which is exactly the GLS formula above — is BLUE. GLS "whitens" correlated, heteroskedastic errors into spherical ones, then applies OLS.

15.2 Weighted least squares as the diagonal case

When $\boldsymbol{\Omega} = \mathrm{diag}(1/w_1, \ldots, 1/w_n)$ (heteroskedastic but uncorrelated), GLS reduces to **weighted least squares**: minimize $\sum_i w_i(y_i - \mathbf{x}_i^\top\boldsymbol{\beta})^2$. Observations with smaller error variance (larger w_i) get more weight — you trust the precise observations more. This is exactly the IRLS structure of logistic regression, and in finance it is how you down-weight high-volatility periods or up-weight liquid, reliable price observations.

Intuition

GLS reveals OLS as a special case ($\boldsymbol{\Omega} = \mathbf{I}$) of a more general principle: *weight each observation inversely to its noise.* The whitening transform is the same idea as the Mahalanobis distance from the probability note — measure distances in units of the noise covariance. When you do not know $\boldsymbol{\Omega}$, you estimate it (feasible GLS) or, more robustly, keep OLS point estimates but fix the standard errors with heteroskedasticity-and-autocorrelation-consistent (HAC / Newey–West) covariance estimators. The latter is the quant default because $\boldsymbol{\Omega}$ is rarely known and misspecifying it can do more harm than good.

16 Python: Least Squares Three Ways, and Regularization Paths

We now connect the theory to runnable code, computing OLS by the normal equations, by SVD (the numerically stable route), and by gradient descent, and then drawing ridge/lasso paths — the practical toolkit.

OLS three ways: normal equations, SVD, and gradient descent

```
import numpy as np

rng = np.random.default_rng(0)
n, p = 200, 5
X = rng.standard_normal((n, p))
beta_true = np.array([3.0, -2.0, 0.0, 1.5, 0.0])
y = X @ beta_true + 0.5 * rng.standard_normal(n)

# 1) Normal equations: beta = (X'X)^{-1} X'y  -- fast but unstable if ill-
    conditioned
beta_ne = np.linalg.solve(X.T @ X, X.T @ y)

# 2) SVD route: numerically stable, the recommended default (this is lstsq)
beta_svd, *_ = np.linalg.lstsq(X, y, rcond=None)

# 3) Gradient descent: how large-scale solvers actually work
L = np.linalg.eigvalsh(X.T @ X).max()       # smoothness constant
eta = 1.0 / L                               # optimal step = 1/L
beta_gd = np.zeros(p)
for _ in range(500):
    grad = X.T @ (X @ beta_gd - y)          # gradient of 1/2||y-Xb||^2
    beta_gd -= eta * grad

print("true      :", beta_true)
```

```
print("normal eq :", np.round(beta_ne, 3))
print("svd/lstsq :", np.round(beta_svd, 3))
print("grad desc :", np.round(beta_gd, 3))
# All three agree to ~3 decimals; the zero coefficients are estimated near zero.
```

The three methods agree because they solve the same convex problem; they differ only in numerical stability and scaling. The SVD route (`lstsq`) is the production default — it never forms the potentially ill-conditioned $\mathbf{X}^\top\mathbf{X}$ and gracefully handles rank deficiency by returning the minimum-norm solution.

Ridge and lasso paths, and the variance-inflation cure for collinearity

```
import numpy as np
from sklearn.linear_model import Ridge, Lasso
from sklearn.preprocessing import StandardScaler

# Build a collinear design: x4 is nearly a copy of x0 (correlation ~0.99)
rng = np.random.default_rng(1)
n = 300
x0 = rng.standard_normal(n)
X = np.column_stack([x0, rng.standard_normal((n, 3)).T.T if False else
                     rng.standard_normal((n, 3))])
X = np.column_stack([x0, rng.standard_normal((n, 2)), x0 + 0.01*rng.
    standard_normal(n)])
beta_true = np.array([2.0, 0.0, -1.0, 1.0])
# note: 4 informative columns + collinear partner handled below
y = X[:, :4] @ beta_true + 0.3 * rng.standard_normal(n)

Xs = StandardScaler().fit_transform(X)

# Variance inflation factor for the near-duplicate column (index 4)
def vif(X, j):
    from numpy.linalg import lstsq
    others = np.delete(X, j, axis=1)
    coef, *_ = lstsq(others, X[:, j], rcond=None)
    r2 = 1 - np.var(X[:, j] - others @ coef) / np.var(X[:, j])
    return 1.0 / (1.0 - r2)

print("VIF of collinear column:", round(vif(Xs, 4), 1))  # large -> unstable OLS

# Ridge stabilizes the collinear pair; lasso selects among them
for lam in [0.0, 0.1, 1.0, 10.0]:
    r = Ridge(alpha=lam, fit_intercept=False).fit(Xs, y)
    print(f"ridge alpha={lam:5}: {np.round(r.coef_, 2)}")

for lam in [0.01, 0.1, 0.5]:
    l = Lasso(alpha=lam, fit_intercept=False).fit(Xs, y)
    print(f"lasso alpha={lam:5}: {np.round(l.coef_, 2)}  nonzero={np.sum(l.coef_
        !=0)}")
# Ridge splits the effect across the collinear pair (grouping);
# lasso drives several coefficients to exactly zero (selection).
```

Running this shows the collinear column's VIF in the tens (OLS would give wildly unstable, sign-flipping coefficients for the duplicated pair), ridge spreading the shared effect smoothly across the correlated columns as λ grows, and lasso zeroing coefficients outright — the selection-versus-shrinkage contrast, now reproducible on a machine. The `StandardScaler` step is not optional: without it the penalties would act unequally across the differently-scaled columns.

Intuition

The code makes a theoretical point tangible: the *same* data fed to OLS, ridge, and lasso yields qualitatively different coefficient vectors, and the right choice depends on your goal (stability vs. selection) and your belief about the truth (dense vs. sparse). In quant practice you would wrap this in time-series cross-validation (the bias–variance note) to choose λ, never trusting a single in-sample fit — and you would standardize inside each fold to avoid the leakage discussed in the evaluation note. The estimator is three lines; the discipline around it is the job.

