Supervised Learning: Regression

OLS, Regularization, Metrics, and Factor Models

Lecture Companion Notes

Data Science with Python – BSc Course

Joerg Osterrieder
April 1, 2026

These notes accompany the regression lectures (L21–L24). They follow a problem-first structure: each section opens with a concrete challenge, builds intuition through visuals and analogies, then formalizes the concept with worked examples. Read before lecture for preparation, revisit after for deeper understanding.

Contents

1 What Line Would You Draw? – Ordinary Least Squares Regression

Opening Problem: The Junior Analyst's First Day

You just started as a junior analyst at a quantitative hedge fund. Your desk is covered with Bloomberg terminals, and your manager drops a spreadsheet on your keyboard: twelve months of daily returns for Apple stock and the S&P 500 index. "Tell me how Apple responds to market moves," she says. "Fit a line. Quantify the relationship. Have it on my desk by lunch."

You open Python. You have 252 data points—one for each trading day. Each point has two numbers: the market return that day and Apple's return that day. Plotted together, they form a cloud of dots with a vaguely upward trend. You could draw a line through that cloud by hand, but which line? Eyeballing it feels arbitrary. Two analysts would draw two different lines. Your manager does not want your artistic impression. She wants the one line that is mathematically the best fit.

That line is the Ordinary Least Squares regression line. This section shows you how to find it, what it means, and why finance cares about it so much.

Discovery Question

If you and 100 classmates each draw a "best fit" line through the same scatter plot by eye, would any two of you draw exactly the same line? What makes one line objectively better than another?

The Scatter Plot Challenge

Suppose you plot Apple's daily return on the vertical axis and the S&P 500's return on the horizontal axis. You get a cloud of dots. Some days both go up. Some days both go down. Some days they disagree. There is a pattern—a general upward drift—but it is noisy.

Now imagine drawing a straight line through that cloud. If you tilt the line too steeply, it overshoots the low points and undershoots the high ones. If you make it too flat, it misses the trend entirely. Somewhere between those extremes lies a line that comes as close as possible to all the dots simultaneously.

Figure 1 shows exactly this situation. Look at it before reading any formula. Your eye already knows roughly where the line should go. OLS just makes that instinct precise.

Three classmates each draw their own line. All three look plausible. None of them agree. Figure 2 shows this dilemma: without a rule, "best fit" is a matter of opinion.

Think of it this way: fitting a line is like stretching a rubber band through a pinball machine. Each data point is a pin. The rubber band wants to get close to every pin, but it cannot pass through all of them because they are scattered. The position where the band settles—minimizing total stretch—is the regression line. The "stretch" at each pin is what we call a *residual*.

The OLS Objective: Minimize Squared Errors

Linear regression model: A model of the form $\hat{y} = \beta_0 + \beta_1 x$, where $\hat{y}$ is the predicted value, x is the input feature, β_0 is the intercept (the prediction when $x = 0$), and β_1 is the slope (the change in $\hat{y}$ per one-unit change in x).

Residual: The difference between the observed value and the predicted value: $e_i = y_i - \hat{y}_i$. A positive residual means the model underpredicted; a negative one means it overpredicted.

Figure 3 shows the key idea visually: each residual is a vertical distance from a data point to the line. OLS squares those distances and adds them up. The line that makes this sum as small

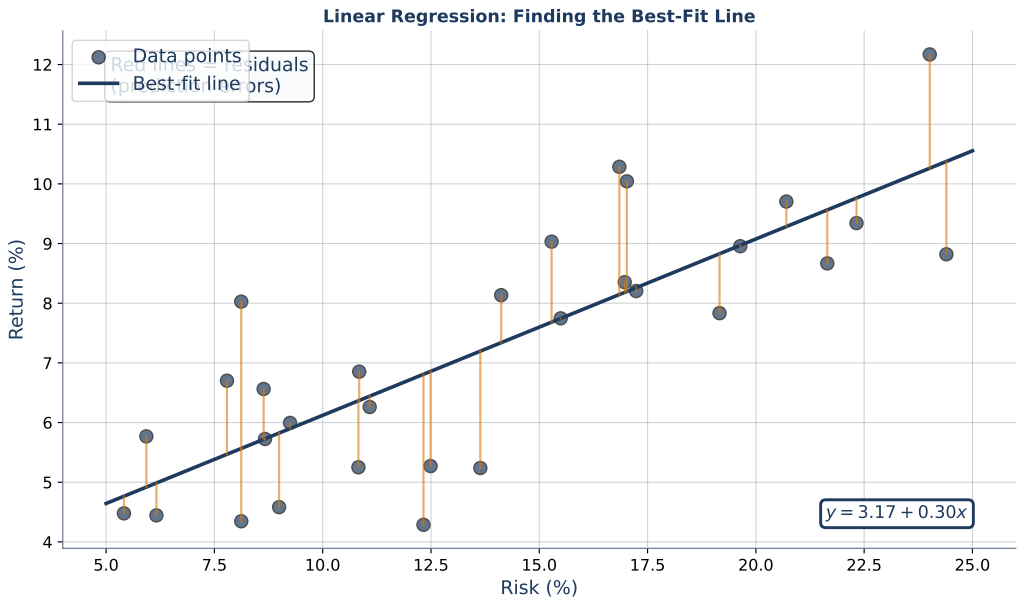


Figure 1: Stock returns vs. market returns. Where would you draw the line?

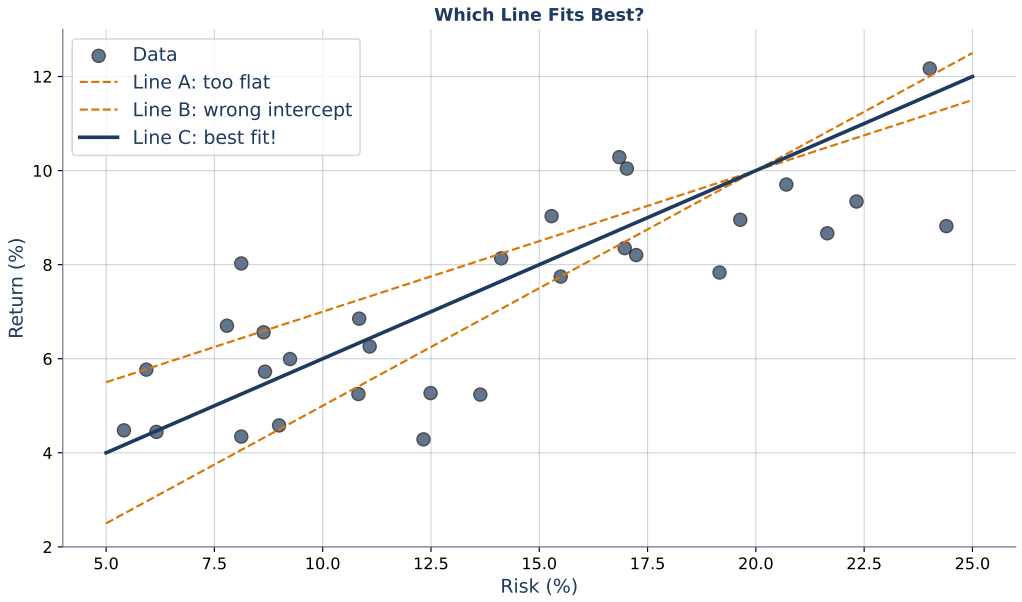


Figure 2: Three candidate lines through the same data. Which one is best? We need a rule.

as possible wins.

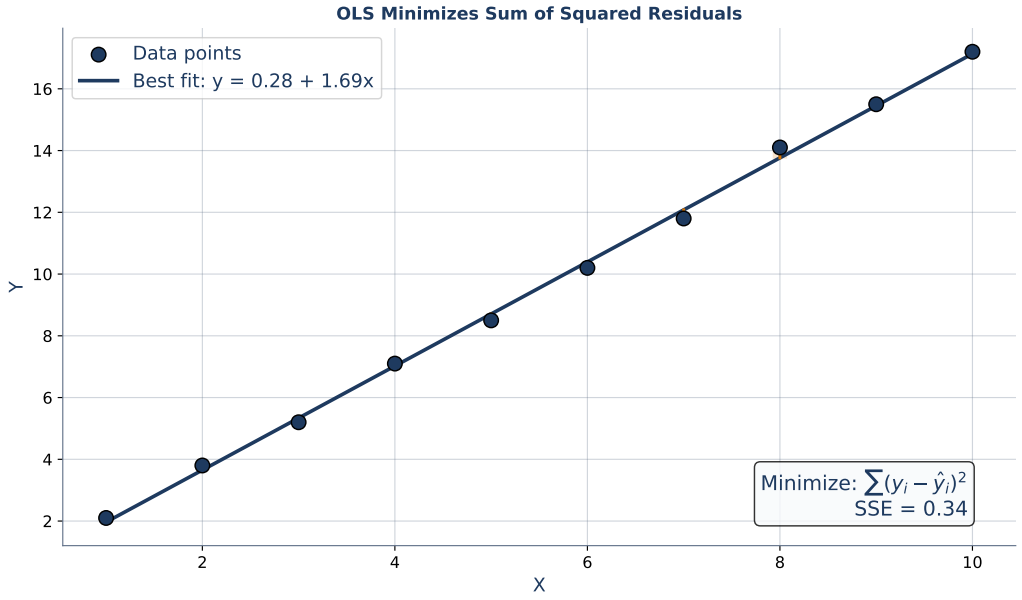


Figure 3: Squared errors visualized. Each red square has an area equal to the squared residual at that point. OLS minimizes the total red area.

Why squared? Three reasons. First, squaring makes all errors positive—a residual of -3 and a residual of $+3$ both contribute 9. Second, squaring penalizes large errors disproportionately: a residual of 6 contributes 36, four times more than a residual of 3. Third—and this is the mathematical payoff—squaring produces a smooth, differentiable function with a unique minimum. You can solve for the best line with a formula rather than trial and error.

Key Formula: The OLS Objective

OLS finds the slope β_1 and intercept β_0 that minimize the **sum of squared residuals** (SSR):

$$\text{SSR} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

where:

- n is the number of observations (e.g., 252 trading days)
- y_i is the actual stock return on day i
- x_i is the market return on day i
- $\hat{y}_i = \beta_0 + \beta_1 x_i$ is the predicted return on day i

Key Formula: The OLS Solution

Setting the derivatives of SSR to zero and solving gives the closed-form solution:

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\text{Cov}(X, Y)}{\text{Var}(X)}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

where:

- $\bar{x}$ is the mean of all x_i values (average market return)
- $\bar{y}$ is the mean of all y_i values (average stock return)
- $\text{Cov}(X, Y)$ measures how X and Y move together
- $\text{Var}(X)$ measures how spread out X is

Plain English: The slope is "how much Y responds when X moves," scaled by how much X actually moves.

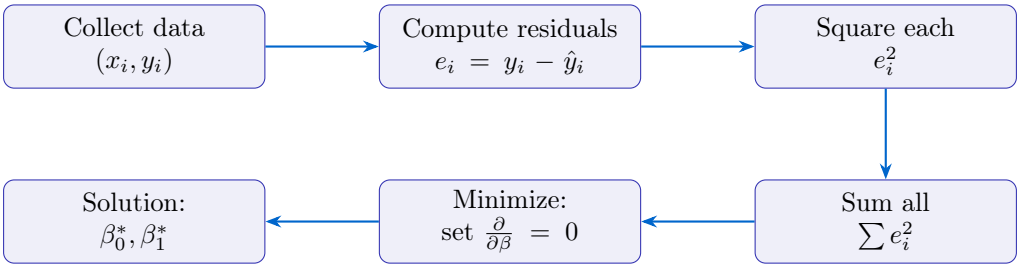
Ordinary Least Squares (OLS): The method of finding the line (or hyperplane) that minimizes the sum of squared residuals. "Ordinary" distinguishes it from weighted or generalized least squares. "Least squares" describes the objective function.

Definition: OLS Regression

Ordinary Least Squares regression fits a linear model $\hat{y} = \beta_0 + \beta_1 x$ by choosing β_0 and β_1 to minimize $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. The solution is unique (provided the data are not all identical in x) and has a closed-form expression. No iterative optimization is required.

Common Misconceptions about OLS

(1) "High R^2 means the model is good." Not necessarily. R^2 can be high because the model is overfitting the training data. A high R^2 on test data is more meaningful—but even then, a model can explain variance without being useful for prediction.
(2) "The regression line predicts individual outcomes." It predicts the *mean* outcome for a given x. Any individual observation will scatter around that mean. Confusing the conditional mean with individual predictions leads to overconfidence.
(3) "Regression proves causation." Regression quantifies association. If stock returns correlate with market returns, that does not prove the market *causes* the stock to move. Both might be driven by a third factor—interest rates, for example.



The diagram above shows the OLS pipeline: data goes in, residuals come out, squaring penalizes large errors, and calculus finds the unique minimum. This entire pipeline runs inside `model.fit(X, y)`.

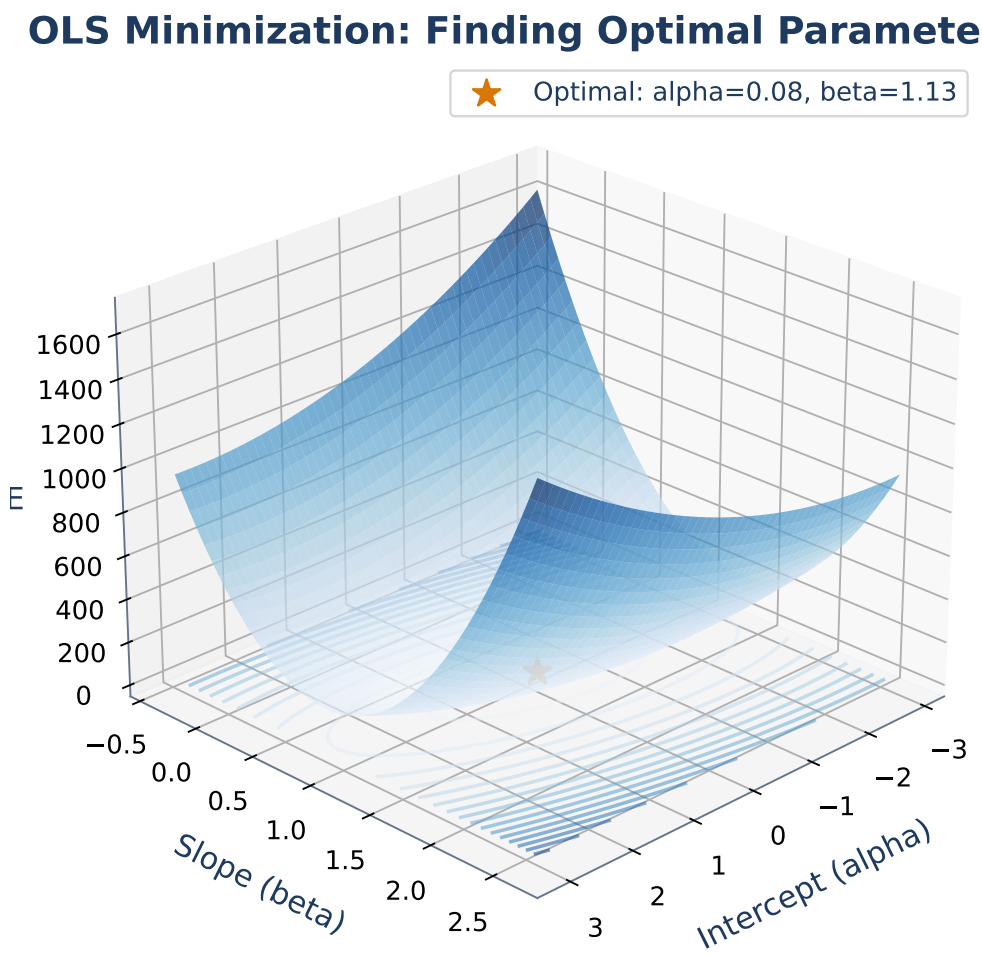


Figure 4: The OLS minimization process. As the slope and intercept change, the total squared error changes. OLS finds the valley of this error surface.

Worked Examples

Worked Example 1: Computing Slope and Intercept by Hand

A student collects five days of data:

Day	Market Return x_i (%)	Stock Return y_i (%)	$x_i - \bar{x}$
1	-2	-3	-2
2	-1	-1	-1
3	0	1	0
4	1	2	1
5	2	4	2

Step 1: Compute means. $\bar{x} = 0$, $\bar{y} = 0.6$.
Step 2: Compute the slope numerator: $\sum(x_i - \bar{x})(y_i - \bar{y}) = (-2)(-3.6) + (-1)(-1.6) + (0)(0.4) + (1)(1.4) + (2)(3.4) = 7.2 + 1.6 + 0 + 1.4 + 6.8 = 17.0$.
Step 3: Compute the slope denominator: $\sum(x_i - \bar{x})^2 = 4 + 1 + 0 + 1 + 4 = 10$.
Step 4: Slope: $\beta_1 = 17.0/10 = 1.7$. For every 1% the market moves, this stock moves 1.7%.
Step 5: Intercept: $\beta_0 = 0.6 - 1.7 \times 0 = 0.6$. Even when the market is flat, this stock tends to return 0.6%.
Fitted model: $\hat{y} = 0.6 + 1.7x$. If the market returns 3% tomorrow, the predicted stock return is $0.6 + 1.7(3) = 5.7\%$.

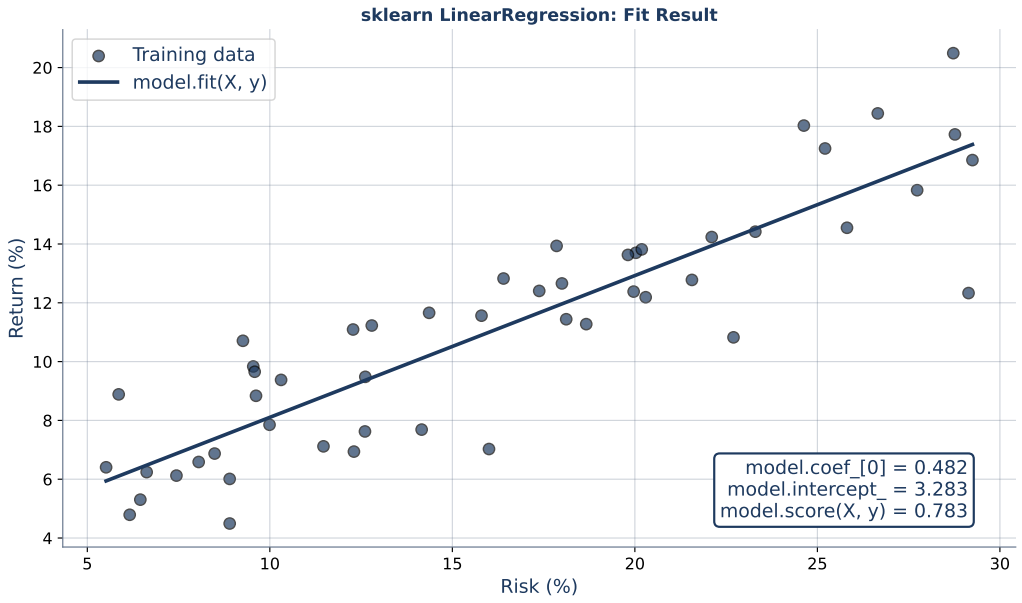


Figure 5: The sklearn `LinearRegression` fit: three lines of code produce the same result as the manual calculation.

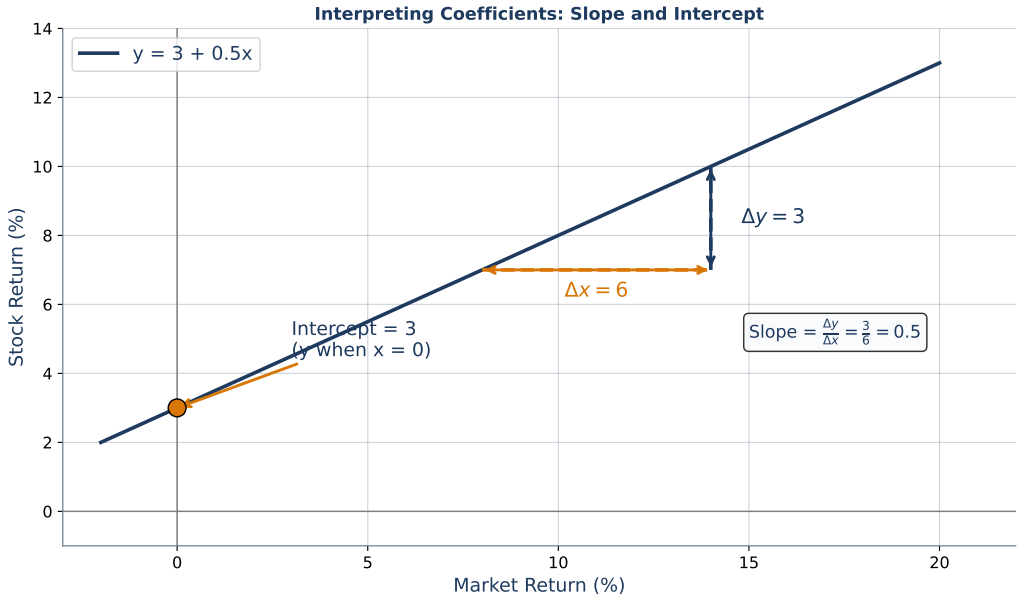


Figure 6: Interpreting the slope and intercept. The slope is the stock's sensitivity to the market; the intercept is the stock's return when the market is flat.

Worked Example 2: Comparing Two Stocks

Stock A has $\beta_1 = 0.8$ and $\beta_0 = 0.1\%$. Stock B has $\beta_1 = 1.5$ and $\beta_0 = -0.05\%$.
Interpretation: Stock A is defensive—when the market drops 2%, Stock A drops only $0.8 \times 2\% = 1.6\%$. Stock B is aggressive—a 2% market drop translates to $1.5 \times 2\% = 3.0\%$.
On an up day: If the market rises 1%, Stock A gains $0.1 + 0.8(1) = 0.9\%$, while Stock B gains $-0.05 + 1.5(1) = 1.45\%$. Stock B amplifies the good news.
On a down day: Stock B amplifies the bad news too. The slope is symmetric. A slope of 1.5 means 50% more volatile than the market—in both directions.

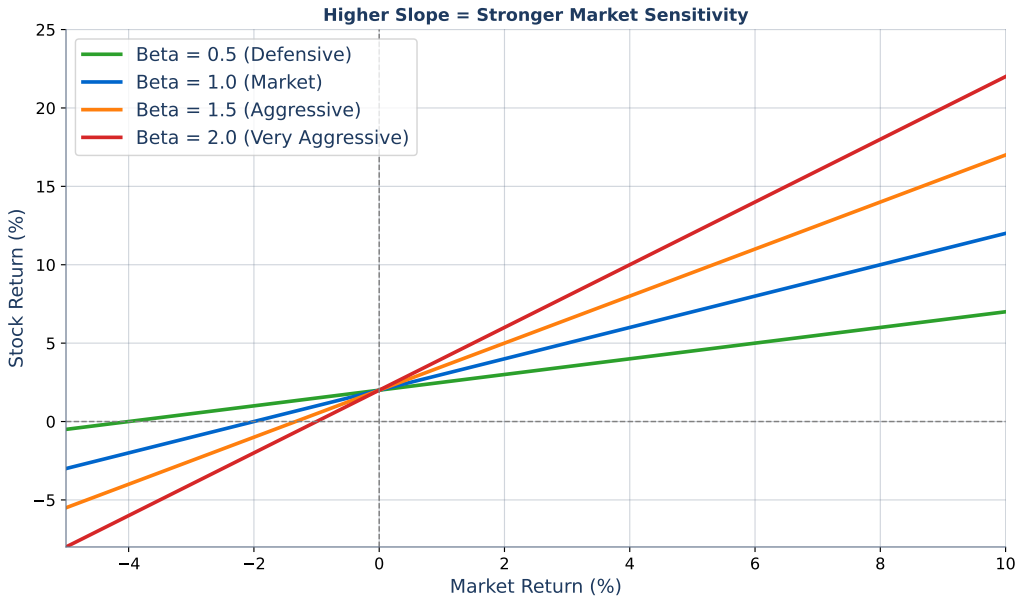


Figure 7: Different slopes tell different stories. Steeper lines mean greater sensitivity to market movements.

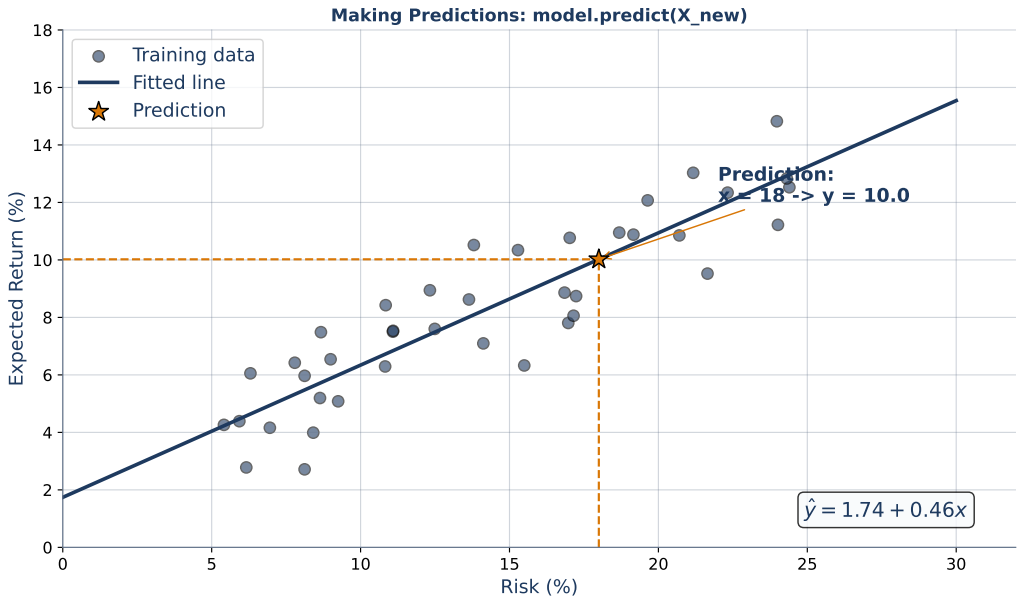


Figure 8: Making predictions: the model maps a market return to a predicted stock return via the fitted line.

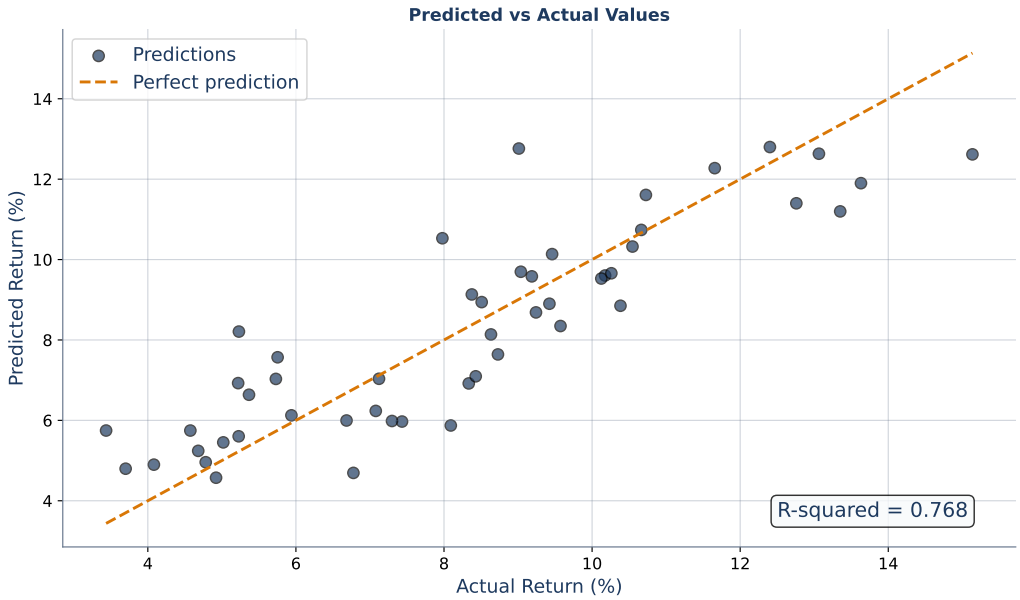


Figure 9: Predicted vs. actual values. Points on the diagonal indicate perfect predictions. Scatter around the diagonal indicates prediction error.

Historical Background: Gauss, Legendre, and a Missing Asteroid (1805–1809)

On January 1, 1801, the Italian astronomer Giuseppe Piazzi discovered Ceres—a small body orbiting between Mars and Jupiter. He tracked it for 41 days before it disappeared behind the Sun. Astronomers needed to predict where Ceres would reappear, but 41 data points from a noisy telescope were not much to work with.

Carl Friedrich Gauss, then 24 years old, applied a method he later called "least squares" to fit an orbit from Piazzi's sparse data. His prediction was spectacularly accurate: when Ceres reappeared in December 1801, it was within half a degree of Gauss's forecast.

Adrien-Marie Legendre published the method first, in 1805. Gauss published later, in 1809, but claimed he had been using it since 1795. The priority dispute lasted their lifetimes. The mathematics was the same either way: find the parameters that minimize the sum of squared errors.

What Gauss did with pencil and paper to find a lost asteroid, you do with `model.fit(X, y)` to find stock trends. The principle has not changed in two centuries.