17 End-to-End Case Study: Predicting Returns with Regularized Regression

We assemble the whole pipeline — data construction, standardization inside cross-validation, model selection, diagnostics, and prediction intervals — on a realistic (synthetic but finance-flavored) dataset, so the isolated pieces become one workflow.

17.1 The setup

We simulate 500 monthly observations with ten candidate factors, only four of which truly drive returns, plus correlated noise factors — a deliberately sparse, partly-collinear world that rewards regularization and selection.

Constructing a sparse, collinear factor dataset

```
import numpy as np
rng = np.random.default_rng(42)
n, p = 500, 10
# Latent common factor injects correlation across several columns
common = rng.standard_normal(n)
X = rng.standard_normal((n, p))
X[:, 1] = 0.8 * common + 0.2 * X[:, 1]    # correlated cluster
X[:, 2] = 0.8 * common + 0.2 * X[:, 2]
X[:, 7] = 0.9 * X[:, 0] + 0.1 * X[:, 7]   # near-duplicate of factor 0

beta_true = np.zeros(p)
beta_true[[0, 3, 5, 8]] = [1.5, -2.0, 1.0, 0.8]    # only 4 real drivers
y = X @ beta_true + 1.0 * rng.standard_normal(n)    # noisy: low signal/noise
```

17.2 Model selection with time-series-aware cross-validation

We choose the lasso penalty by cross-validation, standardizing inside each fold to avoid leakage. For genuine financial series one would use a forward-chaining split; here we use K-fold for brevity but flag the distinction.

Leakage-free CV for the regularization strength

```
from sklearn.linear_model import LassoCV, RidgeCV
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import TimeSeriesSplit

# Pipeline standardizes within each fold -> no leakage of test statistics
tscv = TimeSeriesSplit(n_splits=5)
```

```
lasso = make_pipeline(StandardScaler(),
                      LassoCV(alphas=np.logspace(-3, 1, 50), cv=tscv, max_iter
                          =5000))
lasso.fit(X, y)
coef = lasso[-1].coef_
print("chosen alpha :", round(lasso[-1].alpha_, 4))
print("lasso coefs  :", np.round(coef, 2))
print("selected     :", np.where(np.abs(coef) > 1e-6)[0])
# Lasso should recover {0,3,5,8} (the true drivers), zeroing the noise
# and picking ONE of each collinear pair.
```

The lasso recovers the four genuine drivers and zeros the six noise factors, though under collinearity it arbitrarily keeps one of each correlated pair and drops the other — the instability the elastic net is designed to cure. The chosen α reflects the low signal-to-noise: more noise pushes α up, shrinking harder.

17.3 Diagnostics on the fitted model

A fit is not finished until its residuals are inspected. We check for heteroskedasticity, non-normality, and influential points — the diagnostics table from earlier, now executed.

Residual diagnostics: leverage, studentized residuals, normality

```
import numpy as np
Xs = StandardScaler().fit_transform(X)
Xd = np.column_stack([np.ones(n), Xs])              # add intercept
H = Xd @ np.linalg.solve(Xd.T @ Xd, Xd.T)           # hat matrix
h = np.diag(H)                                      # leverages, sum to p+1
resid = y - Xd @ np.linalg.solve(Xd.T @ Xd, Xd.T @ y)
sigma2 = resid @ resid / (n - Xd.shape[1])
stud = resid / np.sqrt(sigma2 * (1 - h))            # internally studentized

print("mean leverage (=(p+1)/n):", round(h.mean(), 4), "vs", round(Xd.shape[1]/n
    ,4))
print("max leverage            :", round(h.max(), 4))
print("# |studentized| > 3      :", int(np.sum(np.abs(stud) > 3)))
# Breusch-Pagan-style check: regress squared residuals on fitted values
fitted = Xd @ np.linalg.solve(Xd.T @ Xd, Xd.T @ y)
bp_slope = np.polyfit(fitted, resid**2, 1)[0]
print("heterosked. slope (~0 ok):", round(bp_slope, 4))
```

High-leverage points (here flagged when h_{ii} far exceeds the average $(p+1)/n$) and large studentized residuals ($|r| > 3$) localize observations that may distort the fit — in finance, often a crisis month. A nonzero Breusch–Pagan slope signals heteroskedasticity, prompting robust standard errors.

17.4 Prediction intervals, not just point forecasts

A point forecast without an uncertainty band is dangerous for sizing decisions. For a new $\mathbf{x}_0$, the prediction has two variance sources — estimation uncertainty in $\hat{\boldsymbol{\beta}}$ and the irreducible noise:

$$\widehat{\mathrm{Var}}(y_0 - \hat{y}_0) = \hat{\sigma}^2\big(1 + \mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0\big),$$

giving the $(1-\alpha)$ interval $\hat{y}_0 \pm t_{n-p,1-\alpha/2}\,\hat{\sigma}\sqrt{1 + \mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0}$. The "1" is the new observation's own noise; the quadratic term is the estimation uncertainty, which *grows* as $\mathbf{x}_0$ moves away from the training data's center (extrapolation is less certain). This is the regression analogue of the leverage formula and the honest way to report a forecast.

Worked Example

A model forecasts next month's return at +1.2%. The naive user sizes a position on 1.2%. The disciplined user computes the prediction interval: if it is $[-3.1\%, +5.5\%]$, the forecast is nearly worthless for sizing (the band straddles zero by a wide margin), and the right action is a small or no position. The same point forecast with a tight band $[+0.8\%, +1.6\%]$ would justify a real position. The interval, driven by $\hat{\sigma}$ and the leverage term, is what separates a forecast from a gamble — and it widens automatically when you extrapolate to unusual factor values, exactly when you should be most cautious.

Intuition

This case study is the whole series in miniature: linear algebra (the hat matrix, the projection), optimization (the convex fit, gradient descent at scale), probability (the noise model, the prediction interval's distribution theory), regularization (lasso selection under collinearity), and evaluation discipline (leakage-free CV, residual diagnostics). No single piece is hard; the skill is wiring them into a pipeline whose every step respects the assumptions the earlier sections derived. A model that skips the diagnostics or the interval is not wrong so much as *unfinished* — and in finance, unfinished models lose money in the tails the point forecast never mentioned.