Problem 1.1 (Easy)

Given five data points: $(1, 3)$, $(2, 5)$, $(3, 6)$, $(4, 8)$, $(5, 11)$. Compute $\bar{x}$, $\bar{y}$, $\text{Cov}(X, Y)$, $\text{Var}(X)$, and the OLS slope β_1 and intercept β_0.

Solution: see Appendix.

Problem 1.2 (Easy)

A regression model for a utility stock gives $\hat{y} = 0.2 + 0.6x$, where x is the market return in percent and $\hat{y}$ is the predicted stock return. Interpret the slope and intercept in plain English. Is this stock aggressive or defensive?

Solution: see Appendix.

Problem 1.3 (Medium)

Using the model from Problem 1.1, predict $\hat{y}$ when $x = 6$. The actual value turns out to be $y = 12$. Compute the residual. Is the model underpredicting or overpredicting?

Solution: see Appendix.

Problem 1.4 (Medium)

Model A has coefficients $\beta_0 = 0.5$, $\beta_1 = 1.2$ with $R^2 = 0.65$. Model B has $\beta_0 = -0.1$, $\beta_1 = 0.9$ with $R^2 = 0.72$. Which model fits the data better? Which stock is more market-sensitive? Can a model with a lower slope still have a higher R^2? Explain.

Solution: see Appendix.

Problem 1.5 (Hard)

Derive the OLS formula for β_1 by setting the partial derivative of $\text{SSR} = \sum(y_i - \beta_0 - \beta_1 x_i)^2$ with respect to β_1 equal to zero. You will need the result from setting $\frac{\partial \text{SSR}}{\partial \beta_0} = 0$ first (which gives $\beta_0 = \bar{y} - \beta_1 \bar{x}$). Substitute and simplify to show that $\beta_1 = \text{Cov}(X, Y)/\text{Var}(X)$.

Solution: see Appendix.

Connecting Forward

We now have a line. We can compute its slope, interpret it in finance terms, and use it to predict. But a line is only useful if it reflects reality. What if the true relationship is not linear? What if the residuals are not random but show a pattern? What if the spread of errors changes as the input changes?

These are questions about *assumptions*. OLS makes four of them, and Section 2 examines each one. Spoiler: violating them does not always break the model, but it always weakens your confidence in it.

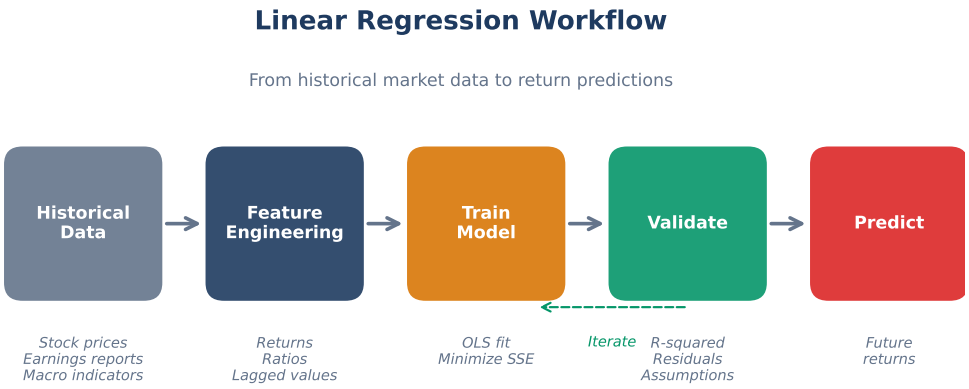


Figure 10: The full regression workflow: data collection, model fitting, diagnostics, and interpretation. We have covered fitting; diagnostics come next.

Key Takeaway: OLS regression finds the unique line that minimizes the sum of squared errors—there is no subjective "best fit," only the mathematically optimal one.

2 Can You Trust Your Line? – Residuals and the LINE Assumptions

Opening Problem: The Model That Worked Until It Didn't

Your regression model from Section 1 predicts Apple's returns with $R^2 = 0.72$ on one year of historical data. Impressive. Your manager approves a trading strategy based on the model's predictions.

The first month goes well. The second month is mediocre. By month three, the model is losing money on more trades than it wins. You re-check the code: no bugs. The data feed is correct. The model's coefficients have not changed. What went wrong?

You plot the residuals—the differences between predicted and actual returns. Instead of a random scatter, you see a clear pattern: the residuals grow larger as the predicted return increases. The model systematically underpredicts big moves. You have a case of *heteroscedasticity*: the variance of the errors is not constant. One of the LINE assumptions was violated, and nobody checked.

This section teaches you to check before trusting. A regression model is only as reliable as the assumptions behind it.

Discovery Question

Your regression model predicts stock returns with an R^2 of 0.85—but last month it lost money on every single trade. Which of the LINE assumptions probably failed, and how would you catch it before trading?

The Doctor's Checklist

Think of a doctor examining a patient. The patient might look healthy—normal weight, good color, steady pulse. But a thorough doctor checks blood pressure, listens to the heart, tests reflexes. Each test probes a different system. Only when all tests come back normal can the doctor say the patient is healthy.

Regression has its own checklist: the LINE assumptions. Each letter stands for one test. If all four pass, your model's predictions and confidence intervals are trustworthy. If any fail, you might still get useful predictions, but the standard errors, p-values, and confidence intervals become unreliable.

The residuals are your diagnostic tool. You do not examine the data directly—you examine the *leftover* pattern after the model has done its best. If the model captured all the systematic pattern, the residuals should look like random noise. Any remaining pattern in the residuals is a red flag.

The LINE Assumptions, One by One

The mnemonic LINE stands for Linearity, Independence, Normality, and Equal variance. Each assumption can be checked with a specific plot. For each one, we show what "good" looks like and what "violated" looks like.

L
Linearity
The true relationship is a straight line

I
Independence
Observations do not influence each other

N
Normality
Residuals follow a bell curve

E
Equal Variance
Residual spread is constant across x

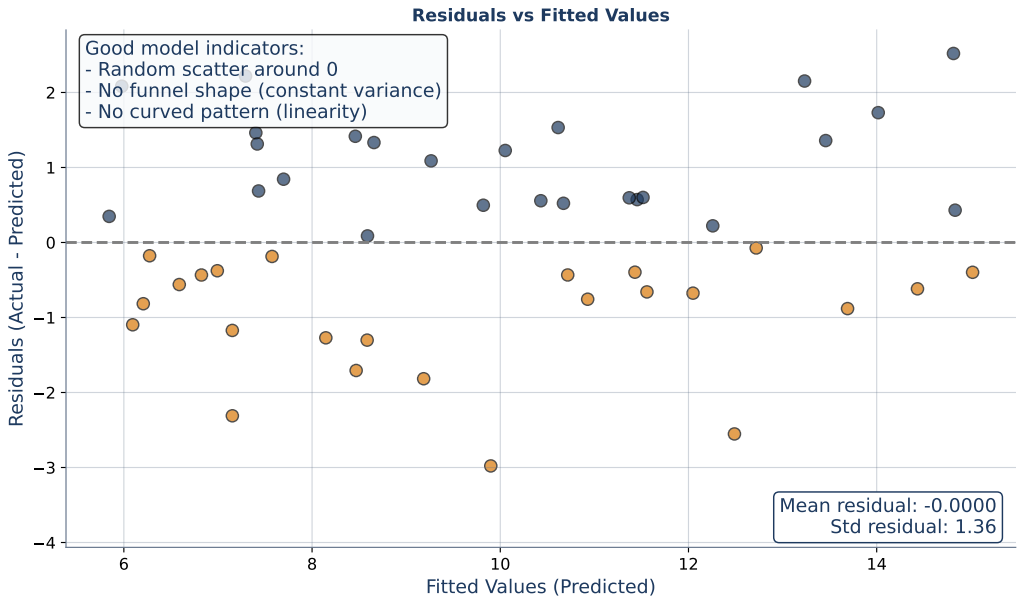


Figure 11: Residuals from an OLS fit. Each vertical line shows the error at one data point. Good residuals look like random noise centered at zero.

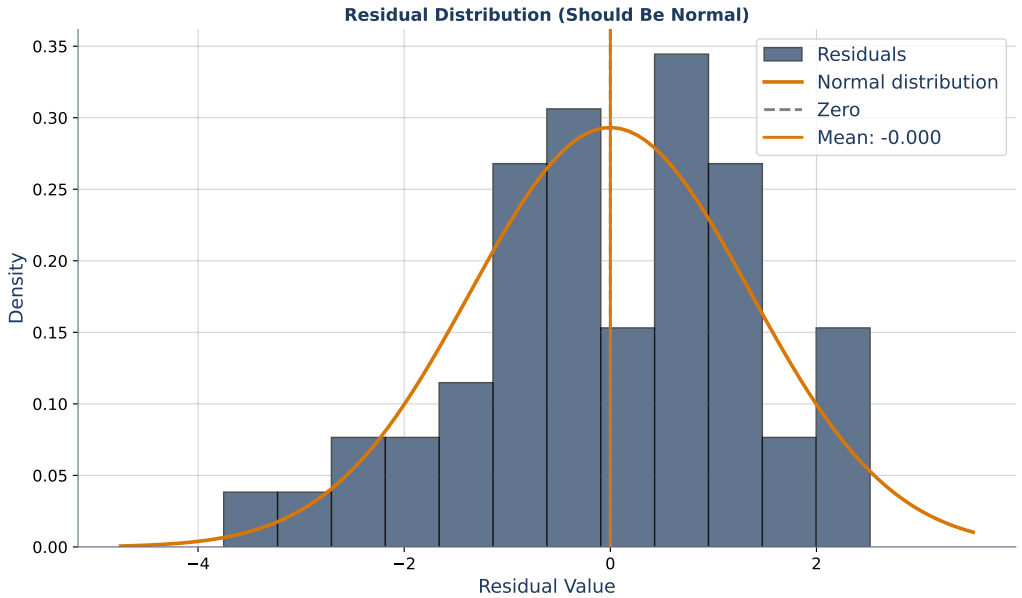


Figure 12: Histogram of residuals. A bell-shaped distribution centered at zero suggests the normality assumption holds.

Linearity assumption: The expected value of y is a linear function of x. If the true relationship is curved, a straight line will systematically miss the pattern and produce biased predictions.

Independence assumption: Each observation's error is unrelated to the errors of other observations. In time series data, today's stock return might correlate with yesterday's, violating independence. This is called autocorrelation.

Normality assumption: The residuals follow a normal (bell-curve) distribution. This assumption is needed for confidence intervals and hypothesis tests on the coefficients. It is *not* needed for the OLS estimates themselves to be unbiased.

Homoscedasticity (equal variance): The variance of the residuals is the same for all values of x. If the spread of errors grows or shrinks with x—a funnel shape in the residual plot—we say the errors are heteroscedastic, and standard errors become unreliable.

L – Linearity

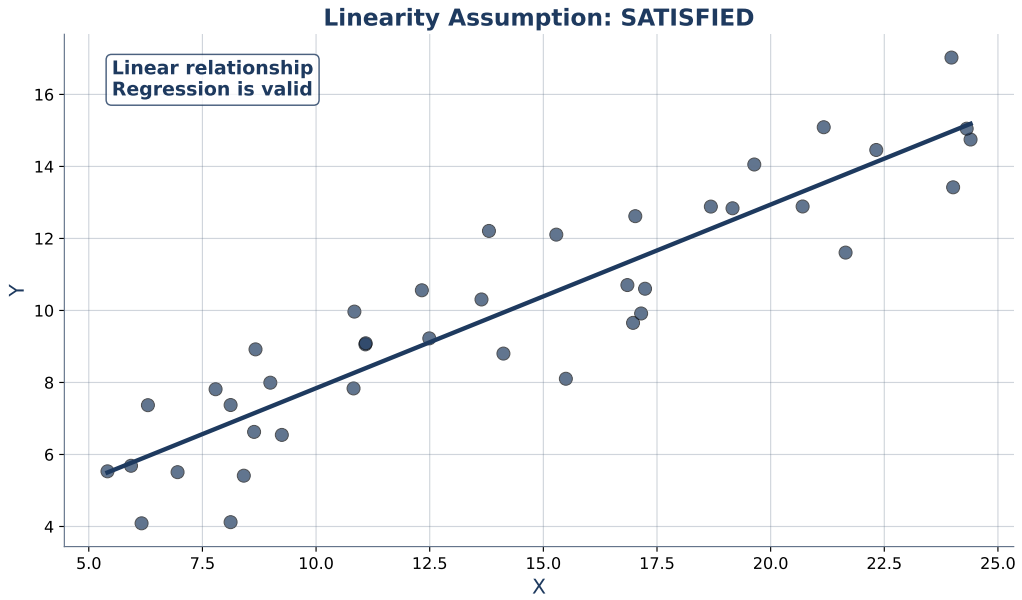


Figure 13: Linearity satisfied: residuals scatter randomly around zero with no curvature.

When linearity is violated, the fix is not to force a straight line harder. The fix is to change the model: add polynomial terms (x^2, x^3), apply a log transformation, or switch to a non-linear model entirely.

I – Independence

In finance, independence violations are extremely common. Stock returns on Tuesday correlate with returns on Monday. Ignoring this makes your standard errors too small, which makes your model look more precise than it actually is. The Durbin-Watson test can detect autocorrelation statistically.

N – Normality

A critical distinction: normality is needed for *inference* (p-values, confidence intervals), not for *estimation*. The OLS estimates are still the best linear unbiased estimates even without normality—that is the Gauss-Markov theorem. But if you want to test whether a coefficient is "significantly different from zero," you need normal residuals.

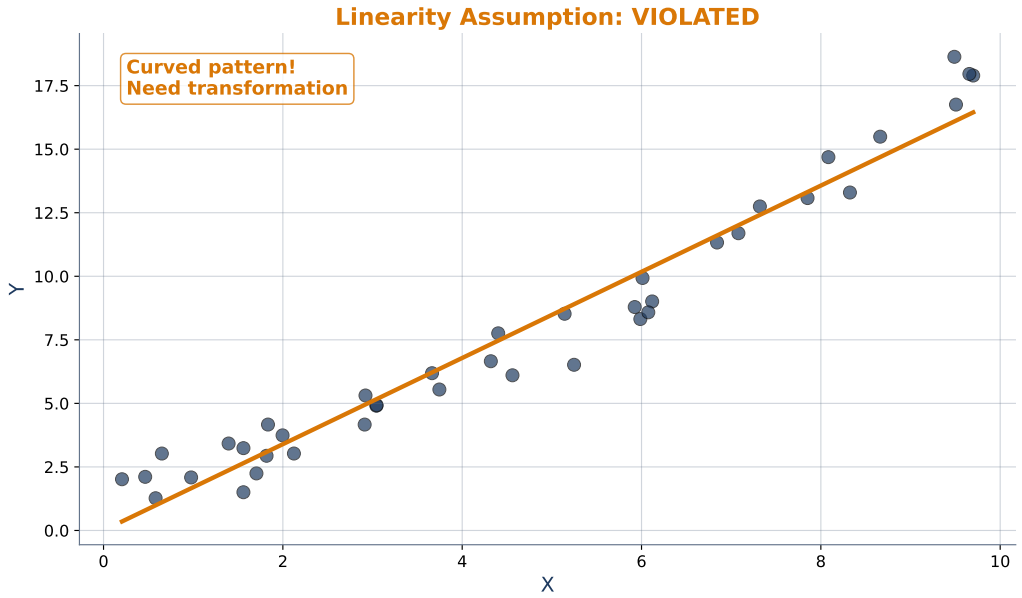


Figure 14: Linearity violated: residuals show a U-shaped pattern, indicating the true relationship is curved.

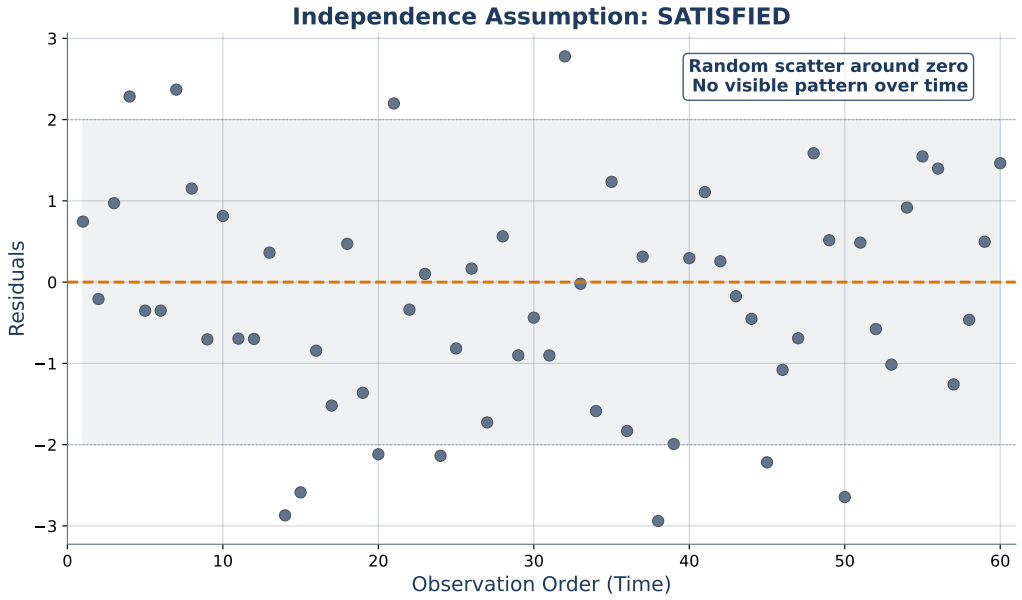


Figure 15: Independence satisfied: residuals show no pattern when plotted in time order.

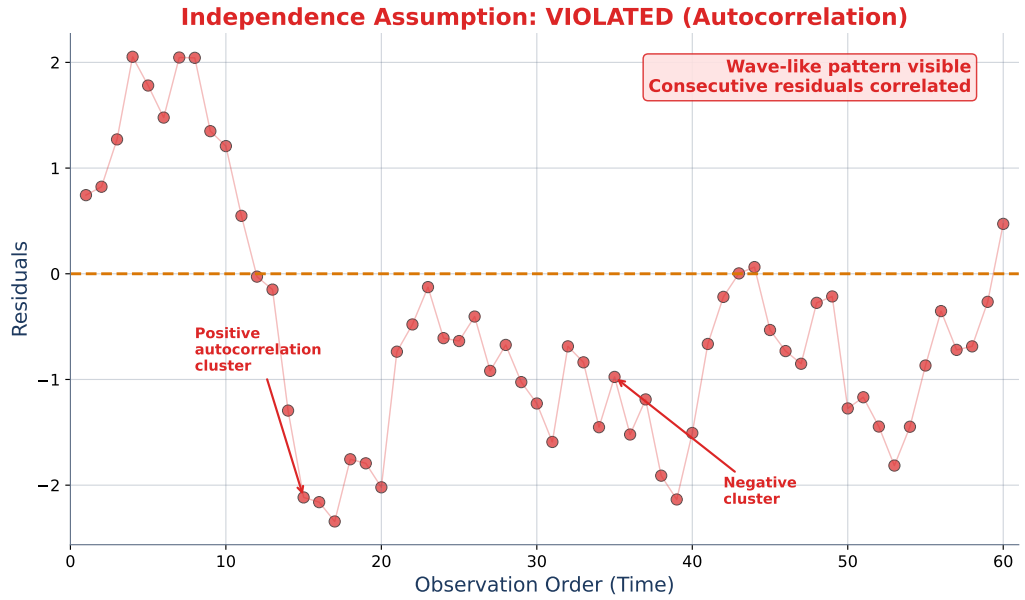


Figure 16: Independence violated: residuals cluster—positive errors follow positive errors, negative follow negative. This is autocorrelation.

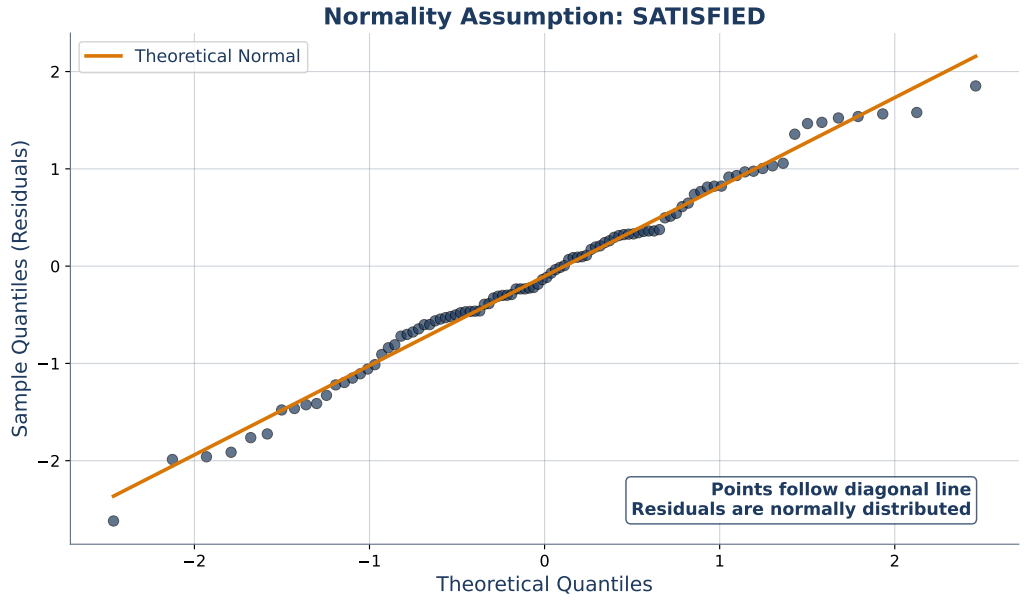


Figure 17: Normality satisfied: residuals form a symmetric bell curve centered at zero.

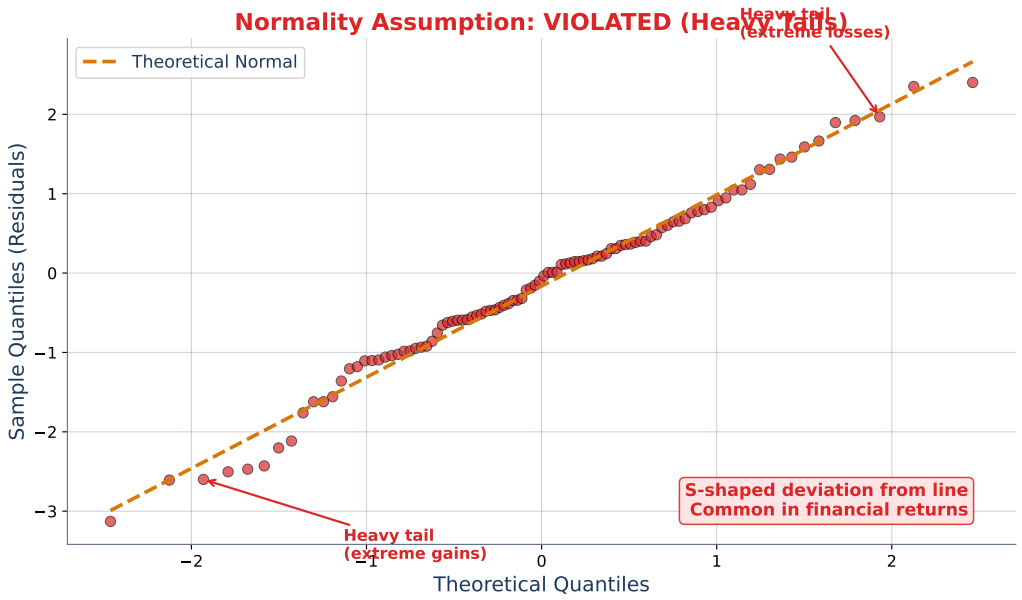


Figure 18: Normality violated: residuals are skewed or have heavy tails. Coefficient estimates remain unbiased, but confidence intervals become untrustworthy.

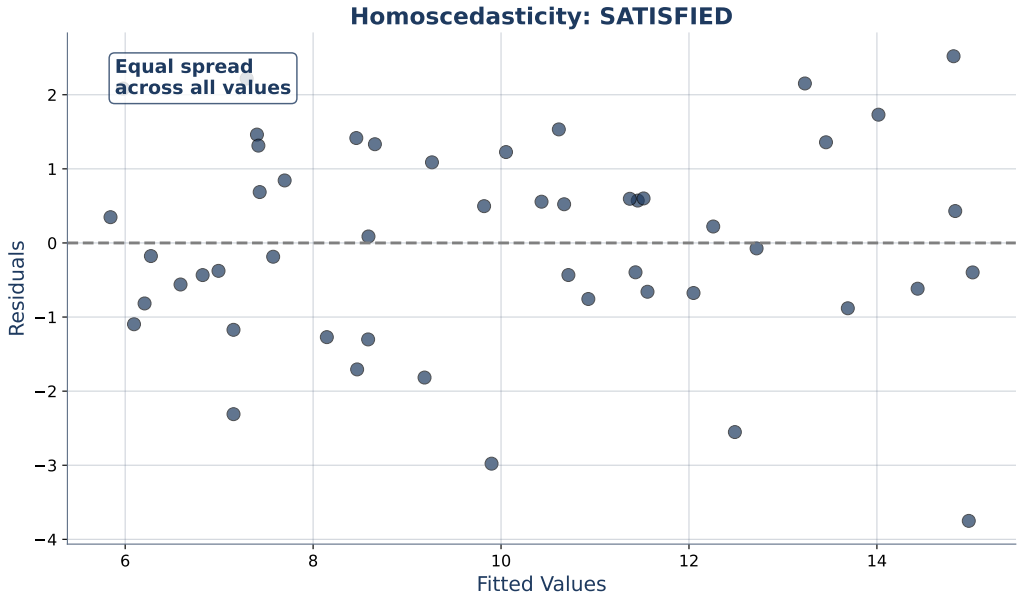


Figure 19: Equal variance satisfied: the spread of residuals is roughly the same across all fitted values.

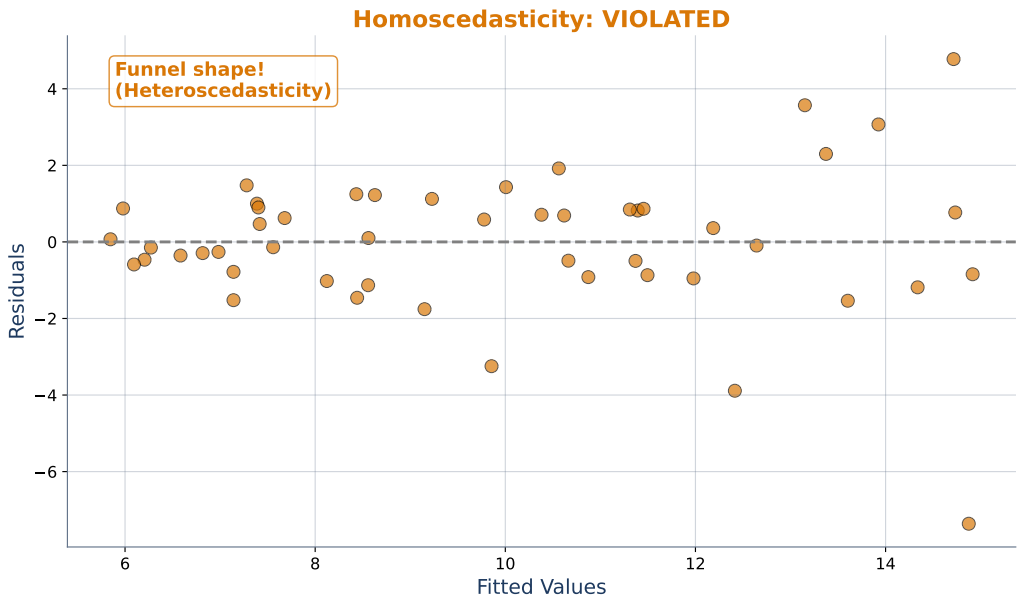


Figure 20: Equal variance violated: the residual spread fans out as fitted values increase (funnel shape). This is heteroscedasticity.

E – Equal Variance (Homoscedasticity)

Heteroscedasticity is the norm in finance. Large stocks have different volatility than small stocks. Returns during crises are more volatile than returns during calm periods. Fixes include log-transforming the response variable, using weighted least squares, or computing robust (Huber-White) standard errors that do not assume constant variance.

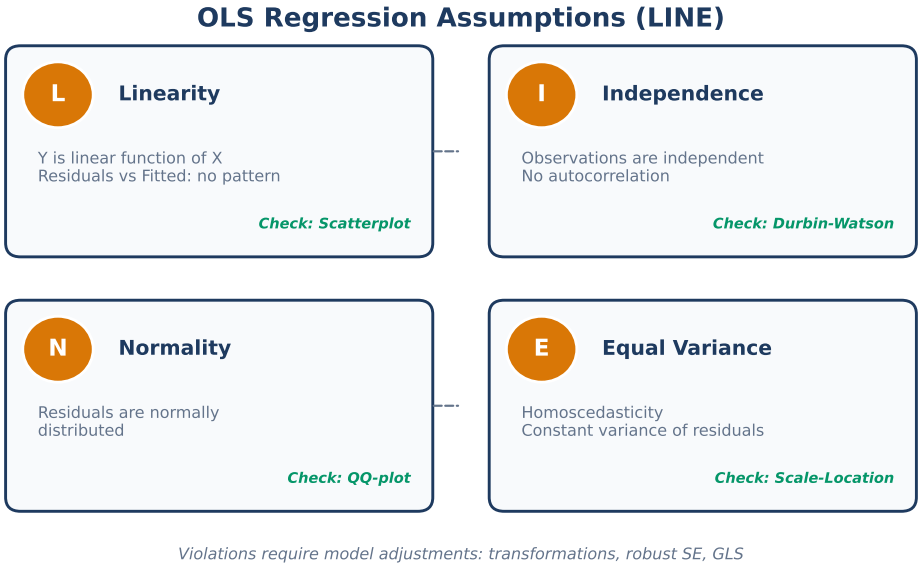


Figure 21: All four LINE assumptions at a glance: "good" on the left, "violated" on the right for each.