18 Robust and Quantile Regression

OLS minimizes squared error, which is optimal under Gaussian noise but disastrous under the heavy tails of financial data — a single outlier can dominate the fit. Robust and quantile methods address this.

18.1 Why OLS is fragile to outliers

The squared loss gives a point with residual r an influence proportional to r (the gradient is $-2r\mathbf{x}$), so a point with residual $10\times$ the typical one exerts $10\times$ the pull. One crisis observation can swing the entire coefficient vector. The breakdown point of OLS is 0: a single bad point, placed far enough, can move the estimate arbitrarily.

18.2 M-estimators and the Huber loss

A robust **M-estimator** replaces the squared loss with a function $\rho(r)$ that grows more slowly for large residuals. The **Huber loss** is quadratic for small residuals (efficient, like OLS) and linear for large ones (robust, bounded influence):

$$\rho_\delta(r) = \begin{cases} \tfrac{1}{2}r^2 & |r| \le \delta \\ \delta(|r| - \tfrac{1}{2}\delta) & |r| > \delta. \end{cases}$$

Its derivative (the influence function) is clipped at $\pm\delta$, so no single point can exert unbounded pull — the breakdown improves and the estimate resists outliers while remaining efficient for the bulk. Huber regression is fit by IRLS (the same engine as logistic regression), with weights that down-weight large-residual points each iteration.

18.3 Quantile regression

Instead of the conditional mean, **quantile regression** models a conditional quantile (e.g. the median, or the 5% tail) by minimizing the asymmetric *pinball loss*

$$\rho_\tau(r) = \begin{cases} \tau r & r \geq 0 \\ (\tau - 1)r & r < 0, \end{cases}$$

whose minimizer is the τ-th conditional quantile. At $\tau = 0.5$ this is least-absolute-deviations (the median, robust to outliers). For a quant, modeling the 5% quantile of returns *is* conditional Value-at-Risk — quantile regression estimates VaR directly as a function of features, rather than assuming a distribution. It is the regression analogue of the MLE-under-Laplace-noise connection: pinball loss is the negative log-likelihood of an asymmetric Laplace distribution.

Worked Example

Estimating downside risk: regress next-day return on volatility and recent drawdown using quantile regression at $\tau = 0.05$. The fitted line gives, for any feature values, the 5% conditional quantile — the level the return falls below only 5% of the time, i.e. the conditional VaR. Unlike a Gaussian VaR (which assumes normal returns and badly underestimates tail risk, per the probability note's CLT caveat), quantile regression makes no distributional assumption and fits the actual tail behavior. Fitting several τ values traces the whole conditional distribution, a far richer risk picture than a single mean forecast.

19 Bayesian Linear Regression

The MAP view connected ridge to a Gaussian prior; the full Bayesian treatment gives the entire posterior, hence honest uncertainty — valuable when data are scarce and overconfidence is costly.

19.1 The conjugate posterior

With likelihood $\mathbf{y} \mid \boldsymbol{\beta} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$ and prior $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$, the posterior is Gaussian (conjugacy, from the probability note's Gaussian-closure results):

$$\boldsymbol{\beta} \mid \mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{post}}, \boldsymbol{\Sigma}_{\text{post}}), \quad \boldsymbol{\Sigma}_{\text{post}} = \left(\tfrac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X} + \tfrac{1}{\tau^2}\mathbf{I}\right)^{-1}, \quad \boldsymbol{\mu}_{\text{post}} = \tfrac{1}{\sigma^2}\boldsymbol{\Sigma}_{\text{post}}\mathbf{X}^\top\mathbf{y}.$$

The posterior *mean* $\boldsymbol{\mu}_{\text{post}}$ is exactly the ridge estimate with $\lambda = \sigma^2/\tau^2$ — recovering the MAP result — but now we also have the posterior *covariance* $\boldsymbol{\Sigma}_{\text{post}}$, quantifying how certain we are about each coefficient. This is the information point estimation throws away.

19.2 The predictive distribution

For a new $\mathbf{x}_0$, integrating over the coefficient posterior gives a Gaussian predictive distribution with mean $\mathbf{x}_0^\top\boldsymbol{\mu}_{\text{post}}$ and variance $\mathbf{x}_0^\top\boldsymbol{\Sigma}_{\text{post}}\mathbf{x}_0 + \sigma^2$ — the first term is parameter uncertainty (large where data are sparse, i.e. extrapolation), the second the irreducible noise. This is the principled version of the prediction interval from the case study, and it widens automatically in regions far from the training data — the Bayesian model *knows what it does not know*, a property point estimates lack.

Intuition

Bayesian linear regression unifies several threads of this series: it is ridge (the posterior mean), it gives the prediction interval (the predictive variance), and its prior is the regularizer (the MAP connection). What it adds is *coherent uncertainty*: every quantity comes with a posterior

distribution, so a forecast is never a bare number but a distribution whose spread reflects both noise and ignorance. For a quant sizing positions, the predictive variance is as important as the predictive mean — it is the difference between a confident bet and a tentative one, and the Bayesian framework delivers it for free as a by-product of the same Gaussian algebra.

20 Dimension Reduction for Regression: PCR and PLS

When predictors are many and collinear, two strategies project them to a low-dimensional space before regressing.

20.1 Principal components regression

PCR regresses $\mathbf{y}$ on the top few principal components of $\mathbf{X}$ (from the unsupervised note's PCA). It tames collinearity (the components are orthogonal) and reduces variance (fewer effective parameters). Its weakness: PCA chooses directions of maximum *predictor* variance, which need not be the directions most predictive of $\mathbf{y}$ — a low-variance component might carry the signal, and PCR would discard it.