Common Misconceptions about LINE

(1) "If residuals look random, all assumptions hold." Randomness in a residual-vs-fitted plot checks linearity and equal variance. It does not check normality (use a QQ plot) or independence (plot residuals in time order).
(2) "Normality of residuals is needed for the model to work." It is only needed for inference—p-values and confidence intervals. The coefficient estimates themselves are unbiased and efficient (among linear estimators) even without normality.
(3) "Outliers should always be removed." Outliers might be data entry errors, or they might be genuine extreme events—a flash crash, a surprise earnings announcement. Removing real outliers because they are inconvenient is bad science. Investigate first, then decide.

Diagnostic Plots in Practice

Worked Example 1: Reading a Residual-vs-Fitted Plot

You fit a regression of stock returns on market returns and plot residuals against fitted values. You observe:

- The residuals scatter roughly evenly above and below zero—good, linearity holds.
- The spread of residuals is wider on the right side of the plot than on the left—bad, equal variance is violated.

Diagnosis: Heteroscedasticity. The model predicts small returns accurately but is less precise for large predicted returns.
Action: Use robust standard errors (Huber-White) when computing confidence intervals. Alternatively, apply a log transformation to the response if returns are strictly positive.

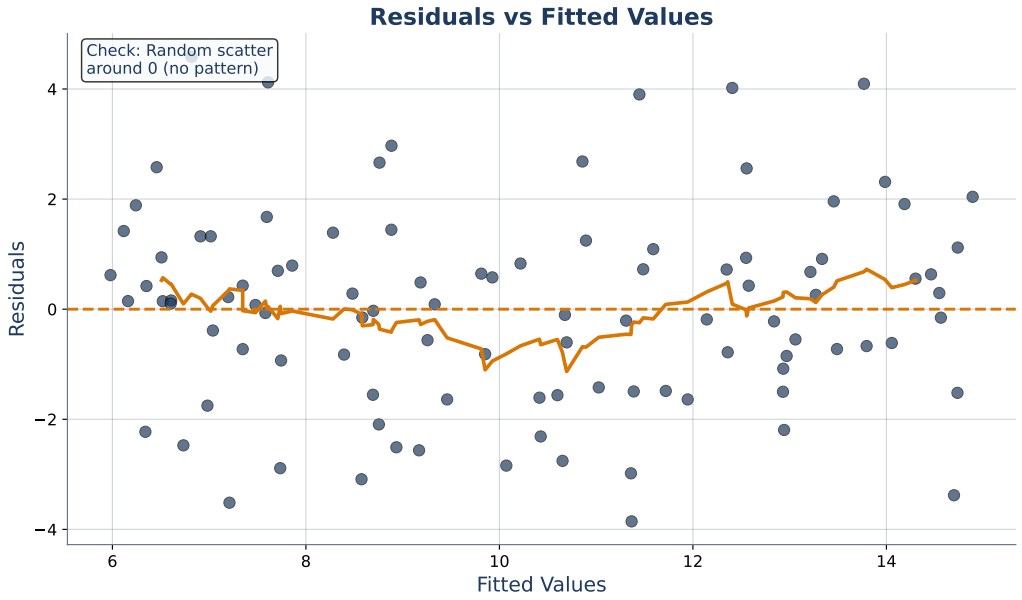


Figure 22: Diagnostic plot: residuals vs. fitted values. Look for patterns (curvature = linearity violated) and changing spread (funnel = heteroscedasticity).

Worked Example 2: Reading a QQ Plot

A QQ (quantile-quantile) plot compares the distribution of your residuals to a theoretical normal distribution. Each point plots one residual's quantile against the corresponding normal quantile.

If normality holds: The points line up along the 45-degree diagonal.

If the tails are heavy: The points curve away from the diagonal at both ends—the residuals have more extreme values than a normal distribution predicts. This is common with stock returns, where market crashes produce residuals far larger than a bell curve would suggest.

If the distribution is skewed: The points curve away from the diagonal on one side, forming an S-shape.

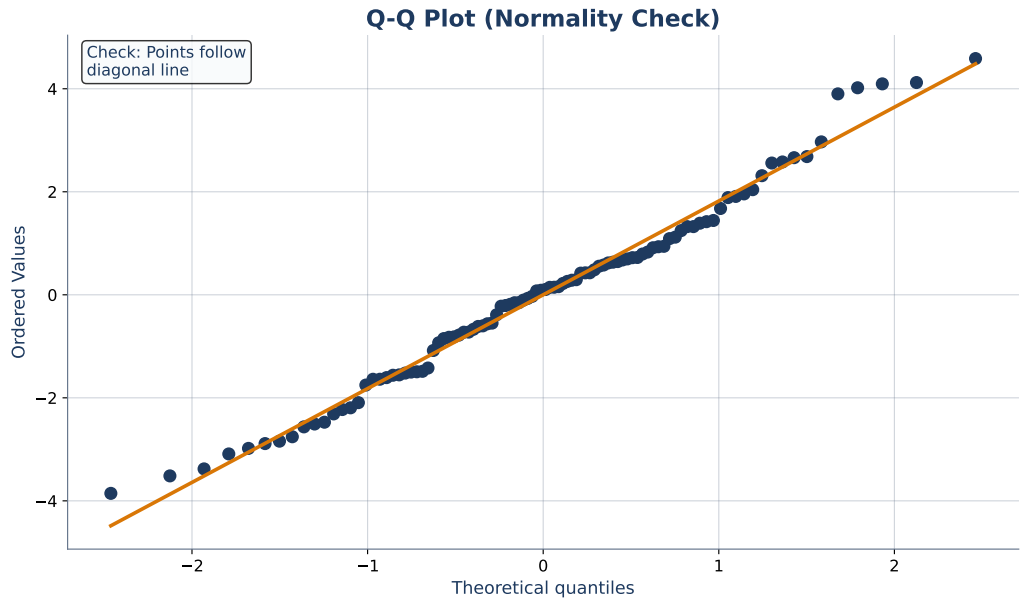


Figure 23: QQ plot: points on the diagonal indicate normally distributed residuals. Deviations at the tails suggest heavy tails or skewness.

Historical Background: Galton and the Origin of "Regression" (1886)

Francis Galton was a Victorian polymath—explorer, meteorologist, fingerprint pioneer, and Charles Darwin's half-cousin. In the 1880s he studied the inheritance of traits in sweet pea plants, and later in humans. He noticed something that initially puzzled him: children of very tall parents tended to be shorter than their parents, and children of very short parents tended to be taller.

He called this "regression toward mediocrity"—later softened to "regression to the mean." The word "regression" stuck as the name for the statistical method, even though the phenomenon Galton described (extreme values reverting toward the average) is a different concept from line-fitting.

Galton's insight was that correlation between parent and child height is real but imperfect. A correlation of 0.5 means that children are, on average, halfway between their parents' height and the population mean. The regression line captures this imperfect inheritance. The modern name is an accident of history, but it reminds us that even the inventor of the method was surprised by what the data showed.

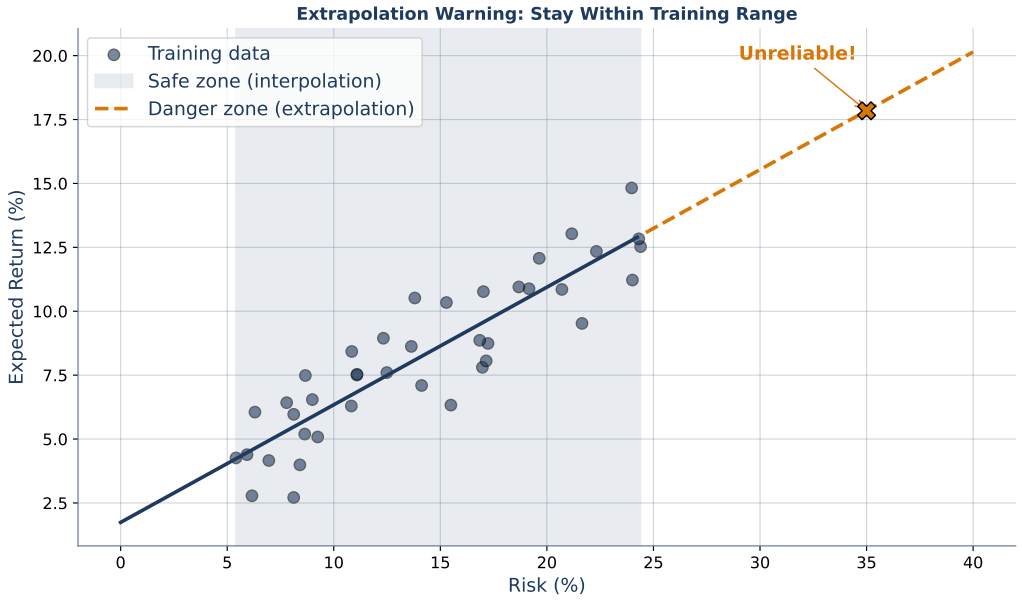


Figure 24: Extrapolation danger: the model's predictions become unreliable outside the range of observed data. In finance, predicting during market crashes from calm-market training data is a form of extrapolation.

Problem 2.1 (Easy)

For each of the following residual plot descriptions, identify which LINE assumption is violated:

(a) Residuals form a clear U-shape when plotted against fitted values.

(b) Residuals fan out in a funnel shape from left to right.

(c) Residuals show a wave pattern when plotted in time order (day 1, day 2, day 3, ...).

(d) The histogram of residuals has a long right tail.

Solution: see Appendix.

Problem 2.2 (Easy)

You are given a QQ plot of residuals. The points lie close to the diagonal in the middle, but curve sharply upward at the right end and downward at the left end. What does this tell you about the distribution of residuals? Which LINE assumption is affected, and what practical consequence does this have?
Solution: see Appendix.

Problem 2.3 (Medium)

A model of daily stock returns produces the following five residuals (in percentage points): $+0.8, +1.2, +0.9, -0.1, -0.5$. Notice the first three are positive and occur on consecutive days.

(a) Compute the mean of these residuals.

(b) Do they appear independent? Why or why not?

(c) Which assumption is most likely violated?

Solution: see Appendix.

Problem 2.4 (Medium)

A financial analyst models house prices as a function of square footage. The residual plot shows a funnel shape: small houses have small residuals, large houses have large residuals.

(a) Name this phenomenon.

(b) Explain why it makes sense economically (hint: consider the price range of large vs. small houses).

(c) Suggest two statistical fixes.

Solution: see Appendix.

Problem 2.5 (Hard)

Explain why autocorrelated residuals make standard OLS standard errors too small. (Hint: think about what happens to the effective sample size when consecutive observations carry the same information.)
Solution: see Appendix.

Connecting Forward

We now have a model (Section 1) and a way to check whether we should trust it (Section 2). But what about the data itself? What if we have 50 potential predictors instead of one? OLS will happily fit all 50, giving each a non-zero coefficient. The model might achieve an R^2 of 0.99 on the training data. That sounds fantastic—until you test it on new data and the R^2 drops to 0.10.

This collapse is called *overfitting*. The model memorized the noise in the training data instead of learning the signal. Section 3 explains exactly how and why this happens, and Section 4 provides the mathematical tools to prevent it.

Key Takeaway: A regression model is only as trustworthy as its assumptions—always check residual plots before trusting predictions.

3 Your Model Memorized the Noise – Overfitting and the Bias-Variance Tradeoff

Opening Problem: The Quant Who Lost Money with a Perfect Model

A quantitative analyst at a hedge fund builds a model to predict daily stock returns. She uses 30 features: momentum indicators, volume ratios, volatility measures, sentiment scores, macroeconomic variables. On the training data—two years of daily observations—the model achieves zero prediction error. Every single historical return is predicted perfectly.

She presents the results to her portfolio manager. He is skeptical: "If you can predict returns with zero error, we should be the richest fund on Wall Street. Let us paper-trade it for a month."

One month later, the paper-trading results come back. The model's predictions are barely better than random. The R^2 on live data is 0.02. The model that was perfect on history is useless on the future.

The analyst made a classic mistake: she built a model so complex that it memorized every quirk and accident in the training data. The accidental patterns—noise—did not repeat. The real patterns—signal—were there, but they were buried under a mountain of overfitted complexity.

Discovery Question

You built a model that predicts tomorrow's stock price with zero error on historical data. Your colleague says it is worthless. Who is right, and why?

The Study Habits Analogy

Imagine two students preparing for a statistics exam. Student A memorizes every practice problem and its exact answer: "Question 3 on Practice Exam 2 has answer B." Student B studies the underlying concepts: how to compute a mean, when to use a t-test, why the central limit theorem works.

On a practice exam, Student A scores 100%. She has seen every question before and recalls each answer. Student B scores 85%—good, but not perfect.

Now the real exam arrives. The questions are similar in spirit but different in detail. Student A freezes. She has never seen these exact questions. Student B applies the concepts she learned and scores 80%.

Student A is the overfitting model. Student B is the well-regularized model. Training accuracy is not the goal. Generalization is the goal.

Figure 25, Figure 26, and Figure 27 show the visual equivalent: a model that is too simple (underfitting), one that captures the pattern (good fit), and one that chases every data point (overfitting).

The Bias-Variance Decomposition

Overfitting: A model that performs well on training data but poorly on unseen data. The model has learned noise specific to the training set that does not generalize.

Underfitting: A model that performs poorly on both training and test data. The model is too simple to capture the underlying pattern.

Bias: The error from simplifying assumptions in the model. A model with high bias consistently misses the true pattern in the same direction. A straight line fitted to curved data has high bias.

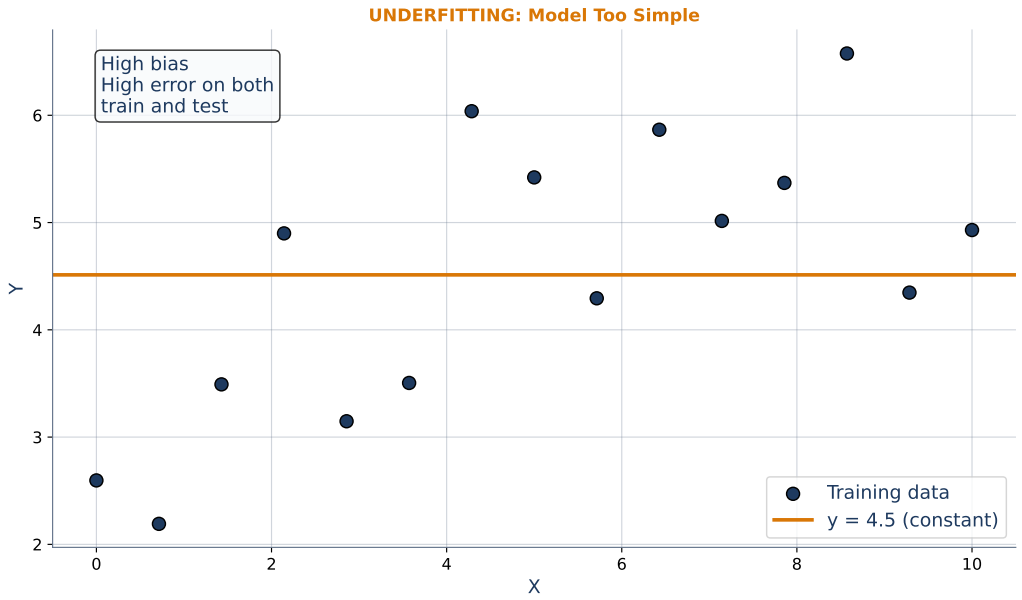


Figure 25: Underfitting: the model is too simple. It misses the underlying pattern and performs poorly on both training and test data.

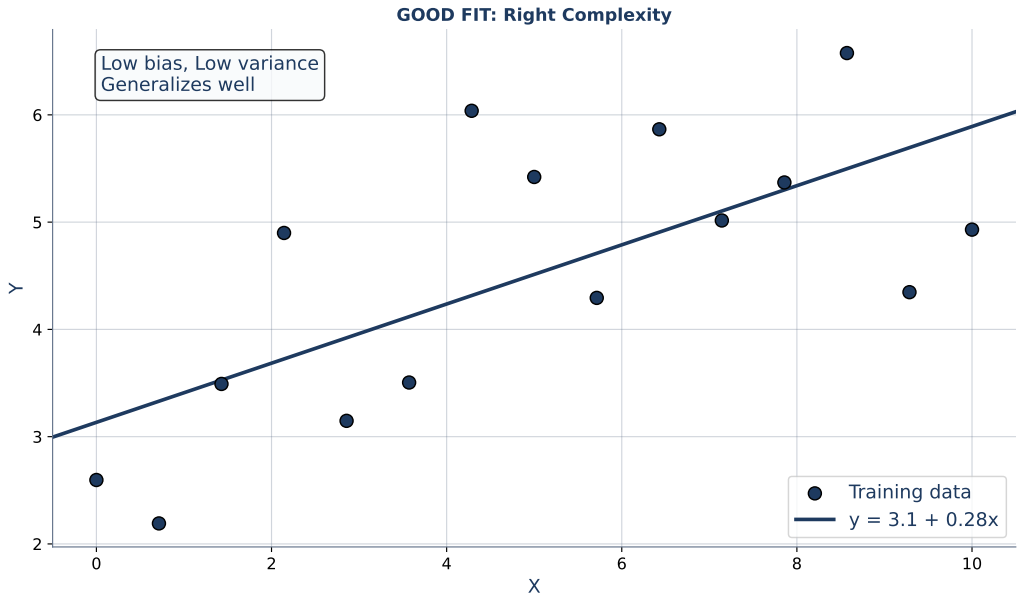


Figure 26: Good fit: the model captures the underlying pattern without chasing noise. It generalizes to new data.

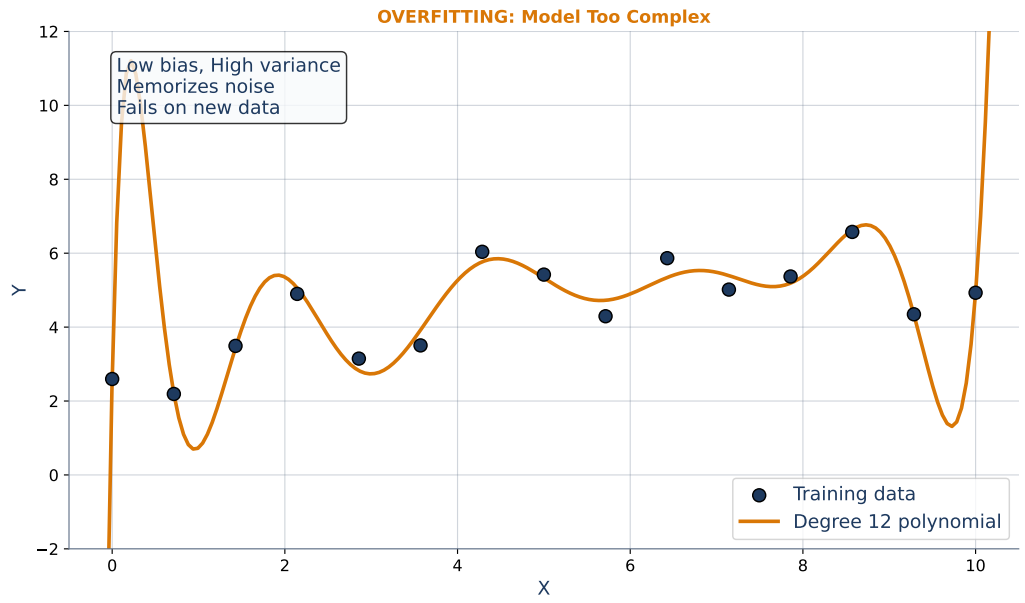


Figure 27: Overfitting: the model is too complex. It passes through every training point but wiggles wildly between them, failing on new data.

Variance: The error from sensitivity to small fluctuations in the training data. A model with high variance produces very different predictions when trained on slightly different datasets. A high-degree polynomial has high variance.

Irreducible error: The noise inherent in the data that no model can eliminate. Even a perfect model cannot predict the random component of stock returns.

Key Formula: The Bias-Variance Decomposition

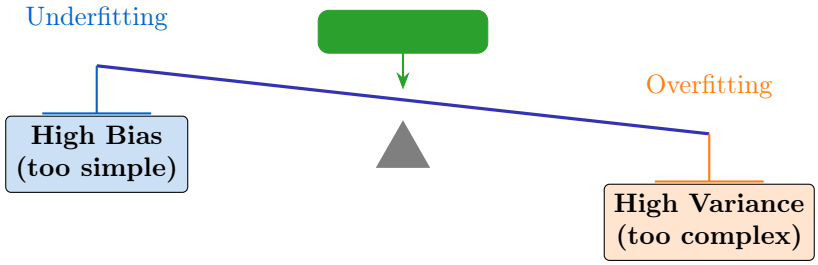
For any model, the expected prediction error decomposes as:

$$\text{Expected Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

where:

- $\text{Bias}^2 = \left[\mathbb{E}[\hat{f}(x)] - f(x)\right]^2$: how far the average prediction is from the truth
- $\text{Variance} = \mathbb{E}\left[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2\right]$: how much predictions vary across different training sets
- Irreducible error $= \sigma^2$: the variance of the noise ϵ

The tradeoff: As model complexity increases, bias decreases (the model can represent more patterns) but variance increases (the model becomes more sensitive to the specific training data). The optimal model minimizes the *sum*.



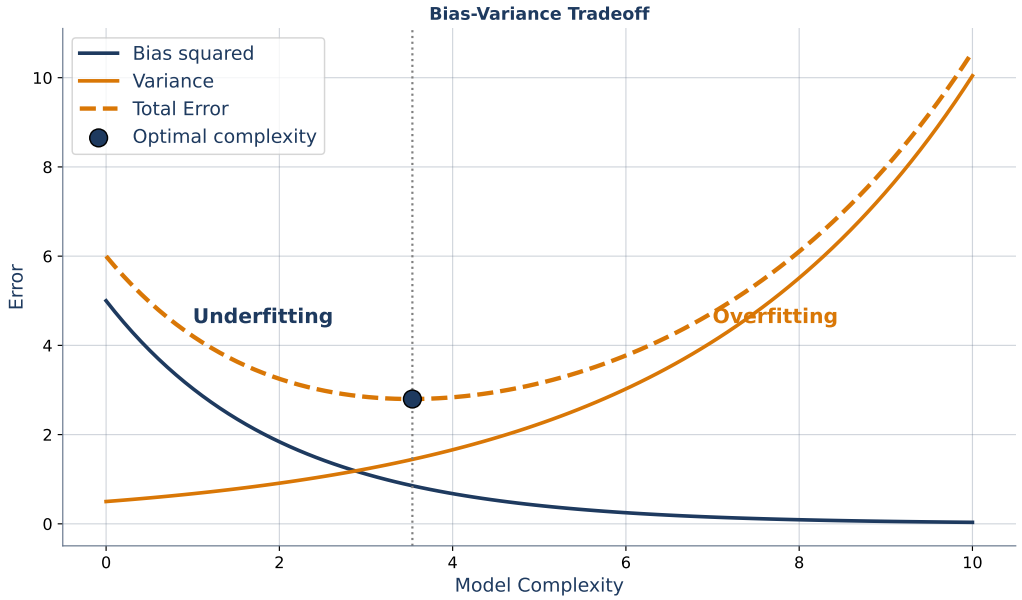


Figure 28: The bias-variance tradeoff. As complexity increases, bias drops but variance rises. Total error has a minimum somewhere in the middle.

The balance scale above captures the core dilemma. Push too far left (simple models) and you get high bias. Push too far right (complex models) and you get high variance. Regularization is the tool that lets you position the fulcrum.

Definition: Bias-Variance Tradeoff

The bias-variance tradeoff states that reducing one component of prediction error (bias or variance) typically increases the other. Model selection is the art of finding the complexity level where total error—bias squared plus variance plus irreducible noise—is minimized.

Common Misconceptions about Overfitting

(1) "A training error of 0 means the model is perfect." A training error of zero means the model has memorized the training data. On new data, it will almost certainly perform worse. Zero training error is a symptom of overfitting, not of perfection.
(2) "More features always improve the model." Each additional feature gives the model more flexibility to fit noise. If the feature carries no signal, it increases variance without reducing bias. The net effect is worse generalization.
(3) "Bias and variance can both be minimized simultaneously." They cannot, except by reducing irreducible error (which is impossible by definition). Every model faces a tradeoff: simpler means higher bias but lower variance; more complex means lower bias but higher variance.

Seeing Overfitting in Practice

Worked Example 1: Training vs. Test Error

A student fits polynomial regression models of degrees 1, 3, 5, 10, and 20 to 50 data points generated from a true cubic relationship with noise.

Degree	Training RMSE	Test RMSE
1	4.2	4.5
3	2.1	2.4
5	1.8	2.6
10	0.9	5.1
20	0.01	12.3

Observation: Training RMSE decreases monotonically as degree increases. Test RMSE decreases from degree 1 to 3 (the true relationship is cubic), then increases. The degree-20 model achieves near-zero training error but catastrophic test error.
Diagnosis: Overfitting begins at degree 5 and is severe at degree 20. The optimal model is degree 3—matching the true data-generating process.

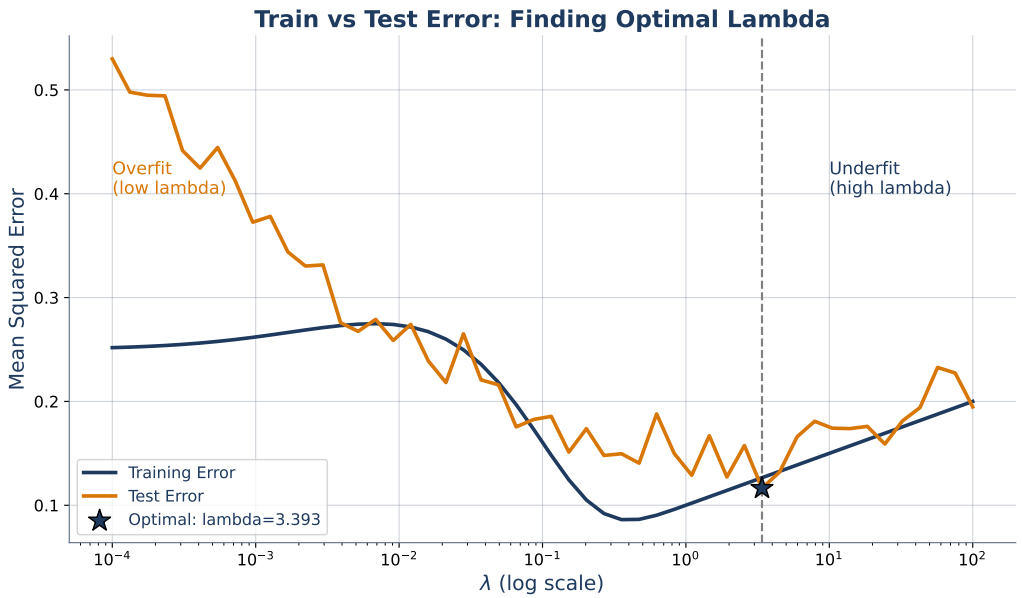


Figure 29: Training error vs. test error as model complexity increases. The gap between them is the overfitting gap.

Worked Example 2: The Overfitting Gap in Finance

A quant team builds three models to predict monthly stock returns using 20 fundamental factors:

Model A (OLS, all 20 factors): Training $R^2 = 0.45$, Test $R^2 = 0.08$. The gap is 0.37—most of the "explanatory power" was noise.

Model B (OLS, 5 selected factors): Training $R^2 = 0.22$, Test $R^2 = 0.15$. The gap is 0.07—much smaller. Fewer features, less overfitting.

Model C (Ridge, all 20 factors): Training $R^2 = 0.28$, Test $R^2 = 0.18$. Ridge uses all 20 factors but shrinks their coefficients, reducing variance.

In finance, an out-of-sample R^2 of 0.15 is considered good for return prediction. The models with smaller gaps between training and test performance are the ones you would actually trade.

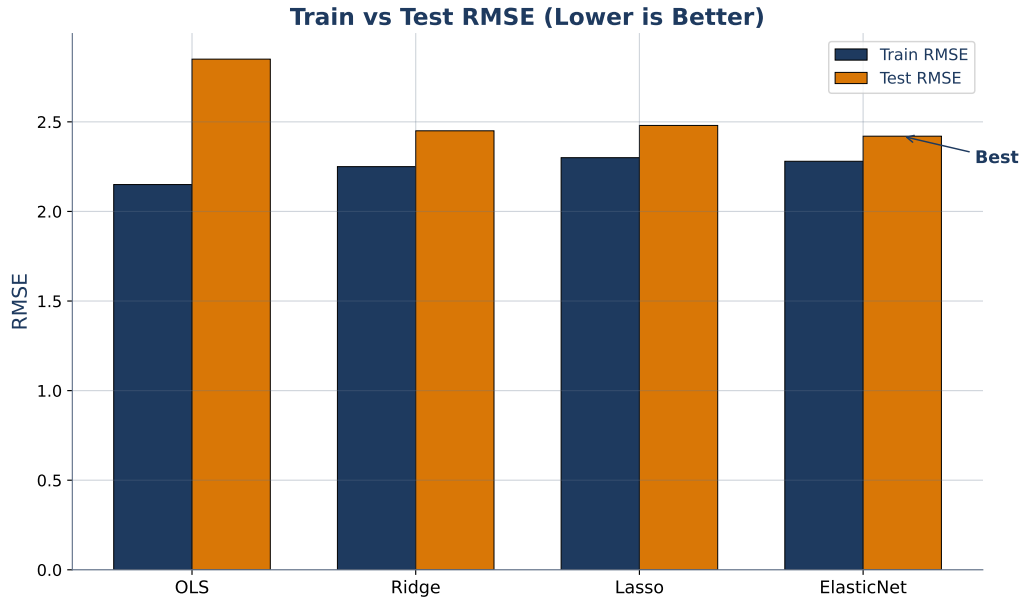


Figure 30: RMSE on training vs. test data across different model complexities. The divergence point marks where overfitting begins.

Historical Background: Geman, Bienenstock, and Doursat (1992)

The bias-variance decomposition was known informally for decades, but Stuart Geman, Elie Bienenstock, and René Doursat gave it a rigorous mathematical treatment in their 1992 paper "Neural Networks and the Bias/Variance Dilemma" (Neural Computation). At the time, neural networks were experiencing their first wave of popularity, and practitioners were confused about why larger networks did not always perform better.

Geman, Bienenstock, and Doursat proved that prediction error decomposes into exactly three additive terms: bias squared, variance, and irreducible noise. They showed that this decomposition applies to *any* estimator, not just neural networks. Their framework gave practitioners a language for discussing model complexity: "bias" for systematic error from simplification, "variance" for instability from data sensitivity.

The paper's central message remains as relevant today as it was in 1992: more parameters do not guarantee better predictions. Every added parameter reduces bias but increases variance. The art of modeling is finding the balance.

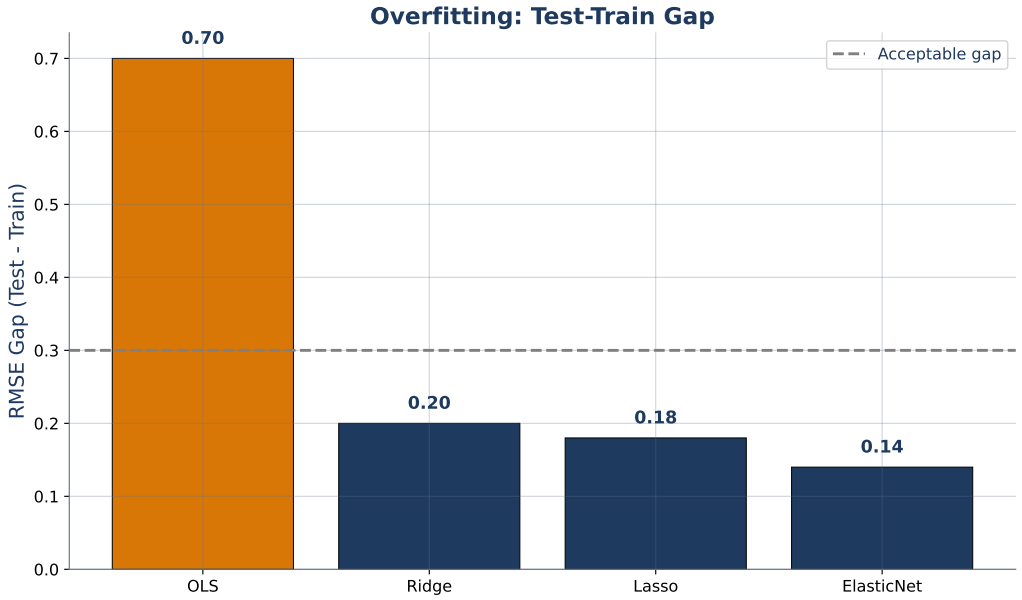


Figure 31: The overfitting gap quantified: the shaded area between training and test curves represents wasted complexity.

Problem 3.1 (Easy)

A student trains three models on the same dataset and records:

Model	Training Error	Test Error
Linear	5.2	5.8
Polynomial (degree 5)	2.1	3.0
Polynomial (degree 15)	0.3	9.4

Which model overfits? Which model underfits? Which model achieves the best generalization?
Solution: see Appendix.

Problem 3.2 (Easy)

Explain the bias-variance tradeoff in your own words, using a non-technical analogy (not the student/exam analogy from the text). Your analogy should make clear why reducing one increases the other.
Solution: see Appendix.

Problem 3.3 (Medium)

You train the same linear regression model on 10 different random subsets of your data (each subset is 80% of the full dataset, sampled randomly). The estimated slope β_1 varies widely: 0.8, 1.3, 0.6, 1.5, 0.9, 1.1, 0.7, 1.4, 0.8, 1.2.

(a) Compute the mean of these slope estimates (an estimate of $\mathbb{E}[\hat{\beta}_1]$).

(b) Compute the variance of these slope estimates.

(c) If the true slope is $\beta_1 = 1.0$, compute the bias.

(d) Is this model suffering more from bias or variance?

Solution: see Appendix.

Problem 3.4 (Medium)

Design an experiment to demonstrate overfitting to a classmate who has never taken a statistics course. You have access to Python, numpy, and matplotlib. Describe:

(a) What data you would generate (how many points, what relationship, how much noise).

(b) What models you would fit (which complexities).

(c) What plot you would show to make overfitting visually obvious.

Solution: see Appendix.

Problem 3.5 (Hard)

Starting from the definition $\text{MSE}(\hat{f}(x)) = \mathbb{E}[(y - \hat{f}(x))^2]$ and writing $y = f(x) + \epsilon$ where $\mathbb{E}[\epsilon] = 0$ and $\text{Var}(\epsilon) = \sigma^2$, prove that:

$$\text{MSE} = \text{Bias}^2(\hat{f}(x)) + \text{Var}(\hat{f}(x)) + \sigma^2$$

Hint: add and subtract $\mathbb{E}[\hat{f}(x)]$ inside the square, expand, and use the fact that ϵ is independent of $\hat{f}$.
Solution: see Appendix.

Connecting Backward and Forward

In Section 1 we built the model. In Section 2 we checked its assumptions. Now we have seen what goes wrong when the model is too complex: it memorizes noise instead of learning signal, and its predictions collapse on new data.

The natural next question is: can we keep using many features but prevent the model from overfitting? Can we mathematically force the model to stay simple?

Yes. The technique is called *regularization*, and it works by adding a penalty for complexity to the OLS objective. Instead of minimizing just the sum of squared residuals, we minimize the residuals *plus* a cost for having large coefficients. Section 4 shows you the two most important forms—Ridge and Lasso—and explains when to use each one.

Key Takeaway: A model that perfectly fits training data has learned the noise, not the signal—

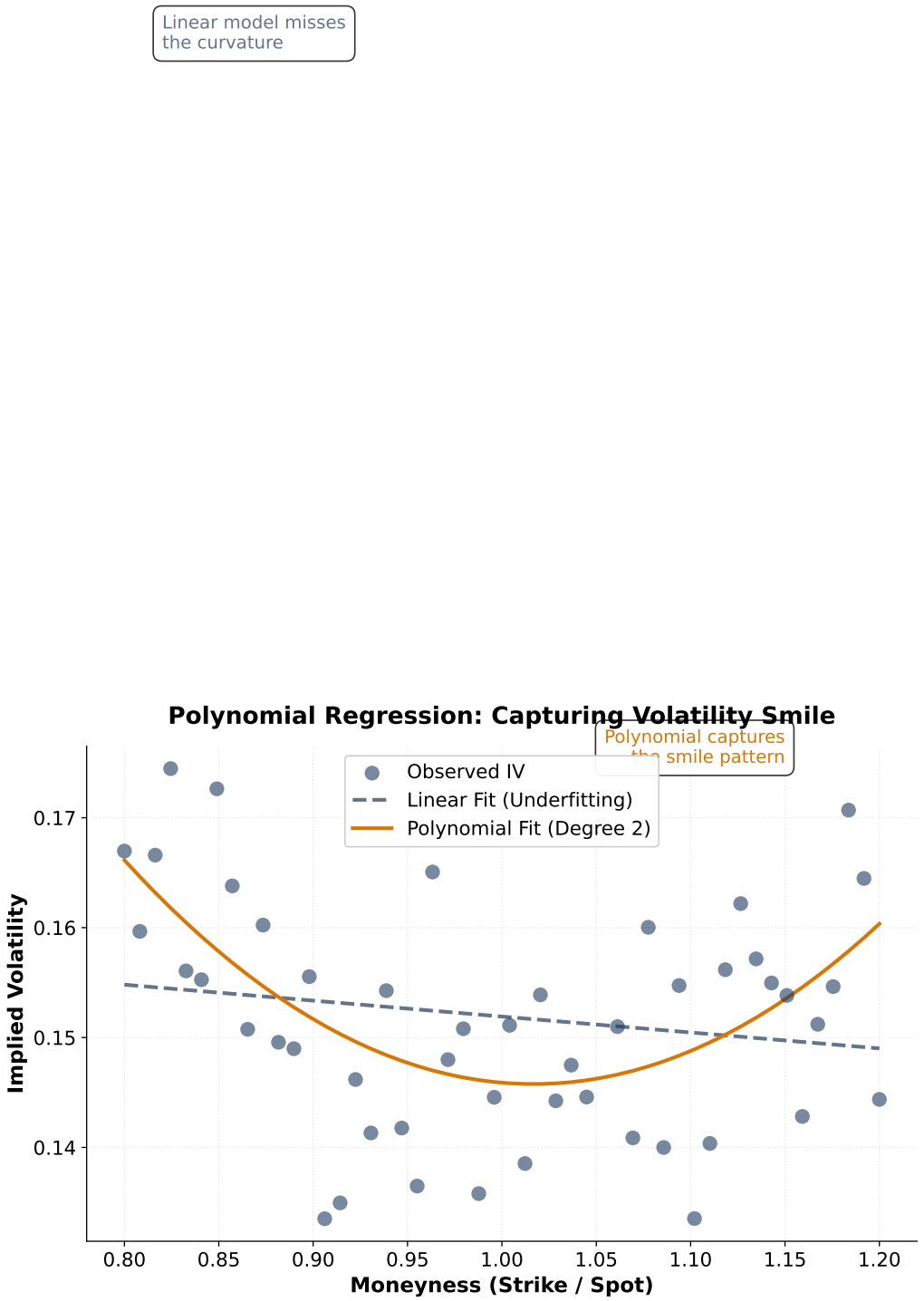


Figure 32: Polynomial features as a complexity dial: degree 1 is simple (high bias), degree 15 is complex (high variance). Regularization offers a gentler way to control this dial.

the goal is to generalize, not to memorize.

4 Mathematical Guardrails – Ridge, Lasso, and ElasticNet

Opening Problem: 200 Factors and No Idea Which Matter

A portfolio manager at an asset management firm wants to predict stock returns. Her research team has compiled a database of 200 potential predictors: price-to-earnings ratios, momentum signals, analyst sentiment, macroeconomic indicators, sector dummies, and dozens more.

She runs an OLS regression with all 200 features. The model returns 200 non-zero coefficients. Some are large, some are tiny, and many have the wrong sign—economic theory says the coefficient on earnings yield should be positive, but the model gives it a negative value of -0.003.

The portfolio manager suspects most of the 200 features are noise. She wants a model that *automatically* decides which features matter and sets the rest to zero. At the same time, she wants to keep features that are genuinely predictive, even if their individual contribution is small. And she wants the model to be stable: running it on slightly different data should give roughly the same answer.

OLS cannot do this. It treats all 200 features equally and fits each one with abandon. She needs a model with guardrails—mathematical constraints that keep coefficients under control. Those guardrails are called regularization.

Discovery Question

You have 200 potential predictors for stock returns. OLS gives you 200 non-zero coefficients, most of them tiny. How do you decide which predictors actually matter—and can you make the model decide for you?

The Volume Knob and the Suitcase

Two analogies capture the two flavors of regularization.

Ridge is a volume knob. Imagine a mixing board in a recording studio with 200 sliders, one per instrument (feature). OLS cranks every slider to wherever it fits the training data best—even if some sliders amplify noise. Ridge adds a constraint: *minimize the total volume.* All sliders get turned down. None go to zero, but the loudest ones get dialed back the most. The result is a quieter, smoother mix.

Lasso is a suitcase packer. You have 200 items to pack for a trip, but your suitcase only holds 20. You cannot take everything in miniature—you must choose. Important items go in at full size; unimportant ones stay home. Lasso does the same: it sets some coefficients to exactly zero, effectively removing those features from the model. The features that remain get their full (or near-full) coefficient values.

Figure 33 and Figure 34 show these constraints geometrically. The circle (Ridge) and the diamond (Lasso) define the regions where coefficients are allowed to live. The OLS solution sits outside these regions; regularization pulls the solution back inside.

Why does the diamond shape cause exact zeros but the circle does not? The diamond has sharp corners on the axes. The error contours (ellipses centered at the OLS solution) are more likely to first touch a corner than a smooth curve. At a corner, one or more coefficients are exactly zero. The circle has no corners—every point on its surface has all coefficients non-zero.

Ridge Regression (L2 Penalty)

Ridge regression: A regularized version of OLS that adds a penalty equal to the sum of squared coefficients. The objective is $\min \sum(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$. Ridge shrinks all coefficients toward zero but never sets any to exactly zero.

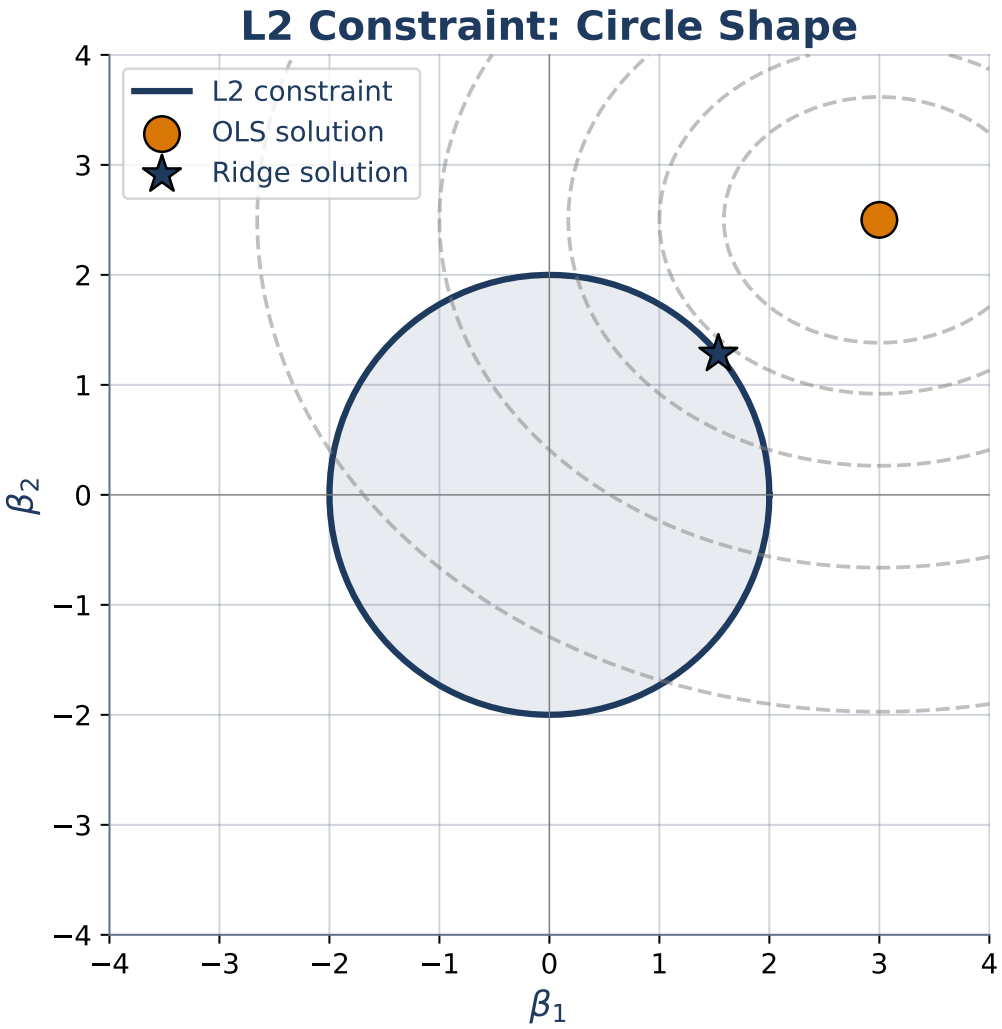


Figure 33: Ridge constraint: coefficients must stay inside a circle (L2 ball). The error contours touch the circle at a point where all coefficients are non-zero but shrunken.

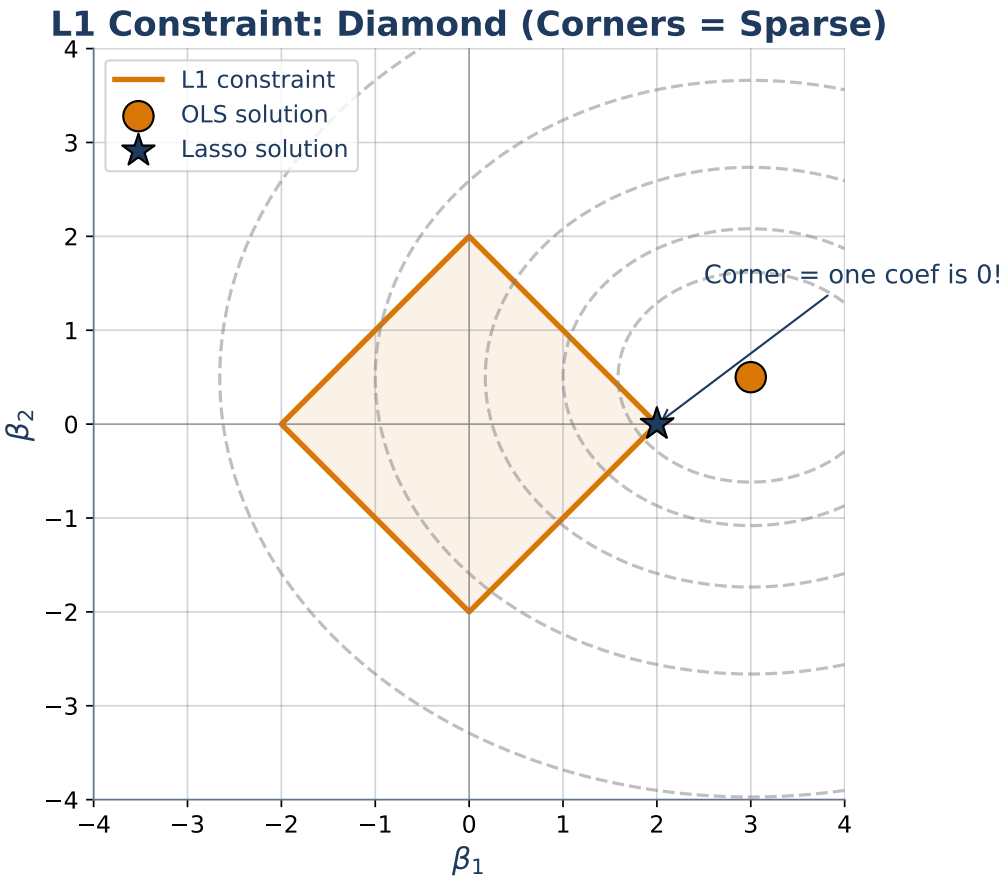


Figure 34: Lasso constraint: coefficients must stay inside a diamond (L1 ball). The corners of the diamond sit on the axes, where one coefficient is exactly zero. Error contours are more likely to touch a corner.

Key Formula: Ridge Regression

Ridge regression minimizes:

$$\text{SSR}_{\text{Ridge}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

where:

- The first term is the usual sum of squared residuals (data fit)
- $\lambda \geq 0$ is the regularization strength (called `alpha` in sklearn)
- $\sum \beta_j^2$ is the L2 penalty (sum of squared coefficients)
- $\lambda = 0$ gives ordinary OLS; $\lambda \to \infty$ shrinks all coefficients to zero

The closed-form solution is:

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

Adding $\lambda\mathbf{I}$ to the diagonal makes the matrix *always* invertible—even when features are perfectly correlated.

Regularization strength (λ): The hyperparameter that controls the balance between fitting the data and keeping coefficients small. A larger λ means stronger shrinkage and a simpler model. The optimal λ is chosen by cross-validation (Section 6).

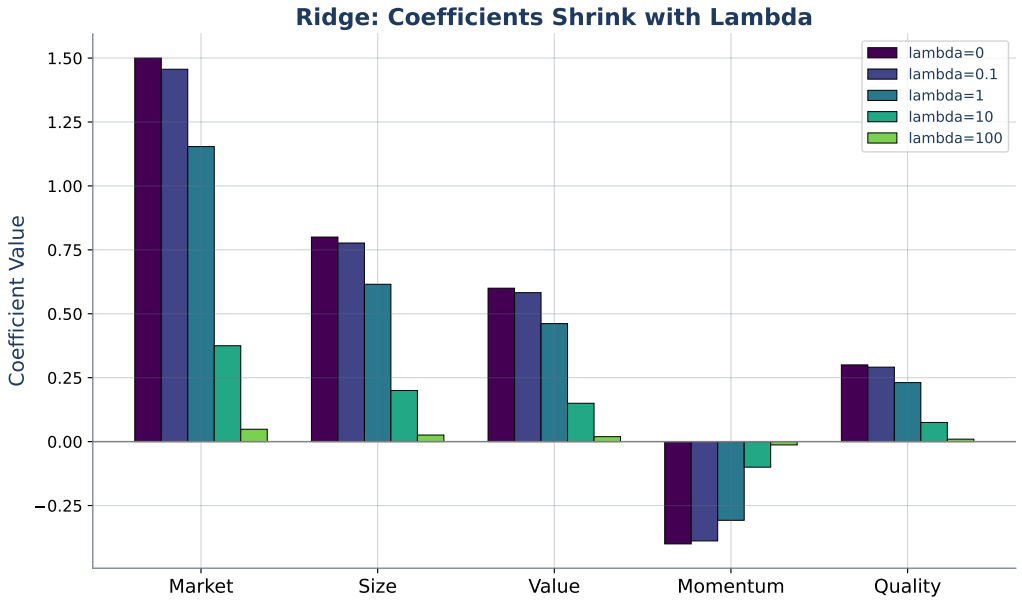


Figure 35: The effect of λ on Ridge coefficients. As λ increases, all coefficients shrink toward zero—but none reach zero exactly.

Lasso Regression (L1 Penalty)

Lasso regression: Least Absolute Shrinkage and Selection Operator. Lasso adds a penalty equal to the sum of *absolute values* of coefficients. The objective is $\min \sum(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$. Lasso can set coefficients to exactly zero, performing automatic feature selection.

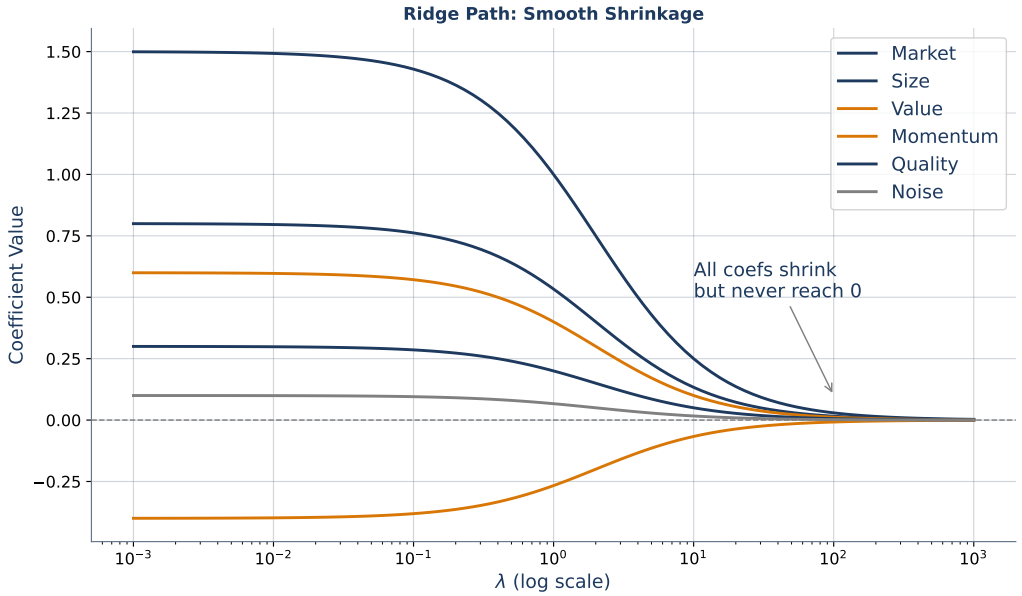


Figure 36: Ridge coefficient path: each line traces one coefficient as λ increases from left to right. All coefficients decay smoothly toward zero.

Key Formula: Lasso Regression

Lasso regression minimizes:

$$\text{SSR}_{\text{Lasso}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

where:

- $\sum |\beta_j|$ is the L1 penalty (sum of absolute coefficient values)
- Unlike Ridge, Lasso has no closed-form solution—it uses coordinate descent
- The L1 penalty produces *sparse* solutions: many $\beta_j = 0$
- The features with non-zero coefficients are the "selected" features

Mnemonic: **L**asso **L**oses features. The L1 penalty forces unimportant coefficients to exactly zero.

ElasticNet: The Compromise

ElasticNet: A regularization method that combines L1 (Lasso) and L2 (Ridge) penalties. The objective is $\min \sum(y_i - \hat{y}_i)^2 + \lambda_1 \sum |\beta_j| + \lambda_2 \sum \beta_j^2$. ElasticNet handles correlated features better than pure Lasso while still producing sparse solutions.

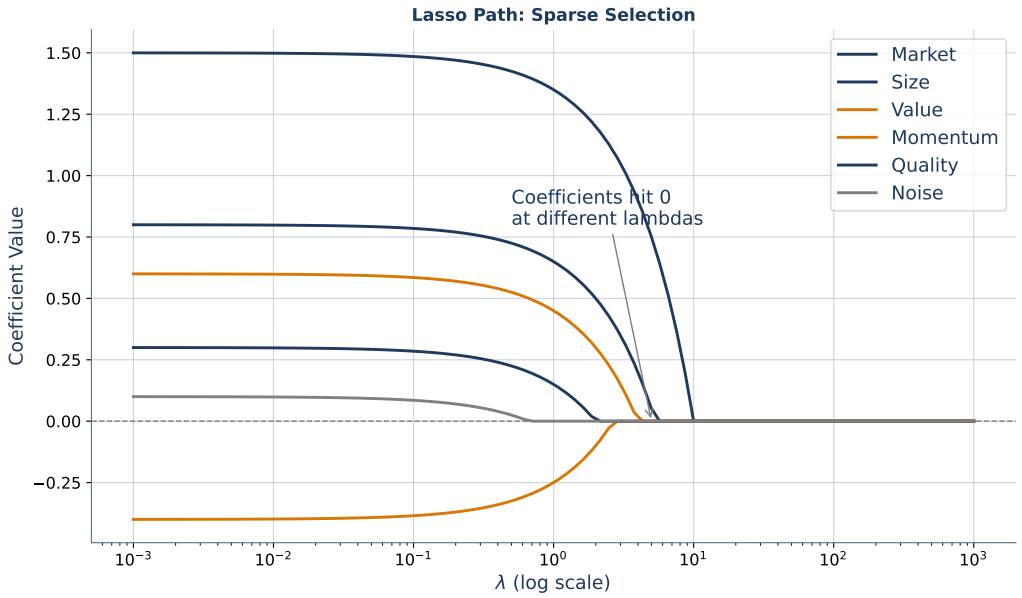


Figure 37: Lasso coefficient path: as λ increases, coefficients hit exactly zero at different points. Once a coefficient reaches zero, it stays there.

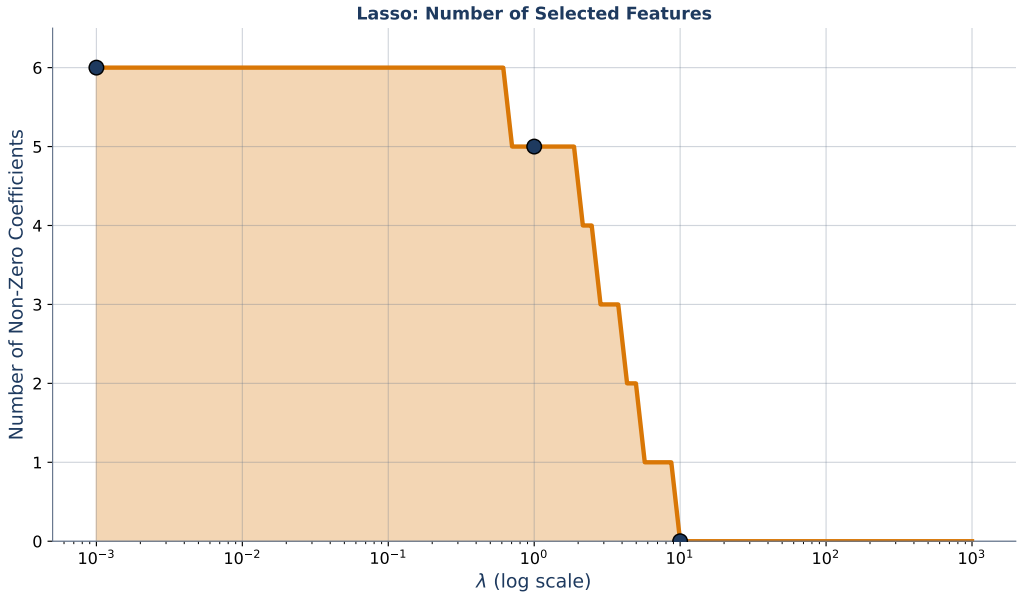


Figure 38: Active features vs. λ. As regularization strength increases, Lasso eliminates features one by one.