20.2 Partial least squares

PLS fixes this by choosing projection directions that maximize the *covariance with the response*, not just predictor variance. Each PLS component is the linear combination of features most correlated with the (current residual of the) target, so PLS finds supervised directions — typically needing fewer components than PCR for the same predictive power. It is the workhorse of chemometrics and appears in factor-return modeling where predictors are many and collinear.

Intuition

PCR and PLS sit between OLS and ridge/lasso as collinearity remedies, differing in *what* they optimize when compressing. PCA/PCR is unsupervised (predictor variance), PLS is supervised (predictor–response covariance), ridge shrinks all directions smoothly, and lasso selects. The right choice depends on whether the signal aligns with high-variance predictor directions (PCR fine), needs supervision to find (PLS better), or is sparse (lasso). For a quant with hundreds of correlated factors, PLS often gives a compact, predictive model, but it must be validated with the same purged time-series CV — supervised compression is another knob that can overfit.

21 Instrumental Variables and Two-Stage Least Squares

OLS is biased when a regressor is correlated with the error — *endogeneity* — which is pervasive in finance and economics (simultaneity, omitted variables, measurement error). Instrumental variables repair it, and the construction is one of the most important in applied econometrics.

21.1 The endogeneity problem

Suppose $y = \beta x + \varepsilon$ but $\mathrm{Cov}(x, \varepsilon) \neq 0$ (e.g. x is price and ε contains demand shocks that also move price). Then OLS is inconsistent: $\hat{\beta}_{\mathrm{OLS}} = \beta + \mathrm{Cov}(x, \varepsilon)/\mathrm{Var}(x) \not\to \beta$. The bias does not vanish with more data — it is structural, not statistical. No amount of data fixes a confounded regression.

21.2 The IV estimator

An **instrument** z satisfies two conditions: *relevance* ($\mathrm{Cov}(z,x)\neq 0$ — it moves the endogenous regressor) and *exogeneity* ($\mathrm{Cov}(z,\varepsilon)=0$ — it affects y only through x). Then

$$\hat{\beta}_{\mathrm{IV}}=\frac{\mathrm{Cov}(z,y)}{\mathrm{Cov}(z,x)}\xrightarrow{p}\beta,$$

because $\mathrm{Cov}(z,y)=\beta\,\mathrm{Cov}(z,x)+\mathrm{Cov}(z,\varepsilon)=\beta\,\mathrm{Cov}(z,x)$ (the second term vanishes by exogeneity). Dividing recovers β consistently. The instrument extracts the variation in x that is uncorrelated with the error, and uses only that clean variation to identify the effect.

21.3 Two-stage least squares

With multiple regressors and instruments, **2SLS** operationalizes IV: (1) regress each endogenous regressor on all instruments (and exogenous controls), keeping the fitted values $\hat{x}$ — the part of x explained by the instruments, hence exogenous; (2) regress y on $\hat{x}$. The matrix form is $\hat{\boldsymbol{\beta}}_{\mathrm{2SLS}}=(\hat{\mathbf{X}}^\top\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}^\top\mathbf{y}$ with $\hat{\mathbf{X}}=\mathbf{P}_Z\mathbf{X}$, where $\mathbf{P}_Z=\mathbf{Z}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top$ is the projection onto the instrument space. So 2SLS projects the regressors onto the instruments (purging the endogenous part) before running OLS — the projection machinery of this document, used to clean rather than to fit.

Intuition

2SLS is the projection theory turned to a causal purpose. The first stage projects $\mathbf{X}$ onto the instrument space $\mathbf{P}_Z\mathbf{X}$, isolating the exogenous variation; the second stage runs OLS on that purged regressor. The cost is variance: $\hat{\mathbf{X}}$ has less variation than $\mathbf{X}$ (it is a projection, hence shorter), so IV estimates are noisier than OLS — a bias–variance trade where we accept variance to remove bias. *Weak instruments* (low relevance, $\mathrm{Cov}(z,x)\approx 0$) make this catastrophic: dividing by a near-zero covariance explodes the variance and reintroduces bias. The first-stage F-statistic must be large (a rule of thumb is $F>10$) for IV to be trustworthy.

Worked Example

Estimating how a fund's size affects its returns: size is endogenous (good past returns attract inflows, so size correlates with the return error). A valid instrument might be a plausibly-exogenous driver of size unrelated to skill — e.g. a mechanical index-inclusion flow. The first stage regresses size on the instrument (isolating size variation driven by mechanical flows, not performance); the second stage regresses returns on this purged size. The IV estimate of the size–return relationship is then free of the reverse-causality bias that contaminates a naive OLS. The validity hinges entirely on the instrument's exogeneity, which is an economic argument, not a statistical test — the hardest and most contestable part of any IV study.

22 Coordinate Descent for the Lasso, Worked to Convergence

The lasso has no closed form, but coordinate descent — cycling through coordinates, each a soft-thresholding step — solves it efficiently. We run it by hand to see the sparsity emerge.

22.1 The soft-thresholding update

For standardized features, the lasso coordinate update for β_j holding others fixed is the soft-threshold of the partial residual correlation:

$$\beta_j\leftarrow S_\lambda\!\left(\tfrac{1}{n}\sum_i x_{ij}r_i^{(-j)}\right),\qquad S_\lambda(z)=\mathrm{sign}(z)\max(|z|-\lambda,0),$$

where $r_i^{(-j)} = y_i - \sum_{k \neq j} x_{ik}\beta_k$ is the residual excluding feature j. The soft-threshold sets the coefficient to exactly zero when the correlation is below λ — the mechanism of lasso sparsity, derived from the KKT conditions of the ℓ_1 penalty.

22.2 A two-feature run

Standardized data, $n = 4$. Feature 1 strongly related to y, feature 2 weakly. After centering, suppose the relevant correlations are $\frac{1}{n}\sum x_{i1}y_i = 0.8$ and $\frac{1}{n}\sum x_{i2}y_i = 0.15$, features orthogonal for simplicity. Take $\lambda = 0.2$, start $\boldsymbol{\beta} = (0, 0)$.

- **Update** β_1: partial residual correlation $= 0.8$ (since $\beta_2 = 0$). $\beta_1 \leftarrow S_{0.2}(0.8) = 0.8 - 0.2 = 0.6$.
- **Update** β_2: correlation $= 0.15$. $\beta_2 \leftarrow S_{0.2}(0.15) = \text{sign}(0.15)\max(0.15 - 0.2, 0) = 0$. *Feature 2 is zeroed* — its signal (0.15) is below the threshold ($\lambda = 0.2$).
- **Second sweep**: with orthogonal features the updates do not change (β_1 recomputes to 0.6, β_2 to 0) — converged.

The lasso has performed selection: feature 1 survives with a *shrunken* coefficient (0.6 rather than its OLS 0.8 — the bias the penalty imposes), feature 2 is eliminated. Lowering λ to 0.1 would admit feature 2 ($S_{0.1}(0.15) = 0.05$); raising it to 0.9 would zero both. The threshold λ is the selection dial, and the worked arithmetic shows exactly how it decides.

Intuition

Coordinate descent is the lasso's workhorse because each step is a trivial scalar soft-threshold, and the algorithm exploits sparsity: once a coefficient is zero, it is cheap to check and often stays zero, so the active set stays small. Cycling with warm starts along a decreasing λ grid (the regularization path) computes the entire lasso path efficiently — the basis of `glmnet` and `sklearn`'s lasso. The soft-threshold operator is the same one that appears in wavelet denoising and proximal gradient methods, a unifying primitive of sparse optimization that the KKT conditions of the ℓ_1 penalty generate.

23 Generalized Additive Models

Between rigid linear models and flexible black boxes sit **generalized additive models**: $g(\mathbb{E}[y]) = \beta_0 + \sum_j f_j(x_j)$, where each f_j is a smooth (usually spline) function learned from data. They relax linearity (each feature gets a flexible curve) while keeping additivity (no interactions unless explicitly added), so the fitted f_j can be *plotted and interpreted* — you see exactly how each feature affects the prediction, nonlinearly. Fitting uses backfitting (cycle through features, fit each f_j to the partial residual — coordinate descent for functions) or penalized splines. GAMs are the interpretable nonlinear model: more flexible than linear regression, more transparent than a boosted ensemble.

Worked Example

Modeling default probability as a GAM: $\text{logit}(p) = \beta_0 + f_1(\text{leverage}) + f_2(\text{age}) + f_3(\text{utilization})$. The fitted f_1 might show default risk rising slowly then sharply above a leverage threshold (a nonlinearity a linear model misses), and the curve is directly inspectable and explainable to a regulator — unlike a boosted tree's opaque interactions. GAMs thus occupy a valuable niche in regulated finance: they capture the dominant nonlinearities while remaining additive and interpretable, and they can be constrained monotone (like the boosted trees of document 5) for defensibility. When the truth is mostly additive nonlinearity, a GAM beats both the linear model (too rigid) and the black box (too opaque).

24 Influence Diagnostics: Cook's Distance

Beyond leverage (which measures a point's *potential* influence from its $\mathbf{x}$ position), **Cook's distance** measures a point's *actual* influence on the fit by how much all fitted values change when it is deleted:

$$D_i = \frac{\sum_j (\hat{y}_j - \hat{y}_{j(i)})^2}{p\,\hat{\sigma}^2} = \frac{e_i^2}{p\,\hat{\sigma}^2} \cdot \frac{h_{ii}}{(1-h_{ii})^2},$$

combining the residual e_i and the leverage h_{ii} — a point is influential only if it has *both* an unusual $\mathbf{x}$ (high leverage) *and* a large residual. The closed form (no actual refitting needed, by the leave-one-out algebra of the hat matrix) makes it cheap to compute for all points. A common flag is $D_i > 4/n$. In finance, the influential points are often crisis observations, and the decision to keep, downweight (robust regression), or model them separately is consequential — a single 2008 or 2020 month can dominate a naively-fit model.

25 Ridge as Spectral Shrinkage: a Worked Computation

The main text showed ridge shrinks along the SVD directions by factors $\frac{\sigma_i^2}{\sigma_i^2+\lambda}$; we compute these for a concrete design to see collinearity's effect made numerical.

25.1 The design and its SVD

Take a centered, standardized $\mathbf{X}$ whose $\mathbf{X}^\top\mathbf{X}$ has eigenvalues (squared singular values) $\sigma_1^2 = 4.0, \sigma_2^2 = 1.0, \sigma_3^2 = 0.05$. The tiny σ_3^2 signals near-collinearity (a direction with almost no variation), and the condition number is $\kappa = \sigma_1^2/\sigma_3^2 = 80$ — moderately ill-conditioned.

25.2 Shrinkage factors

OLS divides each direction's signal by σ_i^2, so the third (collinear) direction's coefficient is divided by 0.05 — a 20× amplification of any noise there, the variance inflation that makes OLS unstable. Ridge with $\lambda = 0.1$ multiplies each direction by the filter factor $f_i = \frac{\sigma_i^2}{\sigma_i^2+\lambda}$:

$$f_1 = \frac{4.0}{4.1} = 0.976, \quad f_2 = \frac{1.0}{1.1} = 0.909, \quad f_3 = \frac{0.05}{0.15} = 0.333.$$

The high-variance directions ($f_1, f_2 \approx 1$) are barely touched, while the noisy collinear direction is shrunk to a third of its OLS value — ridge selectively damps exactly the unstable direction. With $\lambda = 1$: $f_3 = 0.05/1.05 = 0.048$, nearly killing the collinear direction, while $f_1 = 0.8$ still preserves most of the dominant signal. **Ridge is a low-pass filter on the spectrum of $\mathbf{X}^\top\mathbf{X}$**, attenuating low-variance (high-noise) directions while preserving high-variance (high-signal) ones — the precise mechanism of its variance reduction.