Key Formula: ElasticNet

ElasticNet minimizes:

$$\text{SSR}_{\text{ElasticNet}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \left[\alpha \sum_{j=1}^{p} |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 \right]$$

where:

- $\alpha \in [0, 1]$ controls the L1/L2 mix (sklearn's `l1_ratio` parameter)
- $\alpha = 1$: pure Lasso $\alpha = 0$: pure Ridge
- $\alpha = 0.5$: equal mix of L1 and L2 penalties
- ElasticNet selects groups of correlated features together (Lasso would pick only one)

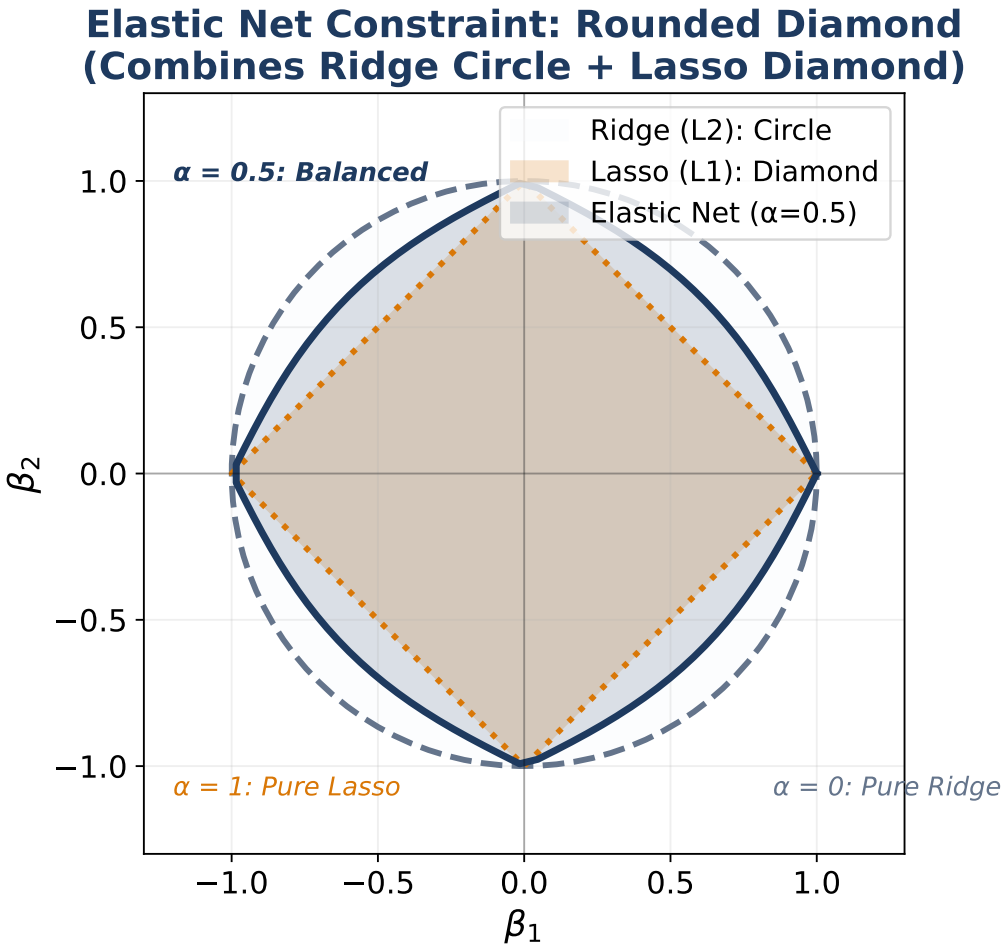


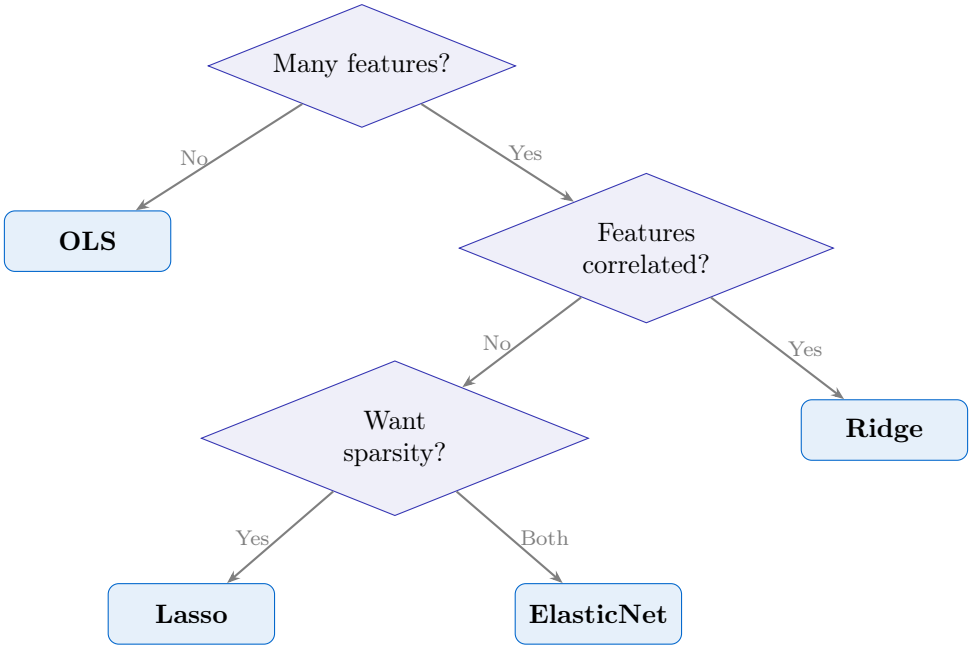
Figure 39: ElasticNet combines the L1 diamond with the L2 circle, creating a rounded diamond. This hybrid constraint region enables both feature selection and coefficient shrinkage.

Definition: Standardization Before Regularization

Regularization penalties depend on the *scale* of the coefficients. If one feature is measured in dollars (range: 10,000–100,000) and another in percentages (range: 0–1), their coefficients will have very different magnitudes—not because of importance but because of units. You must standardize all features to mean 0 and standard deviation 1 *before* applying Ridge or Lasso. In sklearn, use `StandardScaler()` or set `normalize=True`.

Common Misconceptions about Regularization

(1) "Ridge and Lasso do the same thing." They do not. Ridge shrinks all coefficients toward zero but keeps every feature in the model. Lasso sets some coefficients to exactly zero, performing feature selection. The geometric reason is the shape of the constraint: circle (Ridge) vs. diamond (Lasso).
(2) "Larger λ means better regularization." Not necessarily. If λ is too large, the model becomes too simple and underfits. The optimal λ balances data fit against complexity, and cross-validation finds it.
(3) "Standardization is optional before regularization." It is mandatory. The penalty $\lambda \sum \beta_j^2$ treats all coefficients equally—but if features have different scales, the coefficients have different natural magnitudes. Without standardization, the penalty unfairly penalizes features measured in large units.



The decision flowchart above summarizes the choice: few features with no multicollinearity → OLS. Many correlated features → Ridge. Many features, want sparsity → Lasso. Both correlated and want sparsity → ElasticNet.

Worked Examples

Worked Example 1: OLS vs. Lasso Side by Side

A model predicts stock returns using 10 features. After fitting both OLS and Lasso ($\lambda = 0.1$), the coefficients are:

Feature	OLS β	Lasso β
Momentum	0.42	0.38
Value	0.31	0.25
Size	-0.15	-0.10
Volatility	0.08	0.00
Earnings yield	0.05	0.00
Turnover	-0.03	0.00
Analyst score	0.02	0.00
Sector dummy 1	-0.01	0.00
Sector dummy 2	0.004	0.00
Sector dummy 3	-0.002	0.00

OLS gives all 10 features non-zero coefficients, including tiny ones that are likely noise. Lasso zeros out 7 features and keeps only momentum, value, and size. These three are the features with the strongest economic rationale and the largest OLS coefficients. Lasso did what a human expert would do—but automatically and mathematically.

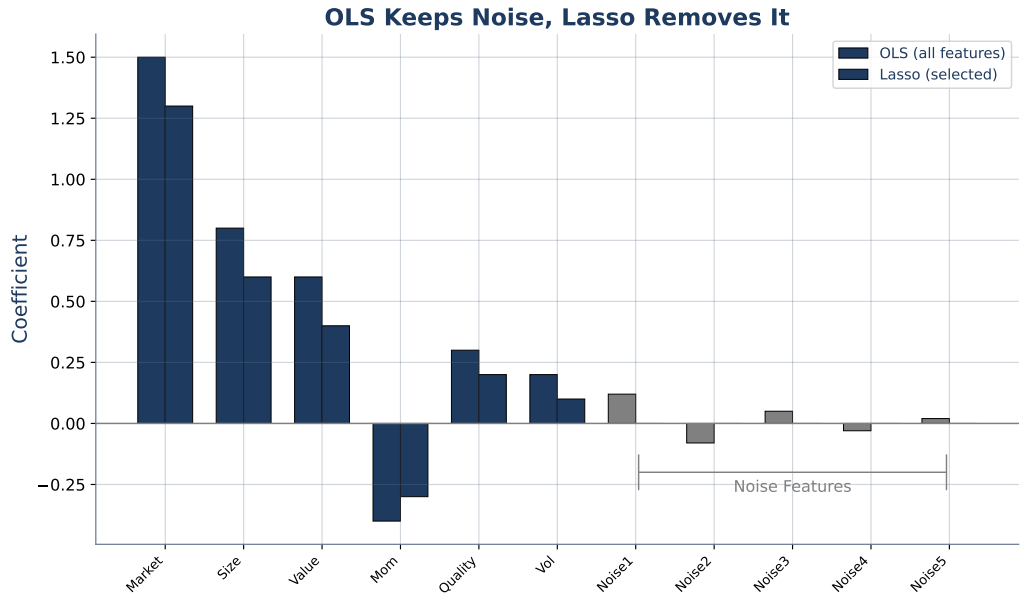


Figure 40: OLS vs. Lasso coefficients. Lasso zeros out weak features while preserving the dominant ones.

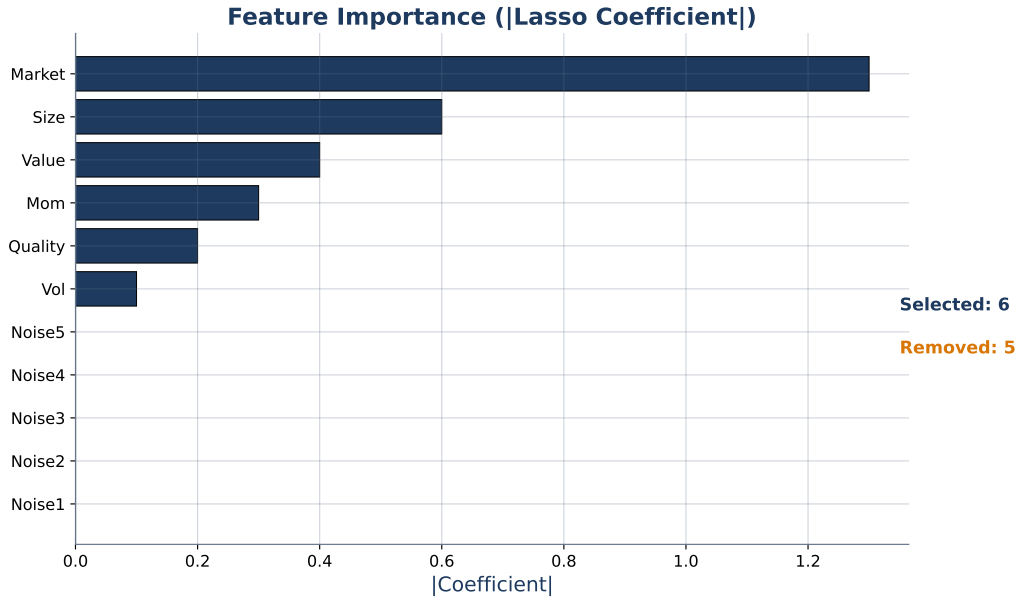


Figure 41: Feature importance from Lasso: non-zero coefficients indicate selected features, ordered by magnitude.

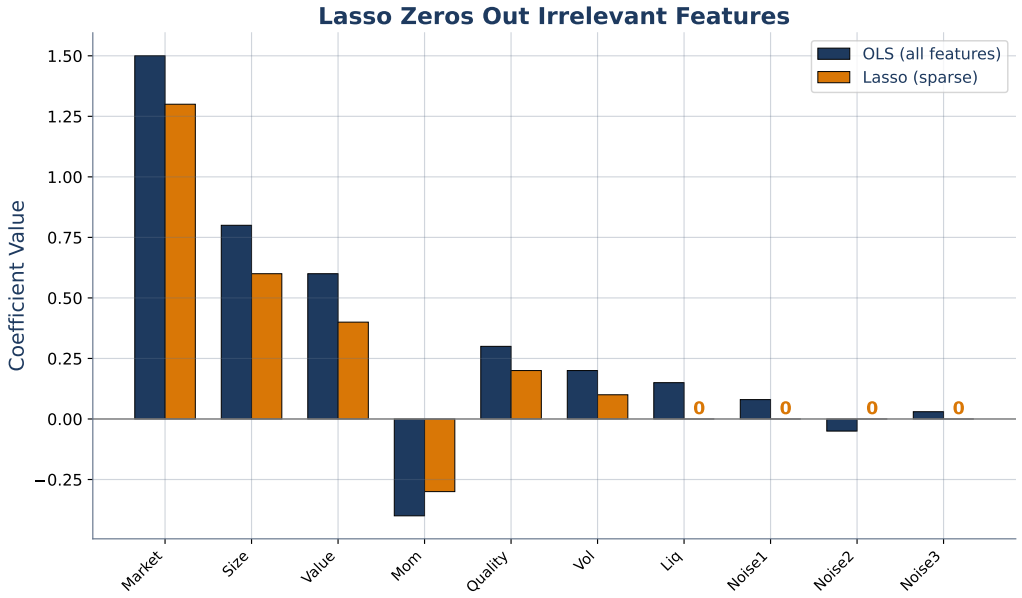


Figure 42: Lasso's automatic feature selection: as λ increases, features are eliminated one by one.

Worked Example 2: Ridge Stabilizes Correlated Features

Suppose two features—book-to-market ratio and earnings yield—are highly correlated ($r = 0.85$). OLS struggles because it cannot tell which feature deserves the credit. The coefficients become unstable: on one sample, book-to-market gets $\beta = 0.8$ and earnings yield gets $\beta = -0.2$; on another sample, the values flip.

Ridge resolves this by shrinking both coefficients. Instead of one large and one negative (which cancel out), Ridge gives both moderate positive values: $\beta_{\text{book}} = 0.35$, $\beta_{\text{earnings}} = 0.30$. The predictions are more stable across different training samples, and the coefficients have the correct sign.

This is exactly what the portfolio manager needed: a model that does not swing wildly when she retrains it next month.

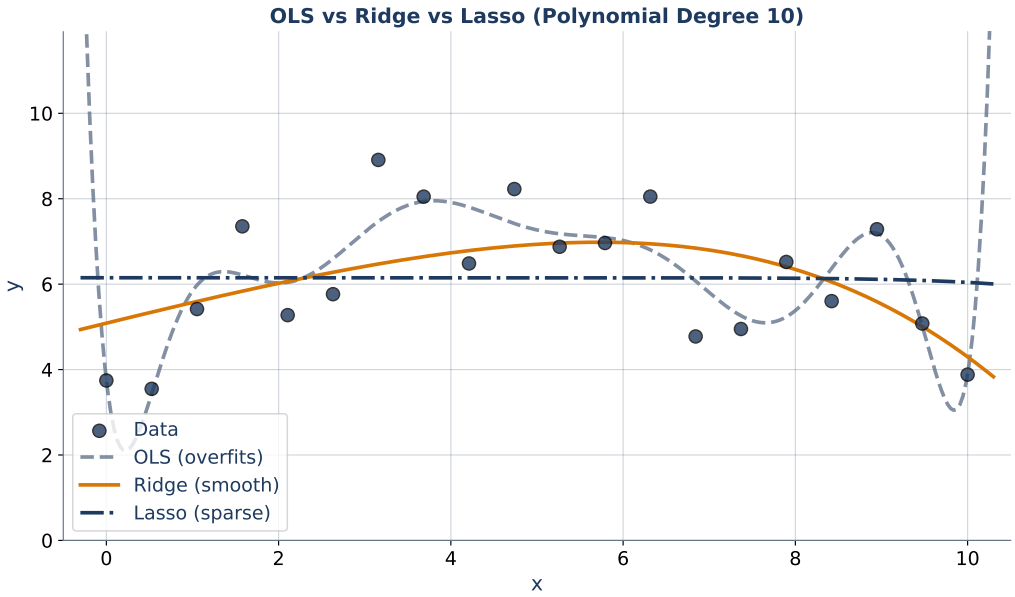


Figure 43: Visual showdown: OLS (wiggly), Ridge (smooth), and Lasso (sparse) fitted to the same polynomial data. Regularization tames overfitting.

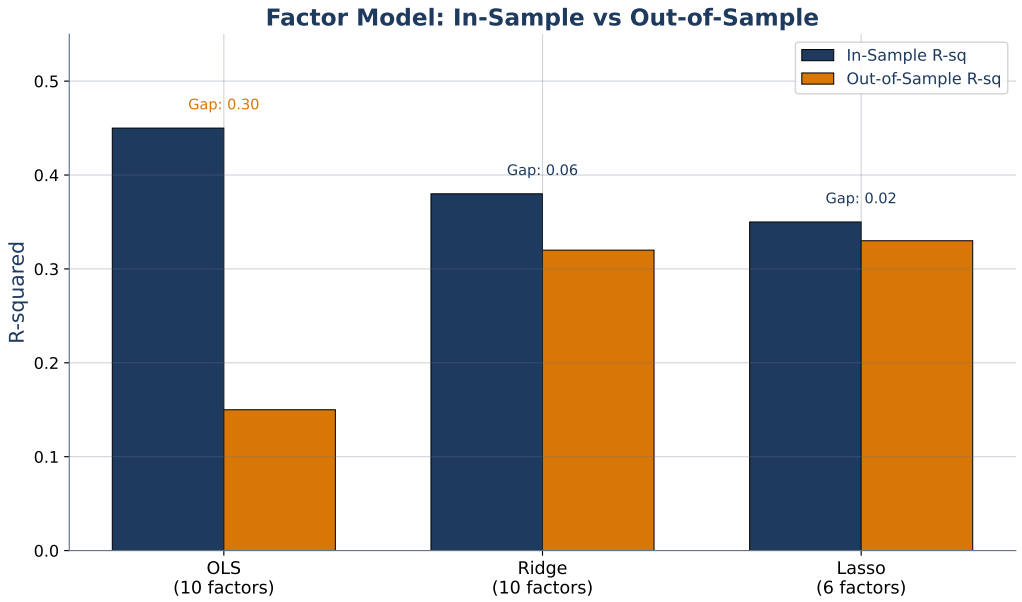


Figure 44: Finance application: a regularized factor model outperforms OLS on out-of-sample data because it avoids overfitting to noise in the training period.

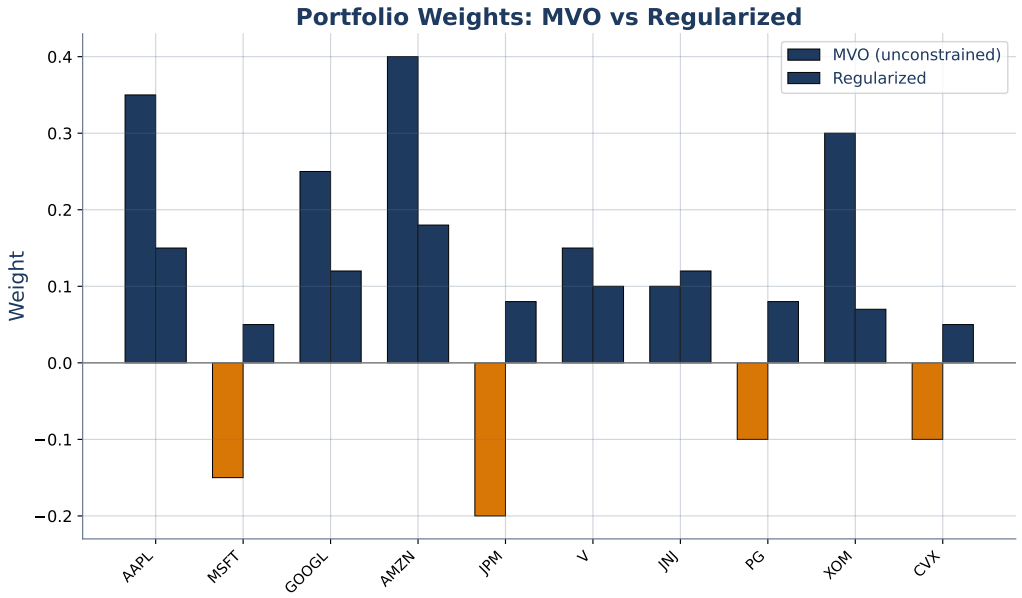


Figure 45: Portfolio weights under OLS vs. regularization. OLS produces extreme allocations; regularization keeps weights reasonable.

Historical Background: Hoerl and Kennard Invent Ridge Regression (1970)

Arthur Hoerl and Robert Kennard were not statisticians by training—they were chemical engineers at the University of Delaware. In the 1960s they were modeling chemical processes with multiple correlated variables, and ordinary least squares kept giving them absurd coefficient estimates. Small changes in the data produced wildly different coefficients, sometimes with the wrong sign.

They diagnosed the problem as multicollinearity: when predictors are correlated, the matrix $\mathbf{X}^\top\mathbf{X}$ becomes nearly singular, and its inverse amplifies noise. Their solution was elegant: add a small positive constant λ to the diagonal of $\mathbf{X}^\top\mathbf{X}$ before inverting. This "ridge" on the diagonal stabilized the inversion.

They published the method in 1970 in *Technometrics* under the title "Ridge Regression: Biased Estimation for Nonorthogonal Problems." The title is revealing: Ridge intentionally introduces bias (the coefficients are shrunken toward zero) in exchange for a much larger reduction in variance. The net effect is lower total error.

Hoerl and Kennard were chemists who noticed ordinary regression gave absurd answers when variables were correlated—they added a "ridge" to stabilize it. Their trick is now used billions of times daily in machine learning systems worldwide.

Problem 4.1 (Easy)

Given two coefficient vectors from the same 5-feature model, identify which is Ridge and which is Lasso:

Model X: $\boldsymbol{\beta} = (0.32, 0.18, 0.05, 0.03, 0.01)$

Model Y: $\boldsymbol{\beta} = (0.40, 0.22, 0, 0, 0)$

Explain your reasoning.

Solution: see Appendix.

Problem 4.2 (Easy)

Explain in your own words why Lasso performs feature selection but Ridge does not. Your answer should reference the geometric shapes (circle vs. diamond) and explain why one produces exact zeros and the other does not.

Solution: see Appendix.

Problem 4.3 (Medium)

You are shown a Lasso coefficient path plot where 10 lines represent 10 features. At $\lambda = 0.01$, all 10 are non-zero. At $\lambda = 0.1$, five are zero. At $\lambda = 1.0$, only one remains non-zero.

(a) Which feature is the strongest predictor?

(b) If cross-validation selects $\lambda = 0.05$ as optimal, approximately how many features will the model retain?

(c) What happens if you set $\lambda = 10$?

Solution: see Appendix.

Problem 4.4 (Medium)

Consider a two-variable regression problem with features x_1 and x_2, where $\mathbf{X}^\top\mathbf{X} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ and $\mathbf{X}^\top\mathbf{y} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$.

(a) Compute the OLS solution $\hat{\boldsymbol{\beta}}_{\text{OLS}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$.

(b) Compute the Ridge solution for $\lambda = 1$: $\hat{\boldsymbol{\beta}}_{\text{Ridge}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$.

(c) By how much did each coefficient shrink?

Solution: see Appendix.

Problem 4.5 (Hard)

Derive the Ridge regression estimator. Start from the Ridge objective:

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

Take the gradient with respect to $\boldsymbol{\beta}$, set it to zero, and solve. Show that $\hat{\boldsymbol{\beta}}_{\text{Ridge}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$.
Solution: see Appendix.

Connecting Backward and Forward

We have built a complete toolkit for fitting regression models:

Section	Tool	What It Does	When to Use
1	OLS	Fits the best line	Few features, no multicollinearity
2	LINE checks	Validates assumptions	Always, before trusting results
3	Bias-variance	Diagnoses overfitting	When test error $\gg$ training error
4	Ridge/Lasso	Controls complexity	Many features or correlated features

Two questions remain. First: *how do we measure* whether a model is good? We have seen R^2 and RMSE in passing, but we have not defined them precisely or discussed their limitations. That is Section 5: regression metrics.

Second: *how do we choose* the best value of λ? We cannot use training error—that always favors $\lambda = 0$ (pure OLS). We need a method that estimates how the model will perform on data it has never seen. That method is cross-validation, and it is Section 6.

Key Takeaway: Regularization is a controlled sacrifice of training accuracy for real-world reliability—Ridge keeps all features quiet, Lasso eliminates the irrelevant ones.

5 Measuring What Matters – MSE, RMSE, MAE, and R^2

Opening Problem: The 95% Accurate Model That Lost Money

A junior analyst at a quantitative trading desk builds a model to predict daily stock returns. She runs it through scikit-learn and proudly reports: "The model explains 95% of in-sample variance. $R^2 = 0.95$."

Her manager authorizes a paper-trading trial. After three weeks, the portfolio is down 4%. The model predicted the direction of returns correctly most of the time, but when it was wrong, it was *spectacularly* wrong. A single day when the model predicted +1.5% but the stock dropped −8% wiped out weeks of small gains.

The analyst used R^2 as her only metric. She never looked at RMSE. She never checked whether her errors were symmetric or dominated by outliers. She never asked: "In the units my portfolio cares about—dollars—how wrong is this model on a typical day?"

This section gives you the full toolkit for answering that question. No single metric tells the whole story. You need several, and you need to know which one matters most in your specific context.

Discovery Question

Two models predict house prices. Model A has RMSE of \$50,000. Model B has MAE of \$30,000. Which model would you trust more for pricing a \$2 million penthouse? What about a \$200,000 apartment?

Errors as Squares and Strips

In Section 1 we saw that OLS minimizes squared residuals. But minimizing squared errors and *reporting* squared errors are different things. MSE—mean squared error—is the average of those squared residuals. It has a problem: its units are squared. If you predict stock returns in percent, MSE is in "percent squared," which is not a number any portfolio manager can interpret at a glance.

Figure 46 makes this concrete. Each residual becomes a literal square whose area represents its contribution to MSE. Large residuals produce disproportionately large squares. A single outlier can dominate the entire metric.

Now consider the alternative. Instead of squaring each error, take its absolute value. A residual of −3 becomes 3; a residual of +5 becomes 5. No squaring, no disproportionate punishment. Figure 47 compares the two philosophies. The squared penalty rises steeply for large errors; the absolute penalty rises linearly.

Think of it as two restaurant critics. Critic A squares their displeasure: a slightly cold soup gets a penalty of 1, but a raw steak gets a penalty of 100. Critic B uses absolute displeasure: cold soup gets 1, raw steak gets 10. Critic A's overall rating is dominated by the worst dish. Critic B gives every dish a fair hearing. Neither approach is wrong—they answer different questions.

The Metrics, Precisely Defined

Mean Squared Error (MSE): The average of the squared residuals: $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. MSE is always non-negative, equals zero only for perfect predictions, and is sensitive to outliers because large errors are squared.

Root Mean Squared Error (RMSE): The square root of MSE: $\text{RMSE} = \sqrt{\text{MSE}}$. RMSE has the same units as the target variable, making it directly interpretable. If stock returns are in percent, RMSE is in percent.