25.3 Effective degrees of freedom

The effective degrees of freedom of a ridge fit is $\mathrm{df}(\lambda) = \sum_i \frac{\sigma_i^2}{\sigma_i^2+\lambda} = \sum_i f_i$. Here at $\lambda = 0.1$: $\mathrm{df} = 0.976 + 0.909 + 0.333 = 2.22$ — the three-parameter model behaves like a 2.22-parameter one, the fractional value capturing the partial shrinkage. At $\lambda = 0$ (OLS) $\mathrm{df} = 3$; at $\lambda \to \infty$, $\mathrm{df} \to 0$. This continuous, fractional model complexity (versus the integer parameter count of subset selection) is what lets ridge tune the bias–variance balance smoothly, and it is the df that enters AIC/Cp-style criteria (document 3) for choosing λ.

Intuition

The spectral view is the deepest way to understand regularization. OLS inverts $\mathbf{X}^\top\mathbf{X}$, dividing by every eigenvalue including the dangerously small ones — so a near-collinear direction (tiny σ_i^2) gets enormous, noise-dominated weight. Ridge adds λ to every eigenvalue before inverting, which barely changes large eigenvalues but lifts small ones away from zero, capping the amplification. The filter factors $\sigma_i^2/(\sigma_i^2+\lambda)$ quantify this exactly, and the effective degrees of freedom sum them into a single complexity number. This connects to the condition number (linear-algebra note), the bias–variance tradeoff (document 3), and the SVD (linear-algebra note) — one picture unifying them.

26 The ANOVA Decomposition and the F-test, Worked

26.1 Sums of squares

The orthogonal split $\mathbf{y} = \hat{\mathbf{y}} + \hat{\boldsymbol{\varepsilon}}$ (the projection geometry of this document) gives, by Pythagoras, the ANOVA identity TSS = ESS + RSS: total sum of squares (about the mean) equals explained plus residual. $R^2 = \text{ESS}/\text{TSS}$ is the fraction explained. We work a case: $n = 20$, $p = 3$ predictors (plus intercept), TSS = 200, RSS = 80. Then ESS = 120, $R^2 = 120/200 = 0.60$.

26.2 The overall F-test

To test whether the predictors jointly explain anything (all slopes zero), the F-statistic compares explained variance per predictor to residual variance per residual degree of freedom:

$$F = \frac{\text{ESS}/p}{\text{RSS}/(n-p-1)} = \frac{120/3}{80/16} = \frac{40}{5} = 8.0.$$

Under the null (no relationship) and Gaussian errors, $F \sim F_{3,16}$ (the ratio of independent scaled χ^2's, from this document's distribution theory — ESS and RSS are quadratic forms in the orthogonal $\mathbf{H}$ and $\mathbf{M}$ projections, hence independent χ^2). The 5% critical value of $F_{3,16}$ is about 3.24; since $8.0 > 3.24$, we reject the null — the predictors jointly matter. The adjusted $R^2 = 1 - \frac{\text{RSS}/(n-p-1)}{\text{TSS}/(n-1)} = 1 - \frac{5}{200/19} = 1 - \frac{5}{10.53} = 0.525$ penalizes the three parameters, falling below the raw 0.60.

Intuition

The F-test is the projection geometry turned into inference: ESS and RSS are the squared lengths of $\mathbf{y}$'s components in the model subspace (p dimensions) and its orthogonal complement ($n-p-1$ dimensions), and their ratio — normalized by those dimensions — is the F-statistic, distributed as a ratio of independent χ^2's precisely because $\mathbf{H}$ and $\mathbf{M}$ are orthogonal projections. Every regression F- and t-test is this geometry plus the Gaussian-error assumption. Seeing it as "signal length per signal dimension over noise length per noise dimension" demystifies the test and ties it directly to the rank-p/rank-$(n-p)$ projection structure proved earlier in this document.

27 Worked GLS: Whitening Autocorrelated Errors

We apply the GLS whitening of the main text to a concrete AR(1) error structure, the canonical financial case. Suppose errors follow $\varepsilon_t = \rho\varepsilon_{t-1} + u_t$ with $\rho = 0.5$ — serial correlation, violating OLS's spherical-error assumption. The error covariance is $\boldsymbol{\Omega}$ with $\Omega_{ts} = \rho^{|t-s|}/(1-\rho^2)$ (the AR(1) Toeplitz structure). The Cholesky-based whitening transform, for AR(1), is the simple **Cochrane–Orcutt** difference: $\tilde{y}_t = y_t - \rho y_{t-1}$, $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \rho\mathbf{x}_{t-1}$ (with a scaling for the first observation).

After this transform, the errors $\tilde{\varepsilon}_t = \varepsilon_t - \rho\varepsilon_{t-1} = u_t$ are white (uncorrelated), so OLS on $(\tilde{y}, \tilde{\mathbf{x}})$ is efficient — this is GLS. Numerically, with $\rho = 0.5$, an observation $y_t = 3$ following $y_{t-1} = 2$ becomes $\tilde{y}_t = 3 - 0.5(2) = 2$, and similarly for the regressors; the differenced regression removes the predictable part of each error from its predecessor. Ignoring the autocorrelation (plain OLS) would leave the coefficients unbiased but their standard errors wrong (typically too small, overstating significance) — the reason financial regressions report Newey–West HAC standard errors when ρ is unknown or the structure is richer than AR(1).

Intuition

Autocorrelated errors are the rule, not the exception, in financial time series, and the GLS/whitening response embodies a principle from across the series: transform the problem into one where the standard assumptions hold, then apply the standard method. Whitening (here, AR(1) differencing) makes the errors spherical so OLS is BLUE; it is the same move as the Mahalanobis/Cholesky decorrelation of the probability note and the kernel trick's lift to a space where the structure is linear. When the exact error structure is unknown, the robust fallback — keep OLS point estimates, fix the standard errors with HAC — is the practical quant default, trading a little efficiency for robustness against misspecifying $\boldsymbol{\Omega}$.

28 Time-Series Regression: Tobit, Censoring, and Regime Models

Financial regressions face censored and regime-dependent data that violate OLS assumptions; this section completes the regression toolkit.