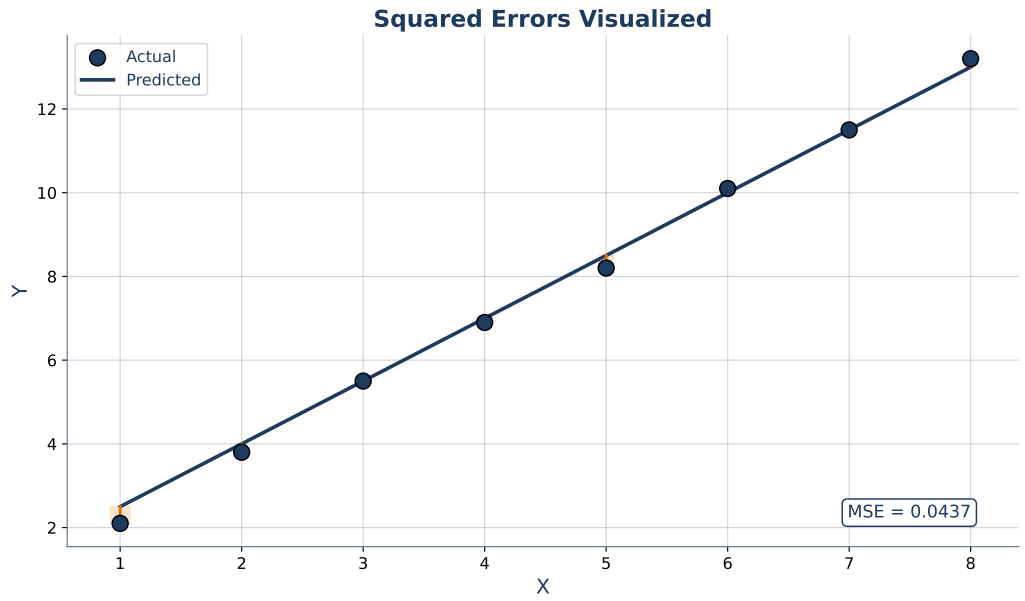


Figure 46: Squared errors as literal squares. Each residual's contribution to MSE is the area of its square. One large error can dominate the total.

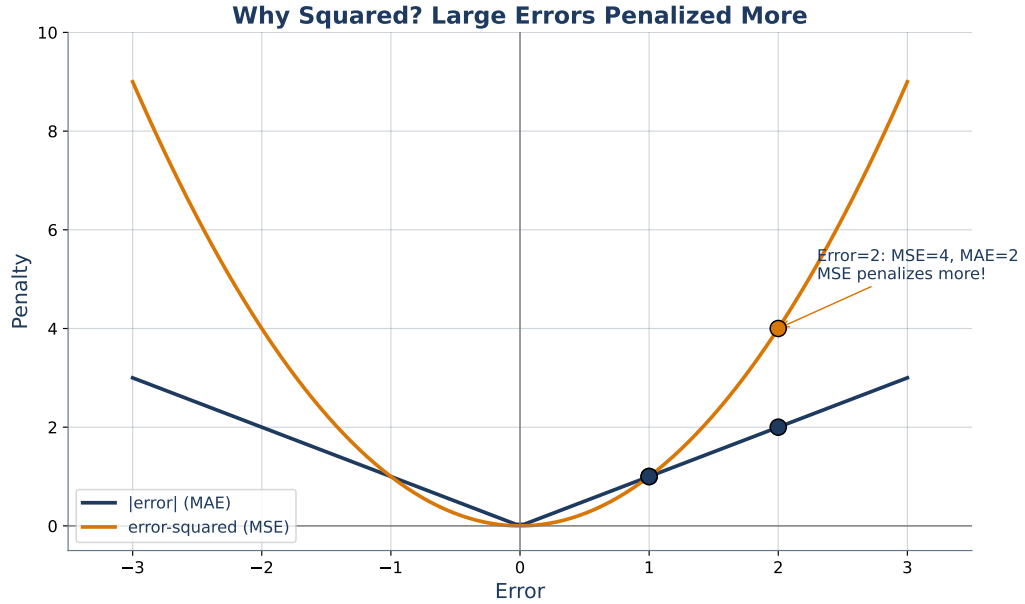


Figure 47: MSE vs. MAE penalty functions. Squaring penalizes large errors quadratically; absolute value penalizes all errors equally.

Key Formula: RMSE

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where:

- y_i is the observed value (actual return on day i)
- $\hat{y}_i$ is the predicted value (model's forecast for day i)
- n is the number of predictions

Plain English: "On average, how far off are my predictions, in the original units, with extra weight on large errors?"

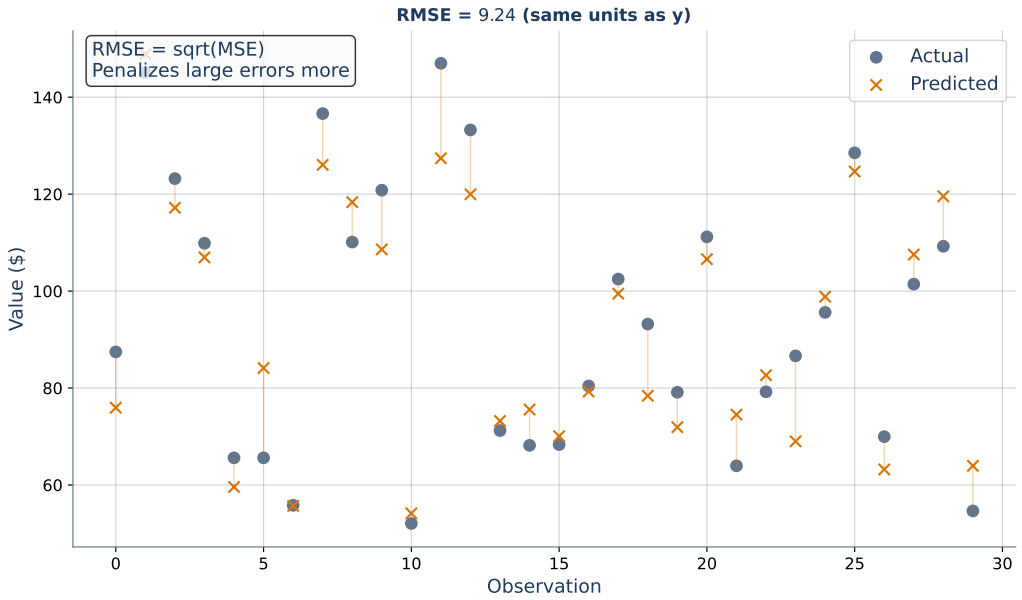


Figure 48: RMSE concept: take the residuals, square them, average, and take the square root. The result is an error measure in the same units as y.

Mean Absolute Error (MAE): The average of the absolute residuals: $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$. MAE treats all errors equally regardless of size. It is more robust to outliers than RMSE.

Key Formula: MAE

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Plain English: "On average, how far off are my predictions, treating all errors equally?"
RMSE vs. MAE rule of thumb: If RMSE is much larger than MAE (say RMSE/MAE > 1.2), your errors are uneven—a few large ones are inflating RMSE. Investigate those predictions before choosing a metric.

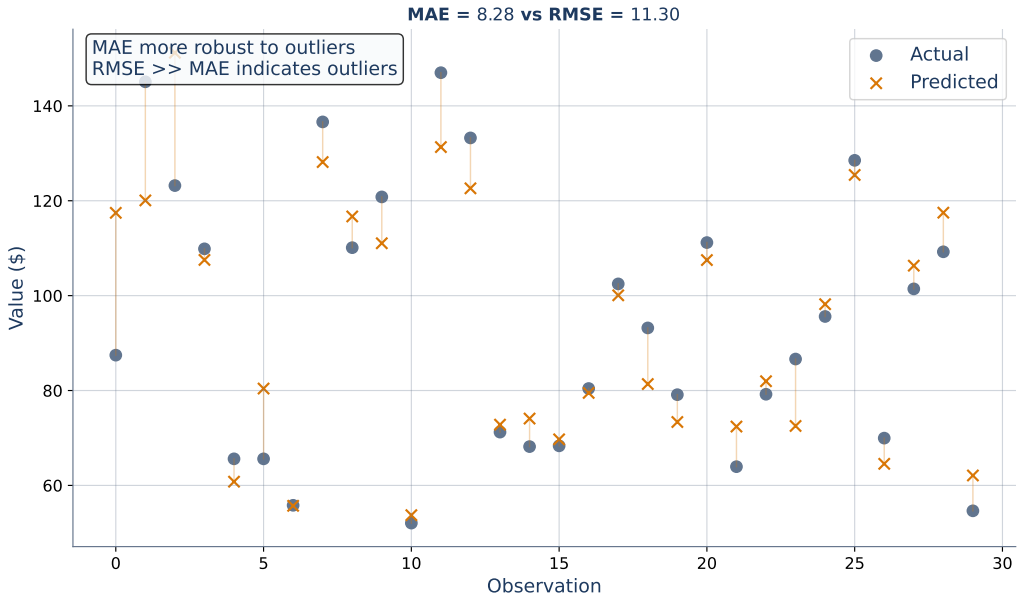


Figure 49: MAE concept: take the absolute residuals and average them. No squaring, no disproportionate punishment.

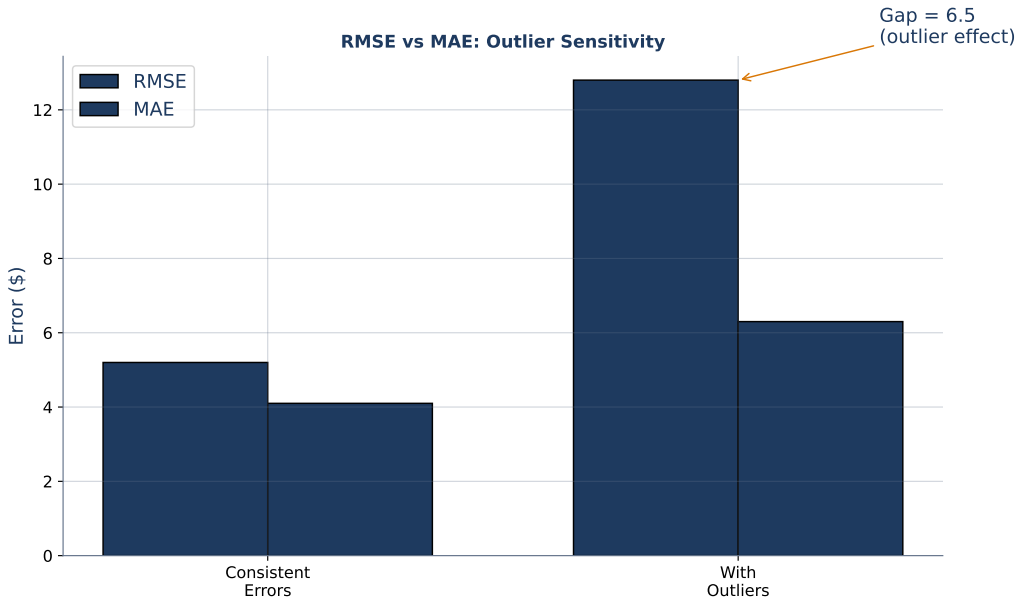


Figure 50: RMSE vs. MAE side by side. When all errors have similar magnitude, RMSE $\approx$ MAE. When outliers are present, RMSE $>$ MAE.

R^2: The Headline Number

RMSE and MAE tell you the *size* of the error. They do not tell you whether that error is large or small relative to the variability of the data. An RMSE of \$50,000 is terrible for predicting apartment prices and acceptable for predicting Manhattan penthouse prices.

R^2 solves this by comparing the model's errors to the errors of a naïve baseline: predicting the mean for every observation.

R^2 (coefficient of determination): The fraction of the target's variance explained by the model: $R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$. An R^2 of 0 means the model does no better than predicting the mean. An R^2 of 1 means perfect predictions. R^2 can be negative if the model is worse than the mean.

Key Formula: R^2

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where:

- $SS_{\text{res}} = \sum(y_i - \hat{y}_i)^2$ is the residual sum of squares (error left over)
- $SS_{\text{tot}} = \sum(y_i - \bar{y})^2$ is the total sum of squares (total variance in y)
- $\bar{y}$ is the mean of the observed values

Three sums of squares:

$$SS_{\text{tot}} = SS_{\text{reg}} + SS_{\text{res}}$$

$SS_{\text{reg}} = \sum(\hat{y}_i - \bar{y})^2$ is the variance the model *explains*.

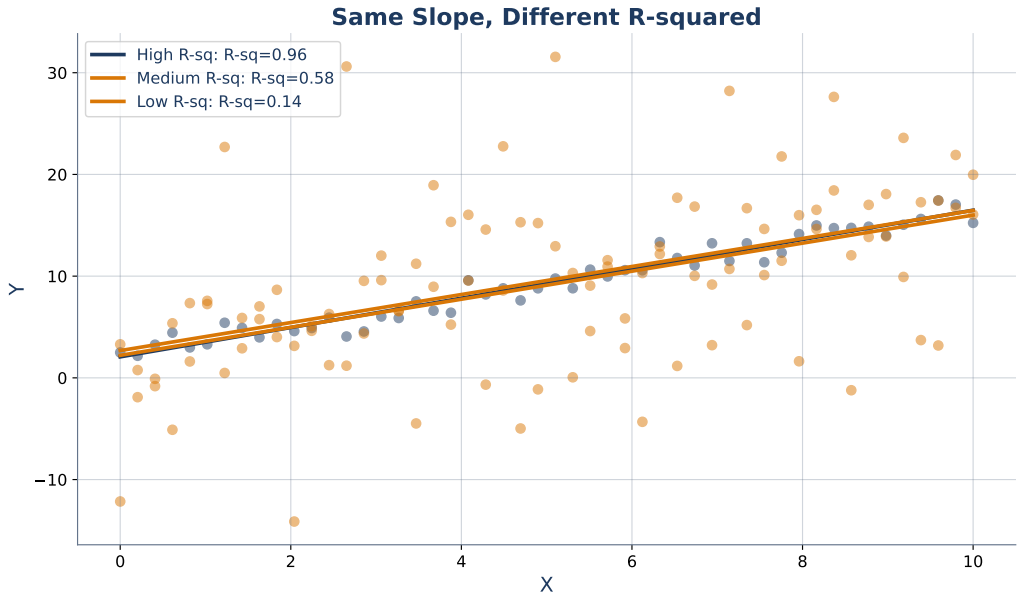


Figure 51: Three models with different R^2 values: 0.3, 0.7, and 0.95. Higher R^2 means tighter scatter around the regression line.

A critical point that trips up beginners: R^2 depends on context. An R^2 of 0.30 is *excellent* in finance (stock returns are noisy; explaining 30% of the variance is a strong signal). The same R^2 would be *embarrassing* in physics, where precise laws govern outcomes. Figure 53 illustrates this domain dependence.

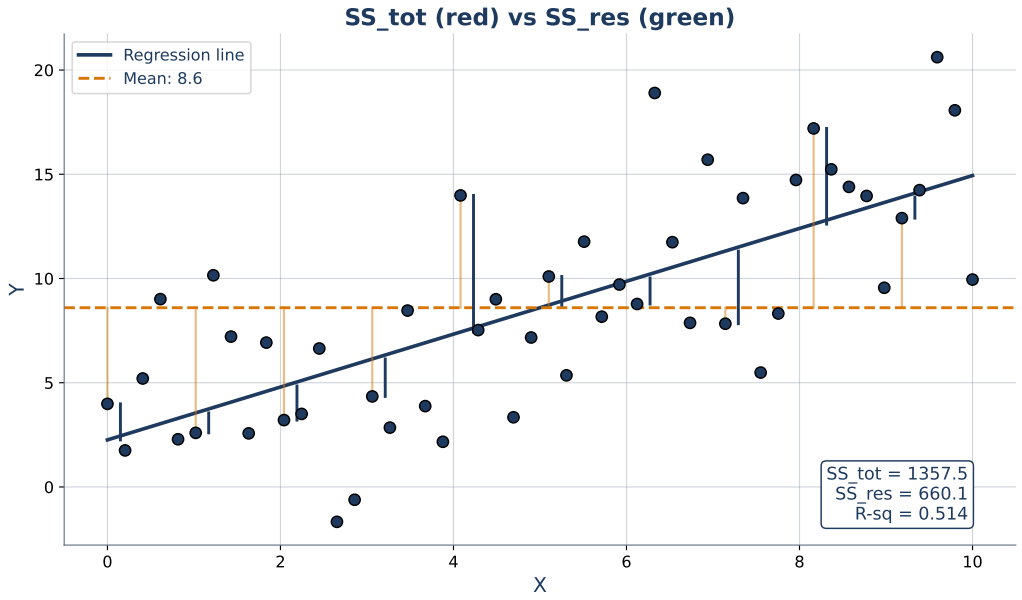


Figure 52: The decomposition of total variance into explained (SS_{reg}) and unexplained (SS_{res}) components. R^2 is the ratio $SS_{\text{reg}}/SS_{\text{tot}}$.

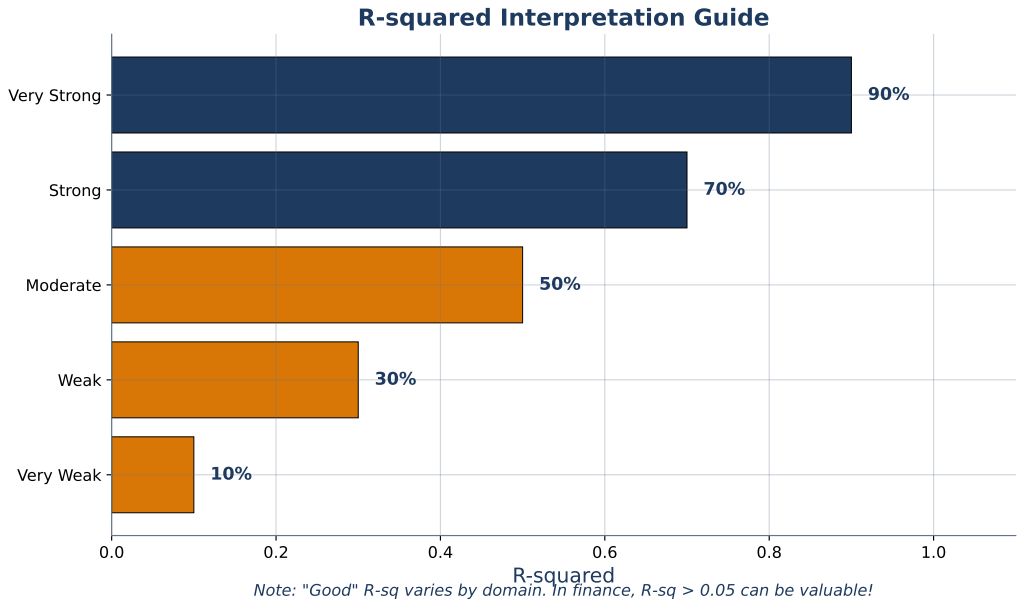


Figure 53: R^2 interpretation varies by domain. In finance, $R^2 = 0.03$ for daily returns is considered a useful signal. In engineering, you might expect $R^2 > 0.99$.

Adjusted R^2: Penalizing Complexity

There is a trap with ordinary R^2: adding *any* feature to a model—even pure random noise—will increase R^2 (or at least not decrease it). The model gets one more degree of freedom to fit the training data, and it uses it. This creates a perverse incentive to add features regardless of their signal.

Adjusted R^2 corrects for this by penalizing the number of features.

Adjusted R^2: A modified version of R^2 that penalizes model complexity:

$$\text{Adj. } R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

where n is the sample size and p is the number of features. Adjusted R^2 can *decrease* when you add a useless feature, because the penalty for adding a parameter outweighs the trivial R^2 improvement.

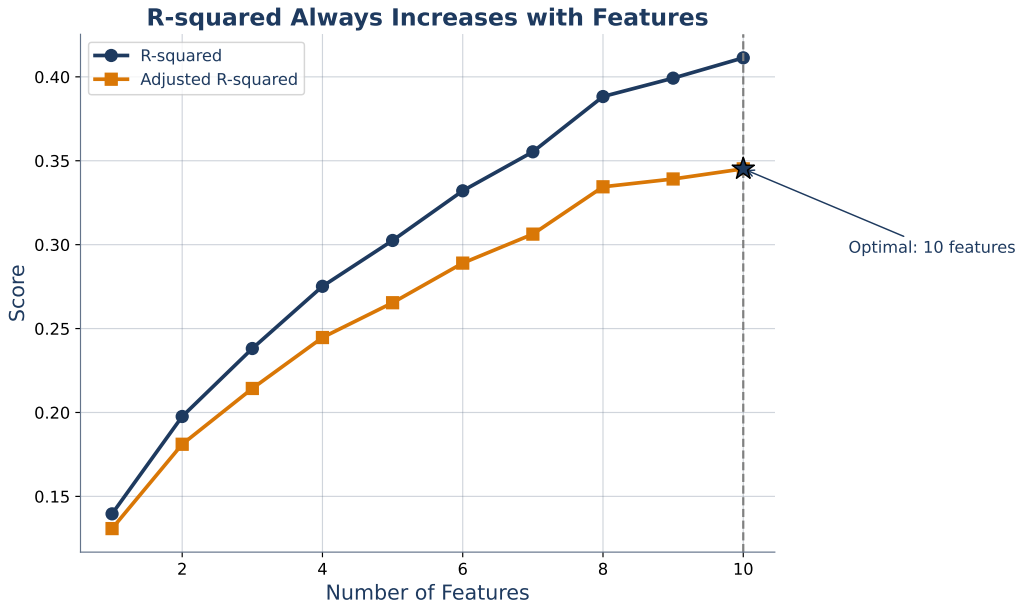


Figure 54: R^2 vs. adjusted R^2 as features are added. R^2 always increases. Adjusted R^2 peaks at the optimal feature count and then declines.

Common Misconceptions about Metrics

(1) "RMSE and MAE are interchangeable." They are not. RMSE penalizes large errors quadratically because of the squaring step. MAE treats a \$100 error and a \$10 error as contributing 10× and 1× respectively. RMSE treats them as 10,000× and 100×. Use RMSE when large errors are catastrophic (portfolio blowup); use MAE when errors are roughly equally costly.

(2) "R^2 of 0.3 means the model is bad." In finance, an R^2 of 0.03 on daily stock returns is publishable. Returns are dominated by noise; even a small fraction of explained variance can generate profitable trading signals. Always interpret R^2 in domain context.

(3) "Adjusted R^2 can increase when you add a feature." Correct—but only if that feature adds more signal than the penalty for complexity. If adjusted R^2 increases, keep the feature. If it decreases, the feature is adding noise, not signal.

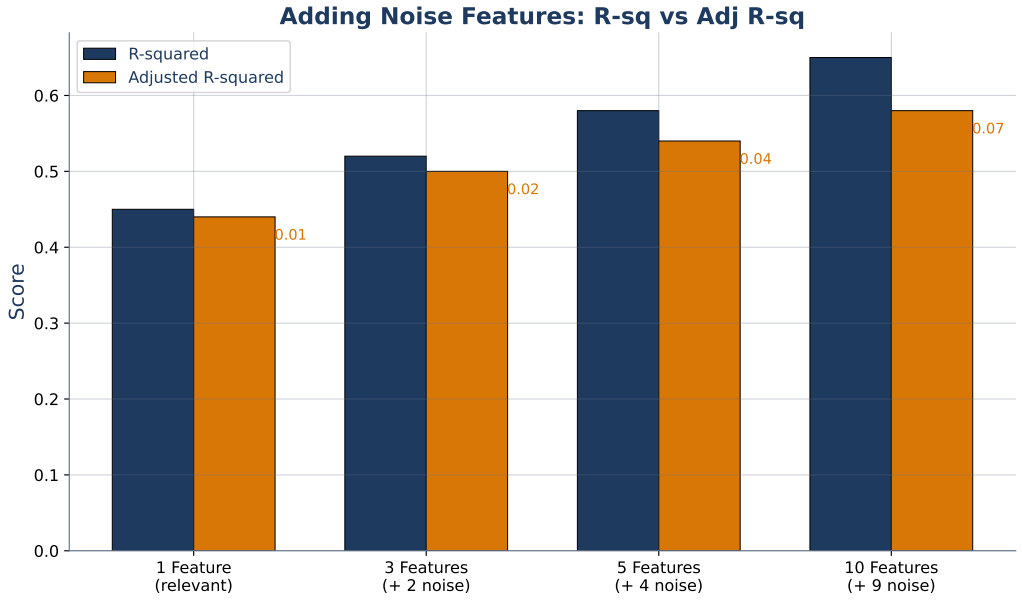
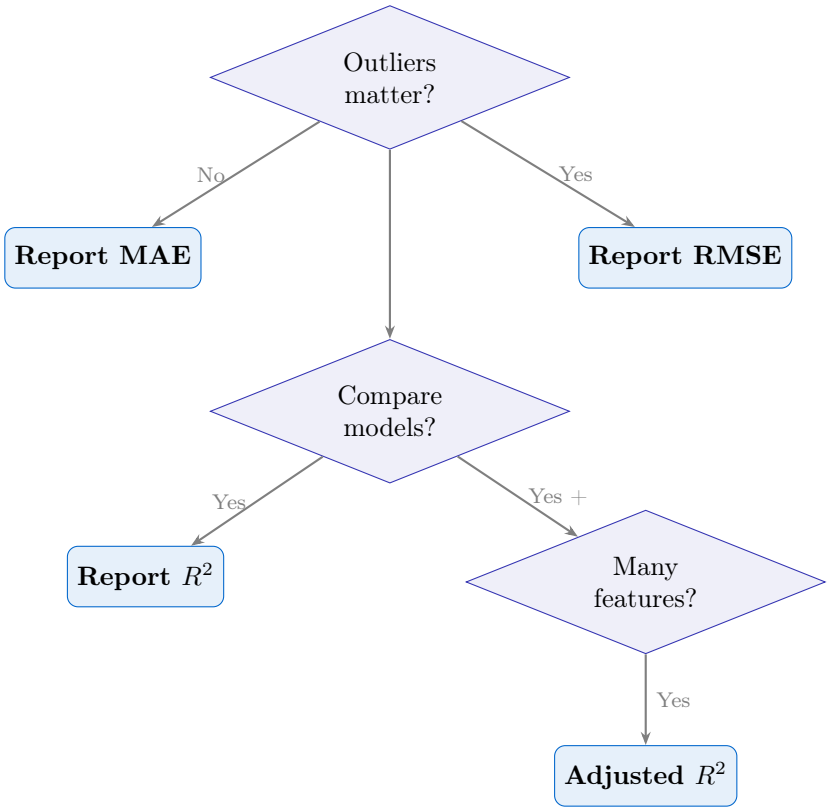


Figure 55: Adding noise features: R^2 rises monotonically, but adjusted R^2 falls after the true features are included. The gap between the two curves reveals the noise.



The flowchart above is a starting point, not a rigid rule. In practice, report both RMSE and R^2 together. Use MAE as a robustness check when you suspect outliers. Use adjusted R^2 whenever you compare models with different numbers of features.

Worked Examples

Worked Example 1: Computing All Four Metrics

A model predicts five daily stock returns (in percent):

Day	Actual y_i	Predicted $\hat{y}_i$	Residual e_i	e_i^2
1	1.2	1.0	0.2	0.04
2	-0.5	-0.3	-0.2	0.04
3	0.8	0.7	0.1	0.01
4	-2.1	-0.5	-1.6	2.56
5	0.3	0.4	-0.1	0.01

MSE $= \frac{0.04+0.04+0.01+2.56+0.01}{5} = \frac{2.66}{5} = 0.532$
RMSE $= \sqrt{0.532} = 0.730\%$
MAE $= \frac{0.2+0.2+0.1+1.6+0.1}{5} = \frac{2.2}{5} = 0.440\%$
RMSE/MAE ratio $= 0.730/0.440 = 1.66$. This is well above 1.2, signaling that one large error (Day 4: -1.6%) dominates the RMSE. Investigate Day 4 before choosing your metric.
R^2**:** We need SS_{tot}. The mean of y is $\bar{y} = (1.2 - 0.5 + 0.8 - 2.1 + 0.3)/5 = -0.06$. Then $SS_{\text{tot}} = (1.26)^2 + (-0.44)^2 + (0.86)^2 + (-2.04)^2 + (0.36)^2 = 1.588 + 0.194 + 0.740 + 4.162 + 0.130 = 6.813$. So $R^2 = 1 - 2.66/6.813 = 0.610$. The model explains 61% of the variance in daily returns—strong for a financial model.

Worked Example 2: Finance-Specific Metrics

In quantitative finance, portfolio managers often use the *information coefficient* (IC) instead of R^2. The IC is simply the Pearson correlation between predicted and actual returns:

$$\text{IC} = \text{corr}(\hat{r}, r)$$

IC benchmarks differ radically from academic R^2 benchmarks:

IC Value	Interpretation
< 0.02	Noise—no signal
0.03–0.05	Decent alpha signal
> 0.05	Strong signal
> 0.10	Suspiciously good—check for data leakage

An IC of 0.05 corresponds to an R^2 of only 0.0025 (since $R^2 \approx \text{IC}^2$ for centered data). That 0.25% of explained variance sounds negligible, but applied to billions of dollars across thousands of trades, it generates real profit. The lesson: what counts as "good" depends entirely on the domain and the economic question.

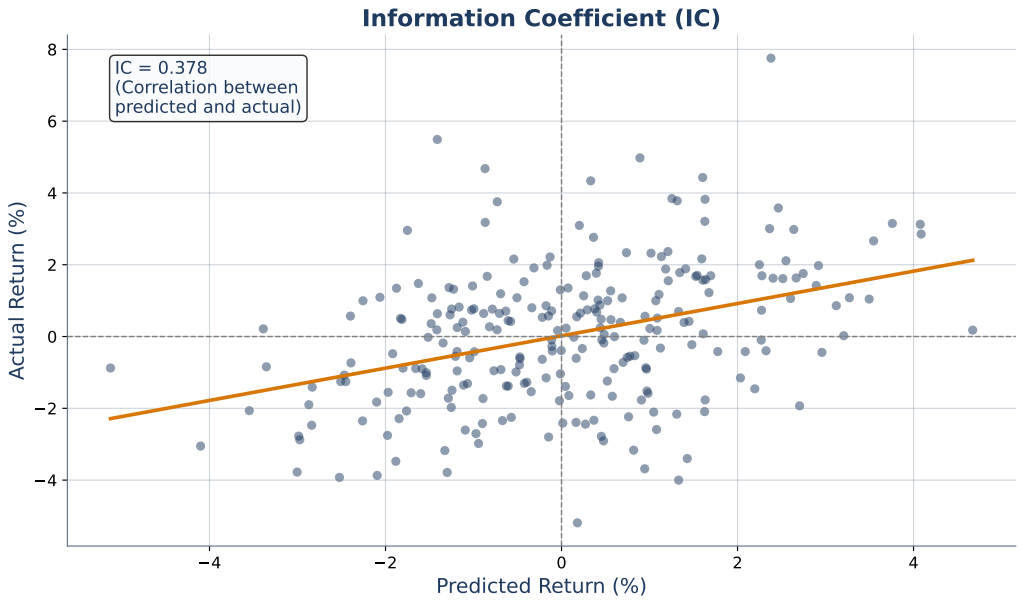


Figure 56: The information coefficient: correlation between predicted and actual returns. Finance uses IC because it focuses on ranking accuracy rather than absolute fit.

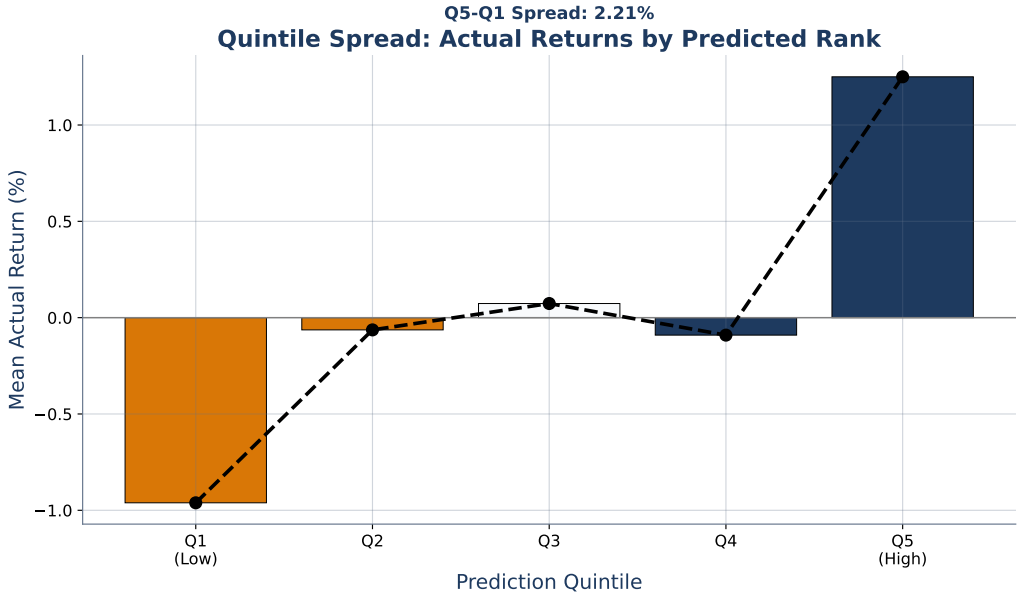


Figure 57: Quintile spread returns: sort stocks by predicted return, form five portfolios, and compare the top quintile to the bottom. A wide spread indicates a model with real predictive power.

Historical Background: Gauss and the Birth of BLUE (1821)

We met Carl Friedrich Gauss in Section 1 as the inventor of least squares. But Gauss made a second, equally profound contribution to regression theory. In 1821 he proved what is now called the Gauss-Markov theorem: among all linear estimators that are unbiased, OLS has the smallest variance. In modern shorthand, OLS is BLUE—the Best Linear Unbiased Estimator.

"Best" means minimum variance. "Linear" means the estimator is a linear function of the observations. "Unbiased" means that on average, across many datasets drawn from the same process, the estimator hits the true parameter. No other estimator that is both linear and unbiased can beat OLS in terms of precision.

The catch—and this connects directly to metrics—is that "best" refers to the *coefficients*, not to prediction accuracy on new data. OLS gives you the most precise coefficient estimates, but if the model is overfit, those precise estimates describe noise rather than signal. That is why metrics like RMSE and R^2 must be evaluated on *test* data, not training data. The Gauss-Markov theorem guarantees the best in-sample fit, but says nothing about generalization.

Problem 5.1 (Easy)

A model produces five residuals: $+2, -1, +3, -2, +1$. Compute: (a) MSE, (b) RMSE, (c) MAE. Which metric is larger, RMSE or MAE? Why?

Solution: see Appendix.

Problem 5.2 (Easy)

A model has $SS_{\text{res}} = 200$ and $SS_{\text{tot}} = 800$. Compute R^2. Interpret the result in one sentence. What fraction of the variance is unexplained?

Solution: see Appendix.

Problem 5.3 (Medium)

You build two models to predict monthly stock returns. Model A uses 3 features and has $R^2 = 0.25$. Model B uses 15 features and has $R^2 = 0.30$. There are $n = 60$ observations.

(a) Compute the adjusted R^2 for both models.

(b) Which model should you prefer, and why?

(c) Explain in your own words why R^2 always increases with more features but adjusted R^2 does not.

Solution: see Appendix.

Problem 5.4 (Medium)

You are building a model to predict house prices. Your colleague argues for using RMSE; your manager prefers MAE. Construct a scenario (with specific numbers) where the two metrics lead to different model choices. Which metric is more appropriate if the client is a luxury real estate broker? Which is better for an affordable housing agency?

Solution: see Appendix.

Problem 5.5 (Hard)

Show algebraically that $R^2 = 1 - \frac{\text{MSE}}{\text{Var}(y)}$, where $\text{Var}(y) = \frac{1}{n}\sum(y_i - \bar{y})^2$ is the variance of the target. Start from the definition $R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$ and simplify.
Solution: see Appendix.

Connecting Backward and Forward

In Section 4 we introduced regularization and noticed that Ridge and Lasso require choosing a hyperparameter λ. We said cross-validation would handle that choice. Now we have the metrics to evaluate any model—RMSE, MAE, R^2, adjusted R^2. These are the yardsticks that cross-validation uses.

But there is a subtle question we have not addressed: *which data* do we evaluate on? If we compute RMSE on the training data, we get an optimistic number (the model has already seen those observations). If we hold out a single test set, our estimate depends on which data happened to land in the test set. And for time series data—the bread and butter of finance—random hold-outs are not just imprecise, they are *invalid*. Predicting the past from the future is not a valid test of a model.

How do we get a reliable, unbiased estimate of a model's real-world performance? That is Section 6: cross-validation and model selection.

Key Takeaway: No single metric tells the whole story—always report RMSE and R^2 together, and interpret them in the context of your domain.

6 How Do You Know It Will Work Tomorrow? – Cross-Validation and Model Selection

Opening Problem: Same Data, Opposite Results

An analyst builds a model to predict monthly stock returns using 10 years of data (2010–2019). She holds out 2020 as a test year. The model performs brilliantly: $R^2 = 0.42$ on the test set. She is ready to deploy.

A colleague is skeptical. He suggests a different test: train on odd-numbered years (2011, 2013, 2015, 2017, 2019) and test on even-numbered years (2010, 2012, 2014, 2016, 2018, 2020). The model's R^2 drops to -0.05. It is worse than predicting the mean.

Same data. Same model. Opposite conclusions. What went wrong?

The colleague's test shuffled time. The model trained on 2019 data and used it to "predict" 2010. That is not prediction—it is time travel. Financial data has a direction: the past comes first, the future comes second. Any evaluation that ignores this direction gives misleading results.

This section shows you how to evaluate models honestly—first with K-fold cross-validation for cross-sectional data, and then with walk-forward validation for time series.

Discovery Question

You train a model on 10 years of stock data and test on the 11th year—it works brilliantly. You train on years 1, 3, 5, 7, 9 and test on years 2, 4, 6, 8, 10—it fails miserably. Same data, opposite results. What went wrong?

The Exam Analogy, Revisited

In Section 3 we used the analogy of a student memorizing practice exams. Cross-validation is the solution to that problem: instead of one practice exam, you take five different exams and average your scores. Each exam tests you on material you did not study from directly.

K-fold cross-validation works the same way. Split the data into K equally sized pieces ("folds"). Train on $K - 1$ folds and test on the remaining fold. Rotate which fold is held out, so every data point gets exactly one turn as a test observation. Average the K test scores. The result is a more stable and less biased estimate of real performance than any single train/test split.

Figure 58 shows the structure. Five folds, five rounds. Every observation is tested exactly once.

K-Fold Cross-Validation

K-fold cross-validation: A resampling method that splits data into K non-overlapping folds. The model is trained K times, each time leaving out one fold for testing. The final performance estimate is the average of the K test scores. Common choices are $K = 5$ or $K = 10$.

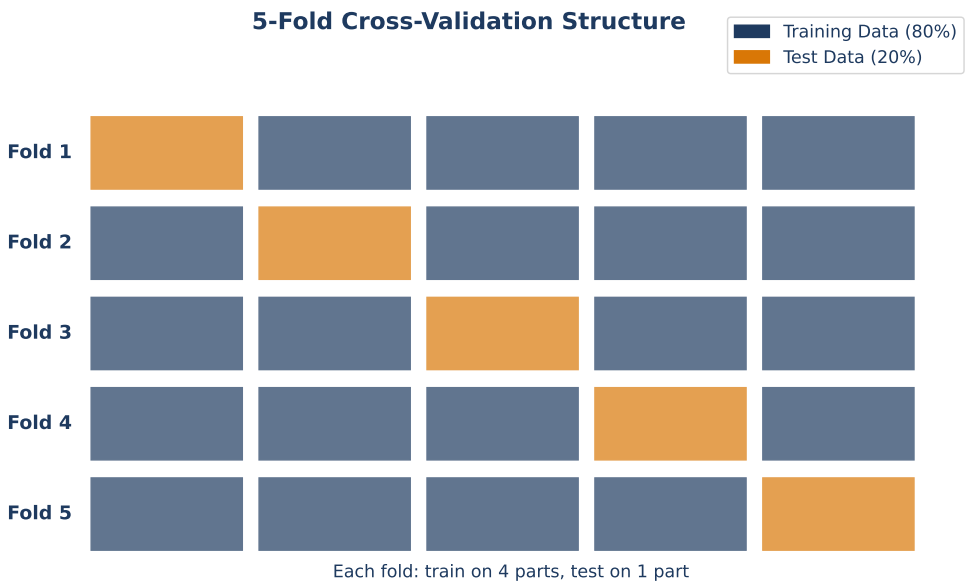


Figure 58: 5-fold cross-validation: data is split into 5 parts. Each round, one part is the test set (shaded) and the other four are the training set. Every observation is tested exactly once.

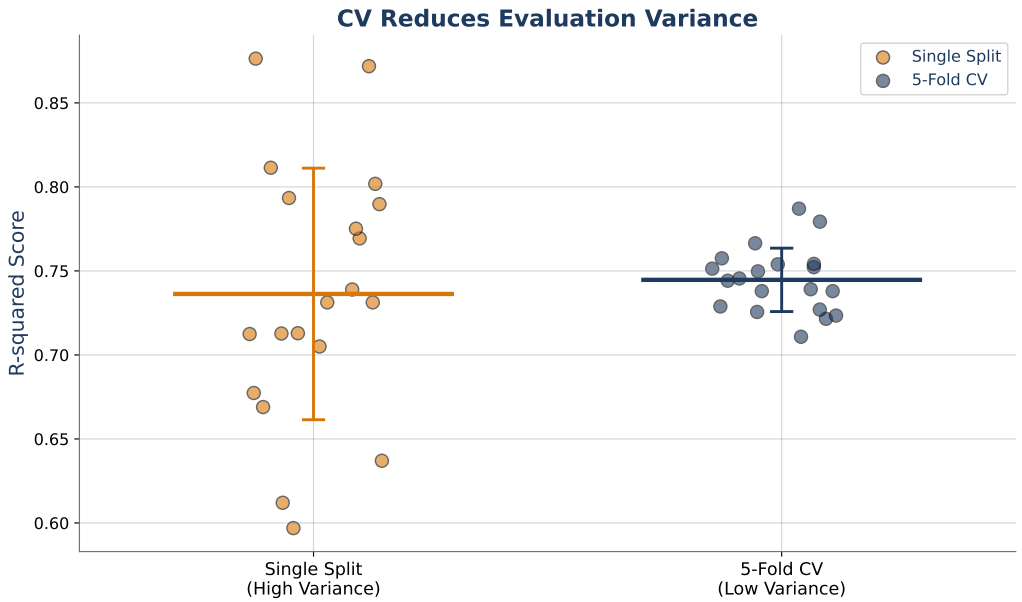


Figure 59: Cross-validation reduces variance: averaging over multiple folds gives a more stable estimate than any single split.

Key Formula: K-Fold CV Estimate

$$\text{CV}(K) = \frac{1}{K}\sum_{k=1}^{K} \text{Error}_k$$

where Error_k is the test metric (e.g., RMSE) on fold k.
Special cases:

- $K = n$ (leave-one-out CV, LOOCV): each fold is one observation. Low bias, high variance, computationally expensive.
- $K = 5$ or $K = 10$: good balance of bias and variance. Most common in practice.
- $K = 2$: high bias (trains on only half the data). Rarely used.

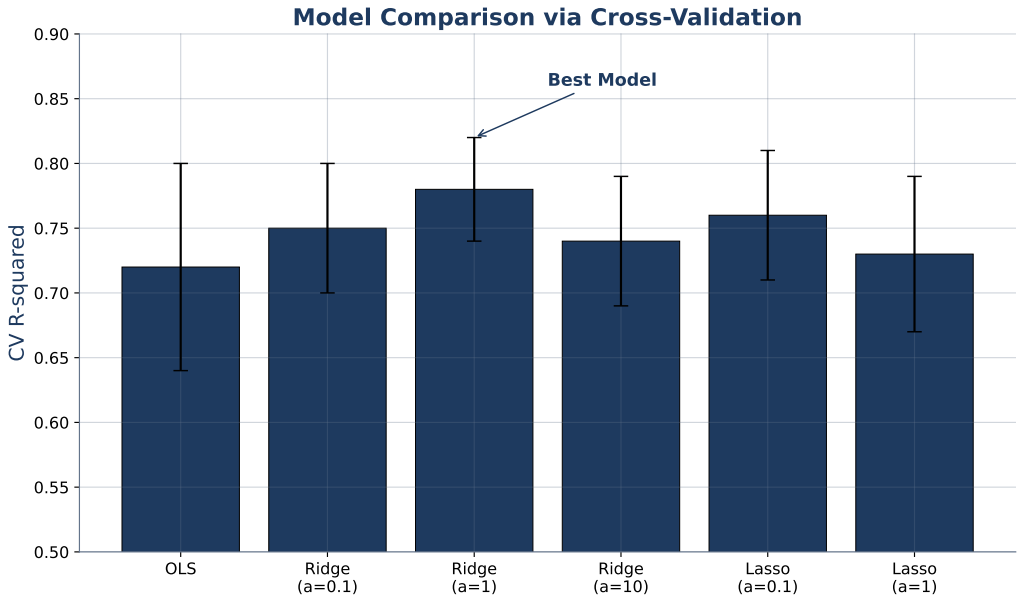


Figure 60: CV for model comparison: three models evaluated on the same folds. The model with the lowest average test error wins.

Grid search: A brute-force method for hyperparameter tuning: define a grid of candidate values (e.g., $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$), evaluate each with cross-validation, and select the value with the best CV score.

Time Series Cross-Validation

Standard K-fold CV shuffles data randomly into folds. For time series, this is catastrophic: the model trains on 2024 data and "predicts" 2019. Information from the future leaks into the training set, producing an R^2 that is far too optimistic. When the model encounters actual future data, it fails.

Figure 63 shows this leakage visually. In a random fold assignment, a data point from December 2023 might land in the training set while a point from January 2020 is in the test set. The model implicitly uses future information to predict the past.

Walk-forward validation: A time-series-aware CV method. In each round, the model trains on all data up to time t and predicts the next period $t+1$. The training window expands forward through time, never looking backward. This mimics how the model would actually be used in production.

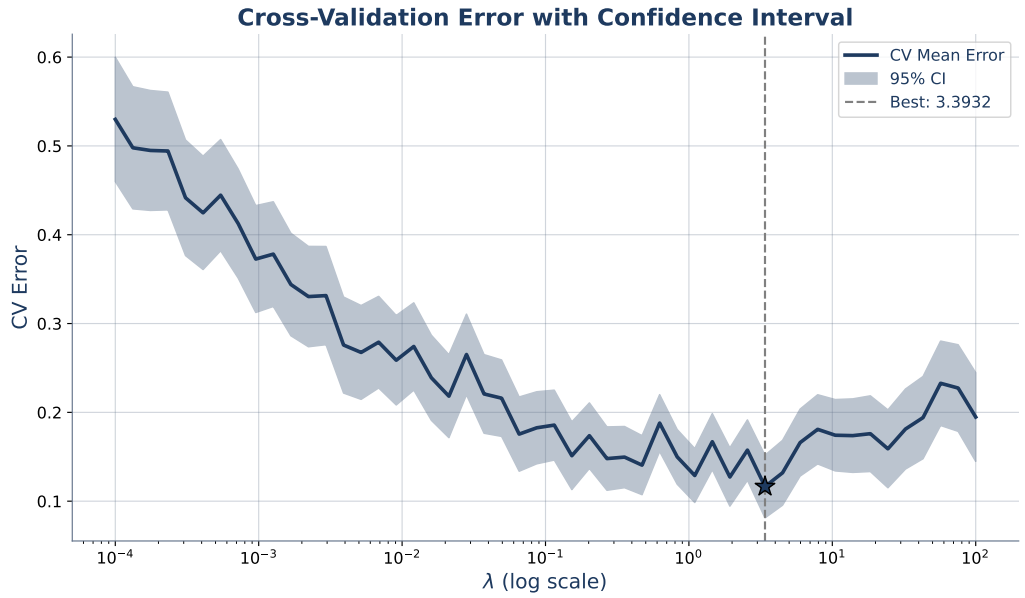


Figure 61: CV with confidence bands: the spread across folds gives an estimate of uncertainty. A model whose confidence band overlaps with another's is not significantly better.

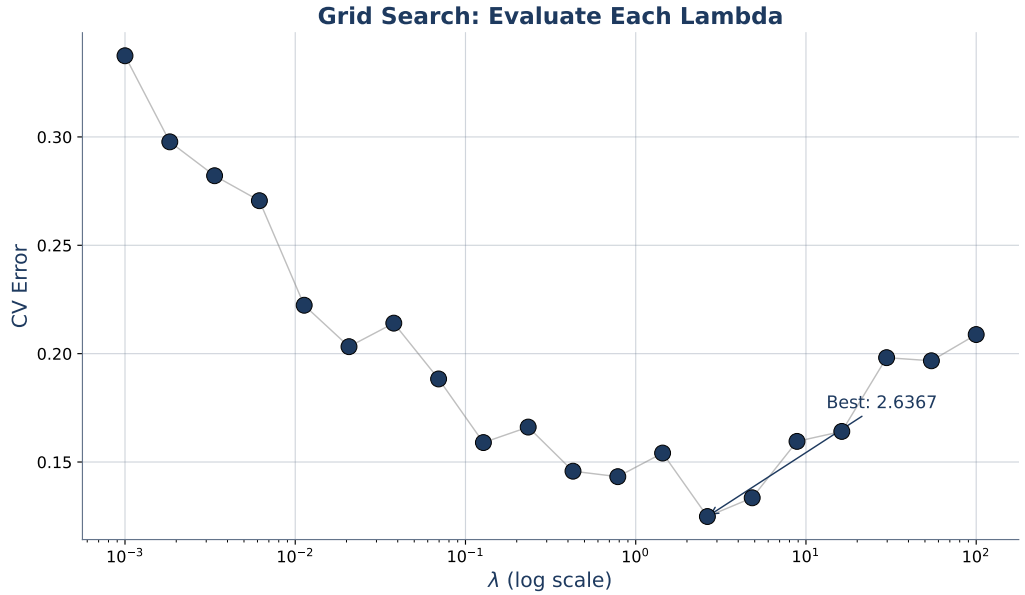


Figure 62: Grid search for λ: each candidate is evaluated by CV. The optimal λ minimizes the CV error curve.

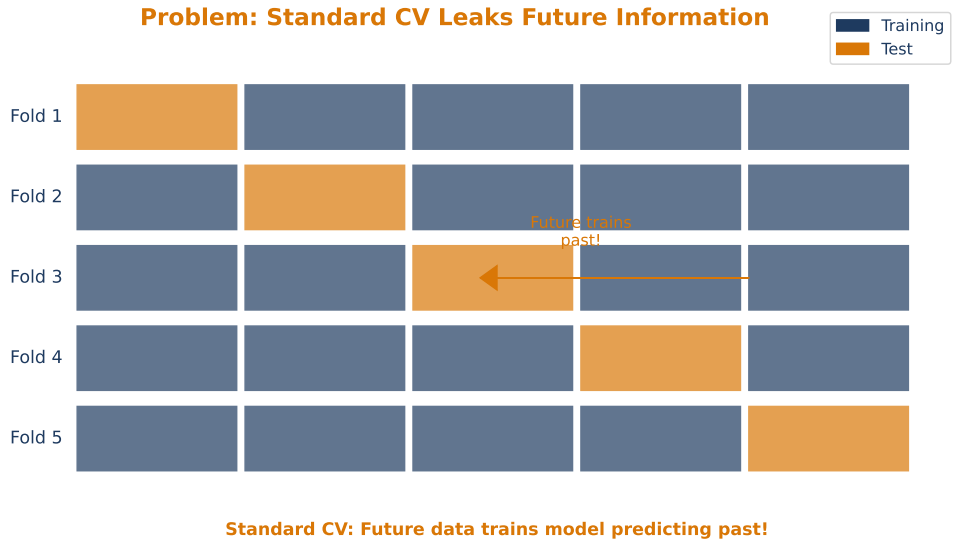


Figure 63: The problem with standard CV for time series: random shuffling allows future data to train the model, which then "predicts" the past. The resulting error estimate is too optimistic.

Key Formula: Walk-Forward Validation

For S splits with expanding window:

- **Split 1:** Train on months 1–12, test on months 13–14
- **Split 2:** Train on months 1–14, test on months 15–16
- **Split S:** Train on months 1–$(12 + 2(S-1))$, test on next 2 months

In sklearn: `TimeSeriesSplit(n_splits=5)`. The training window grows but never overlaps with the test window. The test window always comes *after* the training window in time.

Common Misconceptions about Cross-Validation

(1) "More folds in K-fold always means better." LOOCV ($K = n$) has the lowest bias but the highest variance: each training set differs by only one observation, so the K fitted models are nearly identical, and their test errors are highly correlated. The average of correlated estimates has higher variance than the average of independent ones. $K = 5$ or $K = 10$ gives a better bias-variance balance.
(2) "Random K-fold works for time series." It does not. Random shuffling breaks temporal ordering, allowing the model to train on future data. The CV error estimate is overly optimistic. Always use walk-forward validation for time series.
(3) "Cross-validation tells you the true error." CV estimates the *expected* error of a model trained on a dataset of this size from this process. The true error on tomorrow's data will differ because of randomness, regime changes, and distribution shifts. CV is the best estimate available, but it is still an estimate.

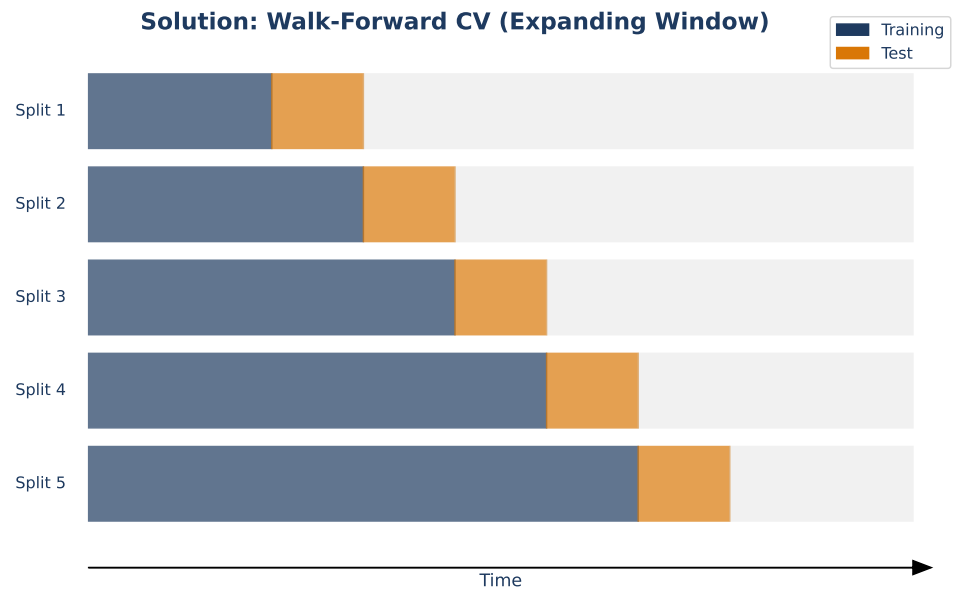


Figure 64: Walk-forward validation: the training window expands forward. Each test set comes strictly after its training set. No future information leaks into the past.

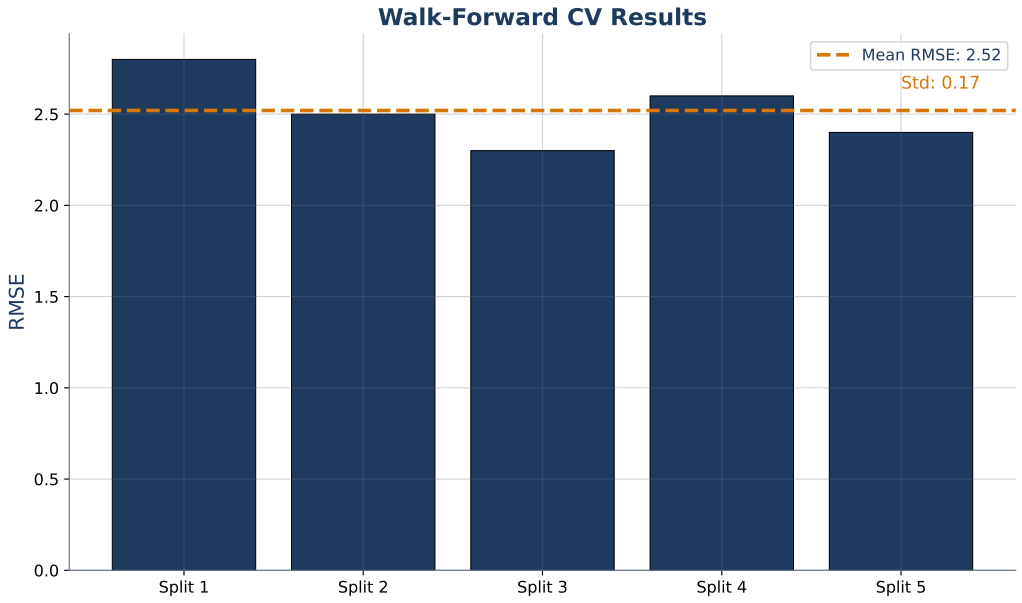
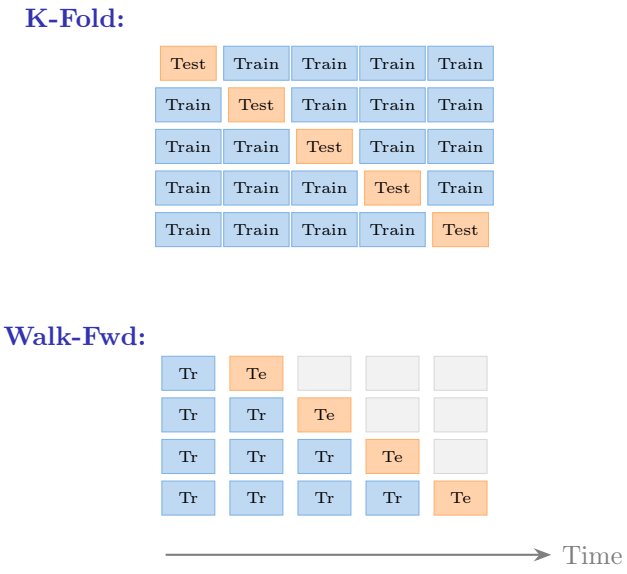


Figure 65: Walk-forward results across splits: test performance varies as market conditions change. The average gives a realistic out-of-sample estimate.



K-fold (top) shuffles data randomly—fine for cross-sectional data like house prices. Walk-forward (bottom) respects time order—mandatory for financial time series. The test set always comes after the training set.

Worked Examples

Worked Example 1: 5-Fold CV for Hyperparameter Selection

You want to choose between Ridge with $\lambda = 0.01$, $\lambda = 0.1$, and $\lambda = 1.0$. Your dataset has 500 observations.
Step 1: Split into 5 folds of 100 observations each.
Step 2: For each λ, run 5 rounds of train/test. Record the test RMSE each round.

λ	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean RMSE
0.01	2.31	2.45	2.28	2.50	2.36	2.38
0.10	2.15	2.22	2.18	2.25	2.20	**2.20**
1.00	2.40	2.48	2.42	2.51	2.45	2.45

Result: $\lambda = 0.1$ has the lowest average RMSE. Select it. Also note: the spread across folds for $\lambda = 0.1$ (range: 2.15–2.25) is tighter than for $\lambda = 0.01$ (range: 2.28–2.50), indicating that $\lambda = 0.1$ is both better on average and more stable.