28.1 Censored regression (Tobit)

When the dependent variable is censored — observed only within a range (e.g. a bid capped at a limit, a return floored by a circuit breaker, demand truncated by capacity) — OLS is biased because it treats the censoring boundary as a real value. The **Tobit model** posits a latent $y^\star = \mathbf{x}^\top\boldsymbol{\beta} + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, observed as $y = \max(y^\star, c)$ (lower censoring at c). Its likelihood combines a density term for uncensored observations and a probability-mass term $\Phi((c - \mathbf{x}^\top\boldsymbol{\beta})/\sigma)$ for censored ones, fit by maximum likelihood. Ignoring censoring and running OLS attenuates the coefficients toward zero (the censored mass flattens the apparent slope), so the Tobit correction matters whenever a meaningful fraction of observations sit at a boundary.

28.2 Regime-switching regression

Market relationships change across regimes (calm vs. crisis), and a single linear fit averages incompatible states. A **Markov-switching regression** lets the coefficients depend on a hidden state following a Markov chain, fit by EM (the latent-state machinery of document 7): the E-step infers state probabilities (the forward–backward algorithm), the M-step refits regime-specific coefficients weighted by those probabilities. The result is a set of regime-specific models plus transition probabilities — capturing, e.g., a low-volatility regime where momentum works and a high-volatility regime where it reverses. This connects regression (document 1), latent-variable EM (document 7), and the regime-classification theme recurring across the series.

Intuition

Censoring and regime-switching are two ways real financial data break the clean OLS setup, and both are handled by the same principle the series returns to: write down the correct likelihood (with the censoring mass, or with the latent regime) and maximize it, reusing the MLE and EM machinery rather than forcing the data into OLS. The Tobit and Markov-switching models are

not exotic — censoring appears wherever limits and floors bind, and regimes are the defining feature of financial time series. Recognizing when the standard linear model's assumptions fail, and which extension repairs them, is the applied skill that the theory exists to support.

29 Seemingly Unrelated Regressions and Systems

Often several related regressions are estimated jointly — returns of multiple assets on common factors, or a system of demand equations. If their error terms are correlated across equations (asset-specific shocks that comove), estimating them *jointly* via **seemingly unrelated regressions** (SUR) is more efficient than separate OLS, because the cross-equation error correlation carries information. SUR is feasible GLS (document 1's GLS) applied to the stacked system, with the error covariance estimated from separate-OLS residuals. The efficiency gain is largest when the regressors differ across equations and the cross-equation correlations are strong — common in factor-model estimation across many assets, where SUR pools the cross-sectional error structure.

Intuition

SUR is GLS lifted to a system of equations: the same whitening-by-the-error-covariance principle (document 1) that makes GLS efficient for one equation makes joint estimation efficient for many correlated equations. For a quant estimating factor loadings across an asset universe, the cross-asset residual correlations are exactly the systematic risk the factor model is trying to capture, so exploiting them in estimation (SUR) rather than discarding them (separate OLS) both improves efficiency and respects the structure. It is another instance of the unifying lesson: model the full covariance structure, and estimation efficiency follows.

30 Case Study: Factor Model for Cross-Sectional Returns

We build a complete cross-sectional return-prediction regression with the full discipline of this document — diagnostics, regularization, and robust inference.

Cross-sectional factor regression with diagnostics and ridge

```
import numpy as np
from sklearn.linear_model import LinearRegression, RidgeCV
from sklearn.preprocessing import StandardScaler

rng = np.random.default_rng(0)
N = 1500                                              # firms
# Factor exposures: value, momentum, size, quality, low-vol
F = rng.standard_normal((N, 5))
F[:,1] += 0.5*F[:,0]                                  # value/momentum correlated
true_premia = np.array([0.03, 0.05, -0.02, 0.04, 0.01])
ret = F @ true_premia + 0.10*rng.standard_normal(N)   # noisy returns

Fs = StandardScaler().fit_transform(F)
# OLS factor premia
ols = LinearRegression().fit(Fs, ret)
# Variance inflation factors detect the value/momentum collinearity
def vif(X):
    return [1/(1-LinearRegression().fit(np.delete(X,j,1), X[:,j]).score(
            np.delete(X,j,1), X[:,j])) for j in range(X.shape[1])]
print("VIFs:", np.round(vif(Fs), 2), " (>5 signals collinearity)")
# Ridge stabilizes the collinear premia
ridge = RidgeCV(alphas=np.logspace(-3,1,20)).fit(Fs, ret)
print("OLS premia  :", np.round(ols.coef_, 4))
```

```
print("Ridge premia:", np.round(ridge.coef_, 4), " alpha=", round(ridge.alpha_,4)
    )
# Newey-West-style robust SE would adjust for cross-sectional correlation;
# ridge shrinks the collinear value/momentum premia toward stable values.
```

This estimates factor risk premia by cross-sectional regression — the workhorse of empirical asset pricing. The VIF diagnostic flags the engineered value/momentum collinearity (document 1's multicollinearity treatment), and ridge stabilizes the affected premia, trading a little bias for much-reduced variance in the correlated coefficients. In practice the standard errors would be adjusted for cross-sectional correlation (the GLS/robust-SE theme), and the regression would be run period-by-period (Fama–MacBeth) with the premia averaged across periods — but the core is this regularized, diagnosed cross-sectional fit.

Intuition

The cross-sectional factor regression is where this document's machinery meets empirical finance head-on: OLS gives the premia, projection geometry underlies the fit, multicollinearity (correlated factors like value and momentum) demands VIF diagnostics and ridge, and robust standard errors handle the cross-sectional dependence that violates the spherical-error assumption. The entire apparatus — estimation, diagnostics, regularization, robust inference — is exercised in one realistic task. A quant who can build, diagnose, and defend this regression has operationalized the document, and the same template (regularized, diagnosed, robustly-inferred regression) recurs throughout quantitative finance.

31 Further Worked Exercises

These exercises consolidate the regression theory through computation and proof.

Worked Example

Exercise. Derive the ridge estimator and show it always exists, even when $\mathbf{X}^\top\mathbf{X}$ is singular.
Solution. The ridge objective is $\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+\lambda\|\boldsymbol{\beta}\|^2$. Setting the gradient to zero: $-2\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+2\lambda\boldsymbol{\beta}=0$, giving $(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})\boldsymbol{\beta}=\mathbf{X}^\top\mathbf{y}$, so $\hat{\boldsymbol{\beta}}_{\text{ridge}}=(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$. The matrix $\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}$ is always invertible for $\lambda>0$: $\mathbf{X}^\top\mathbf{X}$ is positive semidefinite (eigenvalues ≥ 0), so adding $\lambda\mathbf{I}$ shifts all eigenvalues to $\geq\lambda>0$, making it positive definite hence invertible. This is why ridge is defined even when $p>n$ or features are collinear (where OLS's $\mathbf{X}^\top\mathbf{X}$ is singular) — the $\lambda\mathbf{I}$ regularizes the inversion, the spectral mechanism of the main text.

Worked Example

Exercise. Show that adding an irrelevant feature to OLS never increases training R^2 but can increase test error.
Solution. Adding a column to $\mathbf{X}$ enlarges the column space, so the projection of $\mathbf{y}$ onto it can only get closer (or stay equal) to $\mathbf{y}$ — RSS cannot increase, so training R^2 cannot decrease. But the extra coefficient is estimated with noise, adding variance to predictions (the bias–variance tradeoff, document 3); on test data this added variance can outweigh any (here zero) bias reduction, raising test error. This is precisely why training R^2 is a misleading model-selection criterion and why adjusted R^2, AIC, or cross-validation (penalizing parameters) are needed — the central lesson connecting documents 1 and 3.

Worked Example

Exercise. For weighted least squares with weights w_i, derive the estimator and explain when to use it.
Solution. WLS minimizes $\sum_i w_i(y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2$. In matrix form with $\mathbf{W} = \text{diag}(w_i)$: minimize $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$, giving $\hat{\boldsymbol{\beta}}_{\text{WLS}} = (\mathbf{X}^\top \mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{W}\mathbf{y}$. Use it under heteroskedasticity (non-constant error variance): set $w_i = 1/\sigma_i^2$ so noisier observations get less weight, restoring efficiency (the GLS principle, main text). It is also the IRLS inner step for GLMs (document 2). In finance, WLS down-weights high-variance (e.g. crisis) periods or low-liquidity observations when their noise would otherwise dominate the fit.

32 Worked Exercise Solutions

Solution to Exercise 1 (ridge Hessian). The penalized objective $\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|^2$ has gradient $-\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}$ and Hessian $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$. For any $\mathbf{v} \neq \mathbf{0}$, $\mathbf{v}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{v} = \|\mathbf{X}\mathbf{v}\|^2 + \lambda\|\mathbf{v}\|^2 \geq \lambda\|\mathbf{v}\|^2 > 0$, so the Hessian is positive definite even when $\mathbf{X}^\top\mathbf{X}$ is singular — strict convexity and a unique minimizer are guaranteed by the penalty.

Solution to Exercise 2 (R^2 monotone). Adding a column expands the column space $\mathcal{C}(\mathbf{X})$, so the projection of $\mathbf{y}$ onto it can only get closer (or stay equal) — the residual norm cannot increase. Since $R^2 = 1 - \text{RSS}/\text{TSS}$ and RSS is non-increasing while TSS is fixed, R^2 is non-decreasing. Hence R^2 cannot guide feature selection; use adjusted R^2 or out-of-sample error.

Solution to Exercise 3 (soft-threshold from subgradient). In the orthonormal case the objective separates as $\sum_j[\frac{1}{2}(\beta_j - \hat{\beta}_j^{\text{ols}})^2 + \lambda|\beta_j|]$. The subgradient condition $0 \in \beta_j - \hat{\beta}_j^{\text{ols}} + \lambda\partial|\beta_j|$ gives, for $\beta_j > 0$: $\beta_j = \hat{\beta}_j^{\text{ols}} - \lambda$ (valid when $\hat{\beta}_j^{\text{ols}} > \lambda$); symmetrically for $\beta_j < 0$; and $\beta_j = 0$ when $|\hat{\beta}_j^{\text{ols}}| \leq \lambda$ (the subgradient interval $[-\lambda, \lambda]$ contains $\hat{\beta}_j^{\text{ols}}$). This is exactly the soft-threshold operator.

Solution to Exercise 4 (variance and collinearity). $\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$; eigendecomposing $\mathbf{X}^\top\mathbf{X} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$ gives $(\mathbf{X}^\top\mathbf{X})^{-1} = \mathbf{V}\boldsymbol{\Lambda}^{-1}\mathbf{V}^\top$ with eigenvalues $1/\lambda_i$. Collinearity drives some $\lambda_i \to 0$, so $1/\lambda_i \to \infty$ — the coefficient variance along that eigen-direction explodes, the precise mechanism of unstable estimates.

Solution to Exercise 6 (OLS = Gaussian MLE). Under $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$, the log-likelihood is $-\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$; maximizing over $\boldsymbol{\beta}$ minimizes $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$, the OLS criterion. Under Laplace noise $p(\varepsilon) \propto e^{-|\varepsilon|/b}$, the log-likelihood involves $-\frac{1}{b}\sum|y_i - \mathbf{x}_i^\top\boldsymbol{\beta}|$, so the MLE minimizes the *absolute* error — least-absolute-deviations / median regression, robust to outliers.

33 Exercises

1. Derive the ridge estimator by minimizing the penalized objective and show the Hessian is $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I} \succ 0$, hence strict convexity even when $\mathbf{X}^\top\mathbf{X}$ is singular.
2. Prove R^2 is non-decreasing in added regressors using the projection picture.
3. Derive the soft-thresholding solution for lasso in the orthonormal case from the subgradient condition.
4. Show $\text{Var}(\hat{\boldsymbol{\beta}}_{\text{ols}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$ and explain via the eigenvalues why collinearity inflates it.
5. For the ridge SVD filter, compute $\text{df}(\lambda)$ for given singular values and sketch it as a function of λ.
6. Show OLS equals the Gaussian MLE and identify the loss implied by Laplace noise.

7. Explain, with the grouping-effect argument, why elastic net is preferable to lasso for correlated factor exposures.

End of Linear Regression & Regularization.