Worked Example 2: Walk-Forward for Monthly Returns

You have 60 months of stock return data (Jan 2018 – Dec 2022). You want to evaluate a linear factor model using walk-forward validation with an initial training window of 36 months.

Split 1: Train on months 1–36 (Jan 2018 – Dec 2020). Test on months 37–42 (Jan – Jun 2021). RMSE = 3.1%.

Split 2: Train on months 1–42 (Jan 2018 – Jun 2021). Test on months 43–48 (Jul – Dec 2021). RMSE = 2.8%.

Split 3: Train on months 1–48 (Jan 2018 – Dec 2021). Test on months 49–54 (Jan – Jun 2022). RMSE = 4.2%.

Split 4: Train on months 1–54 (Jan 2018 – Jun 2022). Test on months 55–60 (Jul – Dec 2022). RMSE = 3.5%.

Average RMSE: $(3.1 + 2.8 + 4.2 + 3.5)/4 = 3.4\%$.

The elevated RMSE in Split 3 corresponds to the volatile first half of 2022. Walk-forward captures this regime sensitivity—standard K-fold CV would average over it and hide the model's weakness during volatile periods.

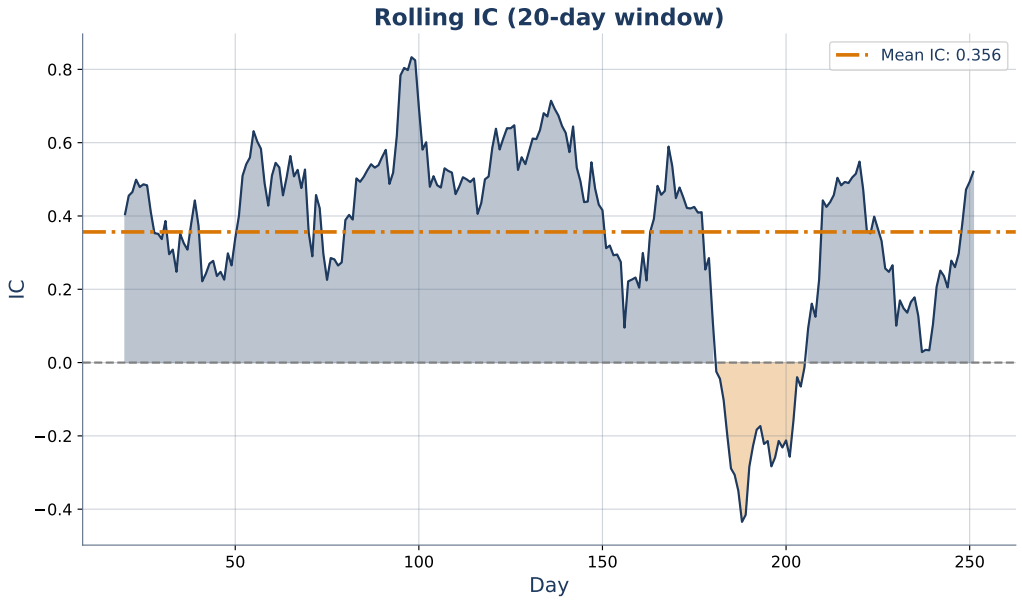


Figure 66: Rolling information coefficient over time: signal strength varies as market conditions change. A stable IC indicates a robust model; a decaying IC suggests the signal is weakening.

Historical Background: Mervyn Stone and Cross-Validation (1974)

The idea of testing a model on data it has not seen is intuitive, but formalizing it took time. Mervyn Stone, a British statistician, published a landmark paper in 1974 in the *Journal of the Royal Statistical Society* titled "Cross-Validatory Choice and Assessment of Statistical Predictions." Stone showed that instead of arguing about which model has the best theoretical properties, you can simply ask each model to predict data it was not trained on and let the predictions speak for themselves.

Stone proved that leave-one-out cross-validation (LOOCV) is asymptotically equivalent to Akaike's Information Criterion (AIC), connecting the resampling approach to information theory. This equivalence gave CV a theoretical foundation: it was not just a practical hack but a principled method for model selection.

Before Stone, model selection was dominated by hypothesis testing and information criteria. After Stone, practitioners had a distribution-free, assumption-light method that worked for any model. CV became the gold standard for choosing between algorithms, tuning hyperparameters, and estimating generalization error. It remains so today.

Problem 6.1 (Easy)

Draw a diagram showing 5-fold cross-validation for a dataset with 20 observations. Label which observations belong to which fold, and indicate which fold is the test set in each round. How many times is each observation used for testing?

Solution: see Appendix.

Problem 6.2 (Easy)

Explain in two or three sentences why you cannot shuffle time series data before splitting into K folds. Use a concrete example with years to make your point.

Solution: see Appendix.

Problem 6.3 (Medium)

A 5-fold CV produces the following test RMSE values: 3.2, 2.8, 3.5, 3.0, 3.1. Compute:

(a) The CV estimate of RMSE.

(b) The standard deviation of the fold RMSEs.

(c) A 95% confidence interval for the true RMSE, using mean $\pm\, 2 \times \frac{\text{sd}}{\sqrt{K}}$.

Solution: see Appendix.

Problem 6.4 (Medium)

You have 120 months of return data (10 years). Design a walk-forward validation scheme with:

- Initial training window: 60 months
- Test window: 12 months
- Expanding training window (no sliding)

How many splits do you get? Write out the training and test ranges for each split.
Solution: see Appendix.

Problem 6.5 (Hard)

Compare LOOCV and 10-fold CV analytically.

(a) Which has lower bias? Explain why. (Hint: how much training data does each use?)

(b) Which has lower variance? Explain why. (Hint: how correlated are the K fitted models?)

(c) Given the bias-variance tradeoff in CV itself, argue why $K = 10$ is often preferred to $K = n$ in practice.

Solution: see Appendix.

Connecting Backward and Forward

We now have every piece of the regression toolkit:

Section	Tool	Purpose	Key Output
1	OLS	Fit the line	β_0, β_1
2	LINE checks	Validate assumptions	Diagnostic plots
3	Bias-variance	Diagnose overfitting	Train/test gap
4	Ridge/Lasso	Control complexity	Regularized β
5	Metrics	Measure quality	RMSE, R^2
6	CV	Select model honestly	CV score

All of these tools were built in abstract terms: "a feature x," "a target y." But in finance, those abstractions have very specific names. The most important is the *market factor*: the return of the broad stock market. A regression of a stock's return on the market's return gives a number called *beta*—and that number is at the heart of the most famous model in all of finance: the Capital Asset Pricing Model.

What is beta? Why did its inventor win the Nobel Prize? And what happens when a single factor is not enough? Sections 7 and 8 answer these questions.

Key Takeaway: Cross-validation estimates how your model will perform on data it has never seen—for time series, always validate forward in time, never backward.

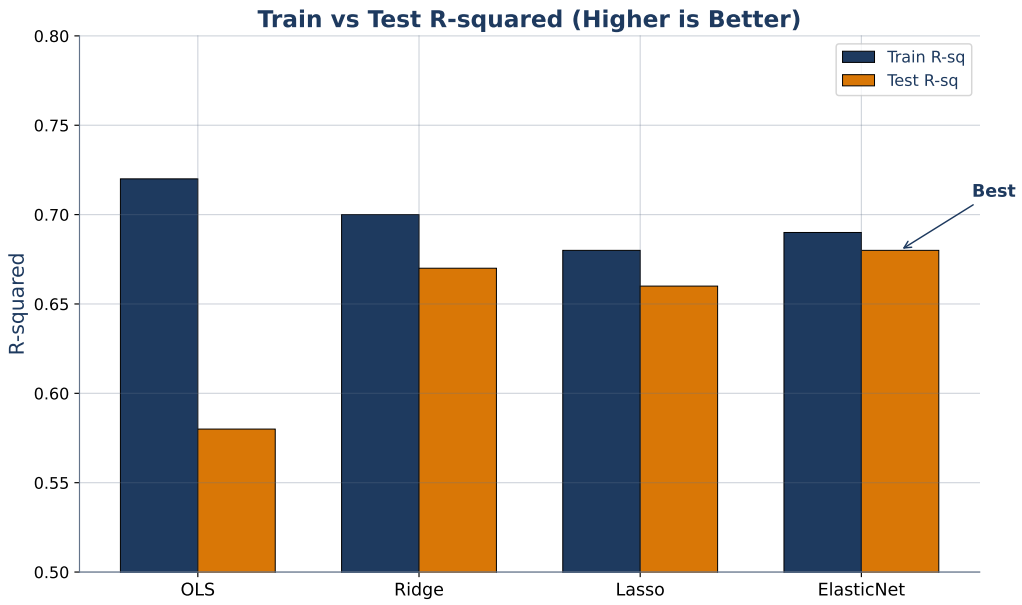


Figure 67: Model comparison using R^2 from cross-validation. The model with the highest out-of-sample R^2—not the highest in-sample R^2—is the winner.

7 From One Factor to Many – CAPM Beta and the Birth of Factor Models

Opening Problem: Tesla Drops 40%

In early 2022, Tesla stock dropped roughly 40% over a few months. During the same period, the S&P 500 fell about 5%. A student calculates: "40%/5% = 8. Tesla's beta is 8." Her classmate shakes his head: "That is not how beta works."

The student's arithmetic is correct for that particular period, but her interpretation is wrong. Beta is not the ratio of two specific moves. It is the *slope* of a regression line—the systematic relationship between the stock's excess return and the market's excess return, estimated over hundreds of days. Some of Tesla's 40% drop was due to the market decline (systematic risk, captured by beta). Some was due to Tesla-specific news: production issues, Elon Musk selling shares, rising interest rates hitting high-valuation tech stocks (idiosyncratic risk, not captured by beta).

Disentangling these two sources of risk is exactly what the Capital Asset Pricing Model does. This section shows you how.

Discovery Question

Tesla stock dropped 40% in a month while the market dropped only 5%. Does that mean Tesla's beta is 8? If not, what does beta actually measure?

One Number That Summarizes Market Sensitivity

Every stock moves partly because the overall market moves and partly for its own reasons. Apple rises on a good market day because the tide lifts all boats. Apple also rises on an iPhone launch day because of Apple-specific news. Beta captures only the first component.

Think of it as a thermostat. The market is the outdoor temperature. Each stock is a room in a building with a different thermostat setting. A room with a thermostat set to "high sensitivity" (beta > 1) amplifies outdoor temperature changes: when it is warm outside, the room is very

warm; when it is cold, the room is very cold. A room with "low sensitivity" (beta < 1) stays moderate regardless.

Figure 68 shows the fundamental picture: stock excess returns on the vertical axis, market excess returns on the horizontal axis, and a regression line through the scatter. The slope of that line is beta.

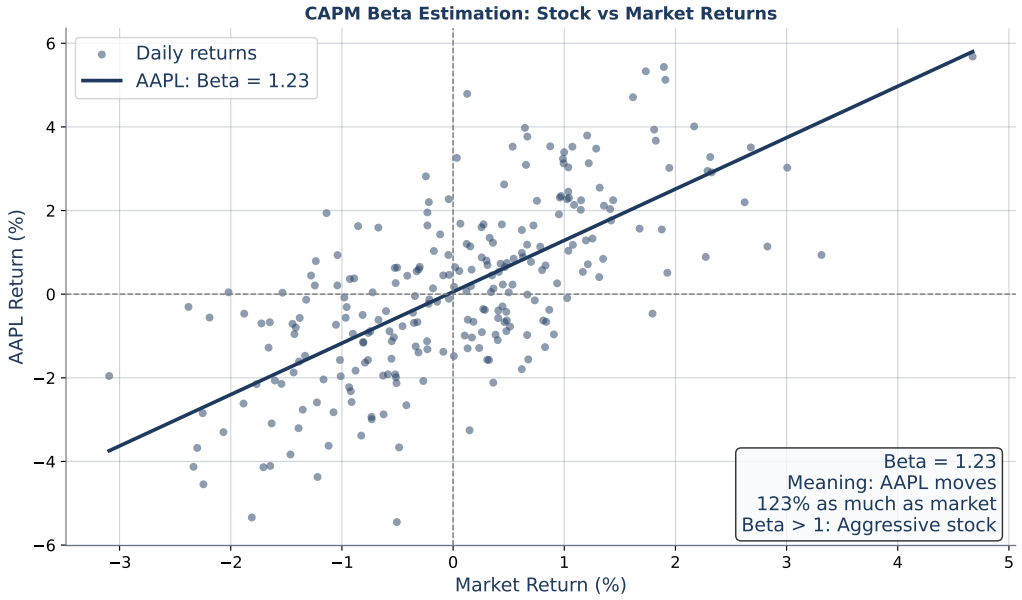


Figure 68: CAPM beta: the slope of the regression of stock excess returns on market excess returns. Each dot is one day. The slope tells you how much the stock moves per unit of market movement.

The CAPM Equation

Excess return: The return of an asset minus the risk-free rate: $R_i - R_f$. The risk-free rate is typically the yield on short-term government bonds (e.g., 3-month U.S. Treasury bills). Excess return measures how much extra return you earned for taking risk beyond the riskless alternative.

Beta (β): The slope of the regression of a stock's excess return on the market's excess return. Beta measures *systematic risk*—the portion of a stock's movement attributable to the overall market. $\beta = 1$ means the stock moves in lockstep with the market. $\beta > 1$ means the stock amplifies market moves. $\beta < 1$ means it dampens them.

Alpha (α): The intercept of the regression of excess returns on market excess returns. Alpha represents the return that is not explained by market exposure. Positive alpha suggests the stock (or fund) outperformed its market-adjusted benchmark. In practice, persistent positive alpha is rare.

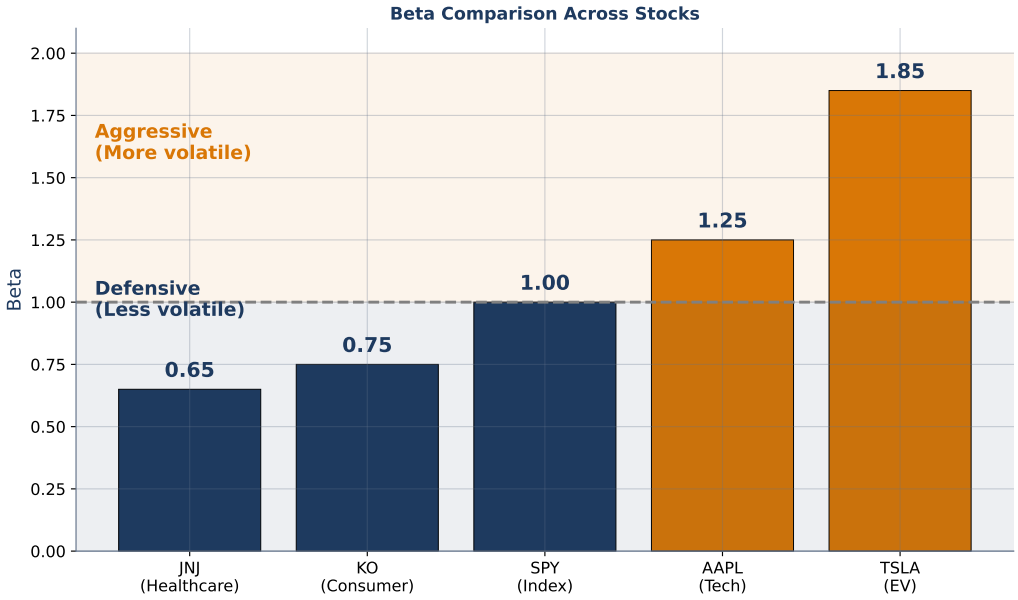


Figure 69: Beta comparison across stocks. A defensive utility stock might have $\beta = 0.5$; a volatile tech stock might have $\beta = 1.5$. The market itself has $\beta = 1$ by definition.

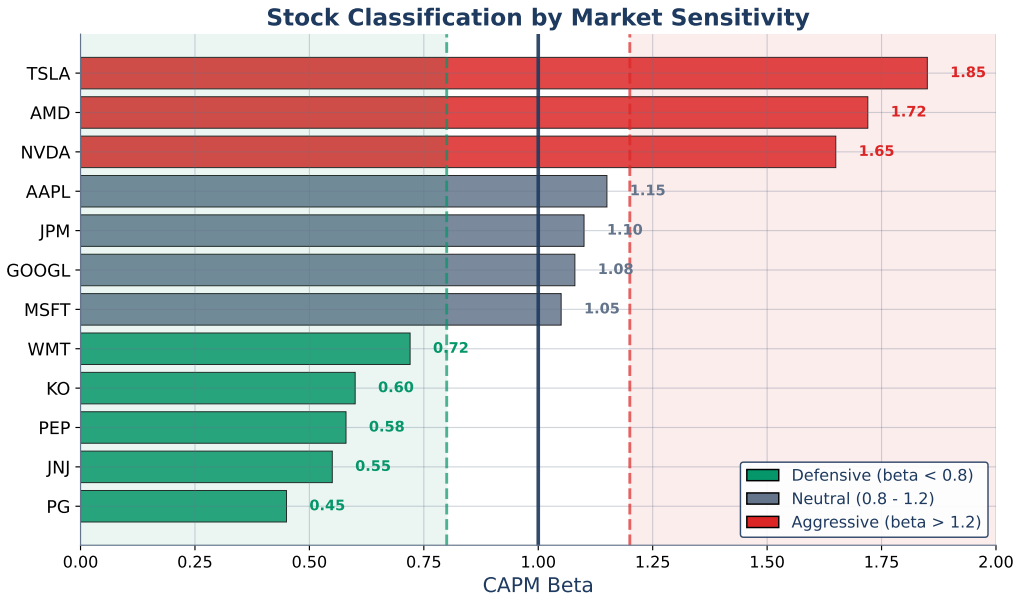


Figure 70: Beta classification bands: $\beta < 0.8$ is defensive, $0.8 < \beta < 1.2$ is market-tracking, $\beta > 1.2$ is aggressive.

Key Formula: The CAPM Regression

$$R_i - R_f = \alpha + \beta(R_m - R_f) + \varepsilon$$

where:

- R_i is the stock's return in a given period
- R_f is the risk-free rate (e.g., Treasury bill yield)
- R_m is the market return (e.g., S&P 500)
- α is the intercept (alpha—excess return beyond what beta explains)
- β is the slope (market sensitivity)
- ε is the error term (idiosyncratic risk—stock-specific news)

Plain English: A stock's excess return is its alpha plus beta times the market's excess return, plus random noise.

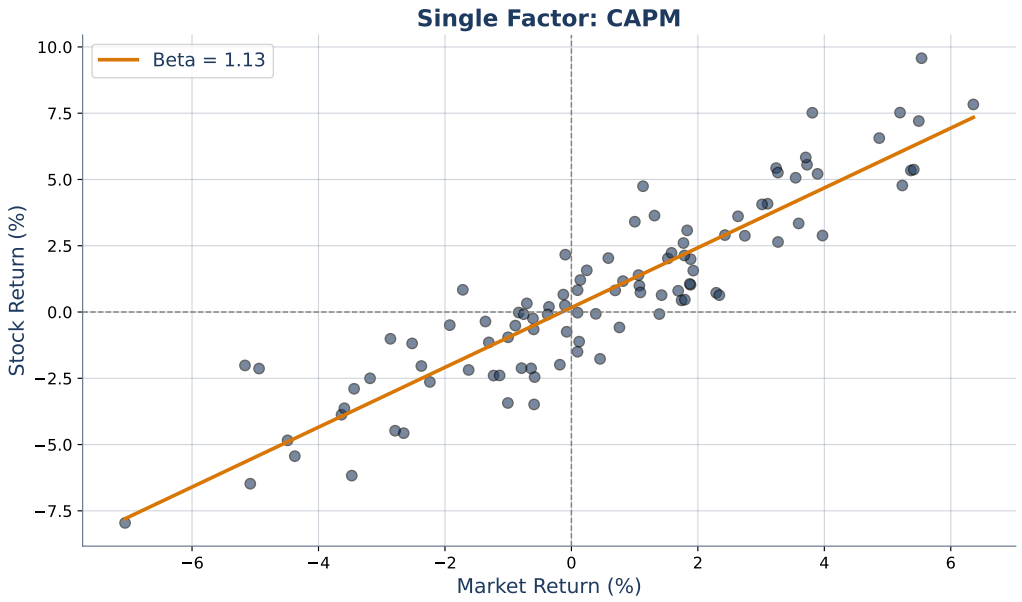
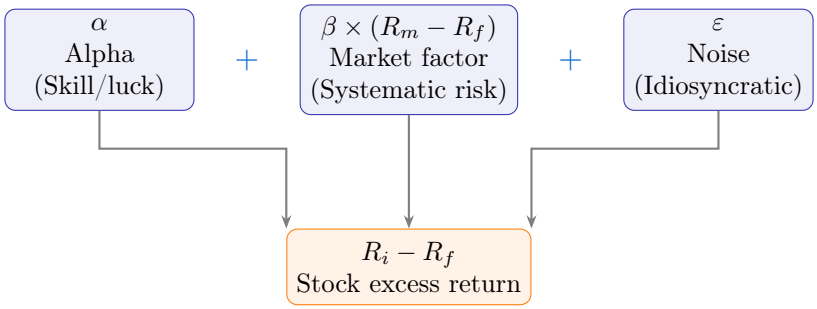


Figure 71: CAPM single-factor regression: stock excess returns vs. market excess returns. The slope is beta; the intercept is alpha.



The CAPM decomposes every stock's excess return into three pieces: alpha (unexplained outperformance), the market factor scaled by beta (systematic risk), and noise (everything else).

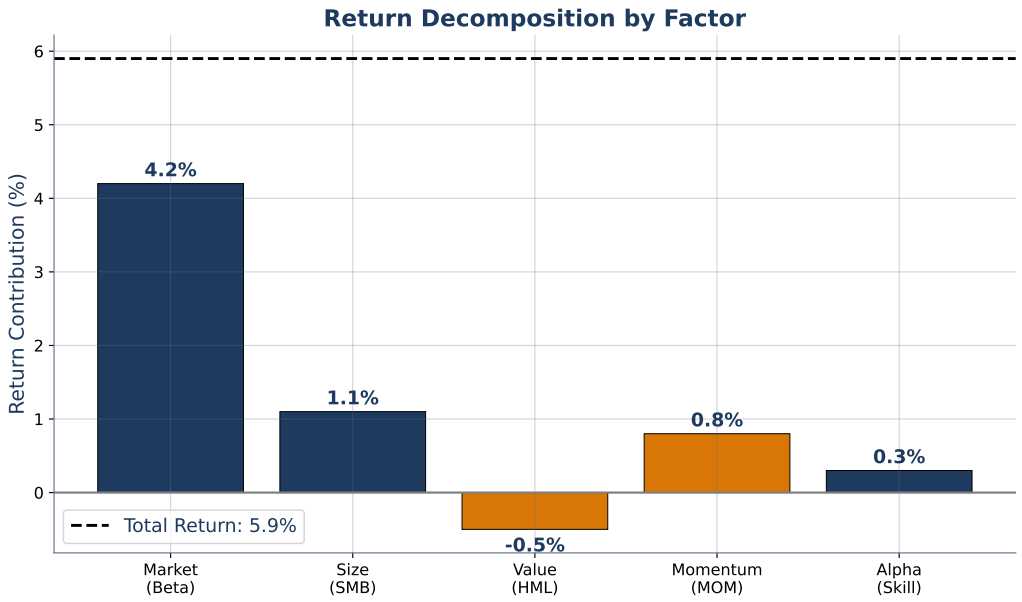


Figure 72: Return decomposition: a stock's return splits into the risk-free rate, the market-driven component ($\beta\times$ market excess return), and alpha plus noise.

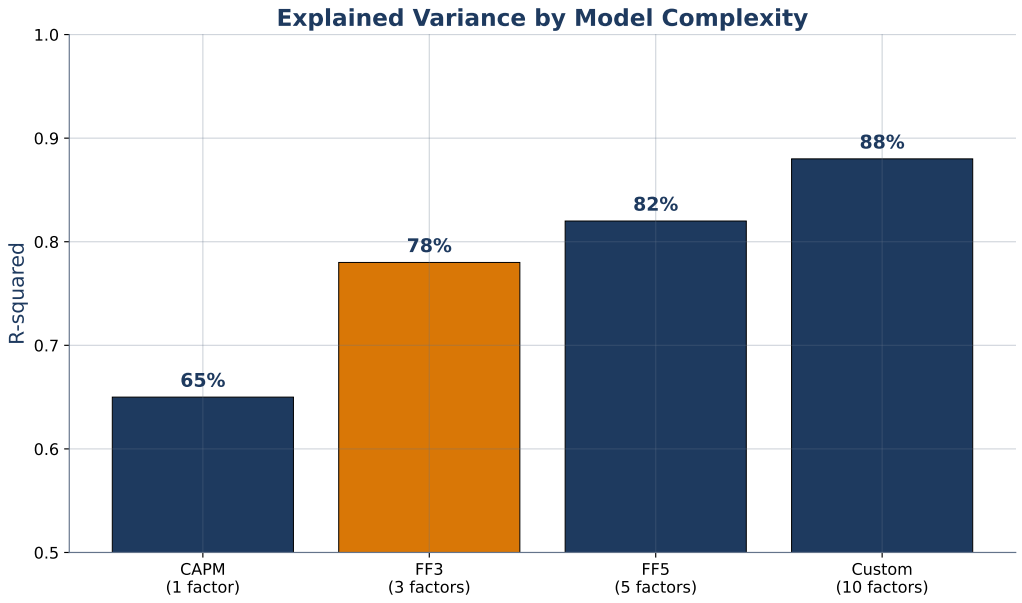


Figure 73: Explained variance by factor count. A single factor (market) explains roughly 60% of a typical stock's return variance. The remaining 40% is idiosyncratic.

Most of the action, for most stocks, is in the middle piece.

Common Misconceptions about CAPM and Beta

(1) "Beta measures how risky a stock is." Beta measures *market* risk—how much of the stock's movement comes from market-wide forces. It says nothing about total risk. A biotech startup might have $\beta = 0.8$ (low market sensitivity) but enormous total volatility from drug trial results.

(2) "A stock with $\beta > 1$ will always outperform the market." Beta is about sensitivity, not direction. When the market rises 2%, a stock with $\beta = 1.5$ is expected to rise 3%. When the market falls 2%, the same stock is expected to fall 3%. Beta amplifies both gains and losses.

(3) "Alpha is the manager's skill." Measured alpha depends entirely on what factors you include. A fund with $\alpha = 3\%$ under CAPM might have $\alpha = -1\%$ under the Fama-French model (Section 8). Alpha is what is *left over* after accounting for known risk factors. Adding more factors shrinks the unexplained residual.

Definition: Systematic vs. Idiosyncratic Risk

Systematic risk is the portion of a stock's movement driven by market-wide factors: economic growth, interest rates, inflation, geopolitical events. It is captured by beta and *cannot* be diversified away.

Idiosyncratic risk is the portion driven by stock-specific events: a product launch, a lawsuit, a CEO resignation. It appears as the residual ε in the CAPM regression and *can* be diversified away by holding many stocks.

A well-diversified portfolio has near-zero idiosyncratic risk. Its return is dominated by systematic risk—by market beta. This is why CAPM says that only beta should be compensated with higher expected returns.

Worked Examples

Worked Example 1: Computing Beta by Hand

You have five months of excess returns:

Month	Market excess (%)	Stock excess (%)
1	2.0	3.4
2	−1.5	−2.0
3	0.5	1.0
4	3.0	4.8
5	−2.0	−3.5

Step 1: $\bar{x} = (2.0-1.5+0.5+3.0-2.0)/5 = 0.4\%$. $\bar{y} = (3.4-2.0+1.0+4.8-3.5)/5 = 0.74\%$.
Step 2: $\text{Cov}(x,y) = \frac{1}{5}\sum(x_i - 0.4)(y_i - 0.74) = \frac{1}{5}[(1.6)(2.66) + (-1.9)(-2.74) + (0.1)(0.26) + (2.6)(4.06) + (-2.4)(-4.24)] = \frac{1}{5}[4.256 + 5.206 + 0.026 + 10.556 + 10.176] = \frac{30.22}{5} = 6.044$.
Step 3: $\text{Var}(x) = \frac{1}{5}\sum(x_i - 0.4)^2 = \frac{1}{5}[2.56 + 3.61 + 0.01 + 6.76 + 5.76] = \frac{18.70}{5} = 3.740$.
Step 4: $\beta = 6.044/3.740 = 1.62$.
Interpretation: This stock moves 1.62% for every 1% market move. It is aggressive—amplifying both up days and down days by 62%.
Step 5: $\alpha = \bar{y} - \beta\bar{x} = 0.74 - 1.62(0.4) = 0.74 - 0.648 = 0.09\%$.
A monthly alpha of 0.09% is small and probably not statistically significant with only five data points. More data would be needed to draw conclusions about alpha.

Worked Example 2: CAPM Residuals Reveal Missing Factors

You run a CAPM regression for a small-cap value stock and examine the residuals. The CAPM predicts the stock should return about the same as the market (adjusted for beta), but the residuals show a persistent positive pattern: the stock consistently outperforms its CAPM prediction.
Figure 74 shows this. The residuals are not random—they cluster above zero, especially during periods when small and value stocks outperform. This pattern suggests the CAPM is missing something. The stock's returns are driven by factors beyond the market alone. This is exactly the observation that led Fama and French to propose additional factors. If the residuals have a pattern, the model is missing a variable. Section 8 adds those variables.

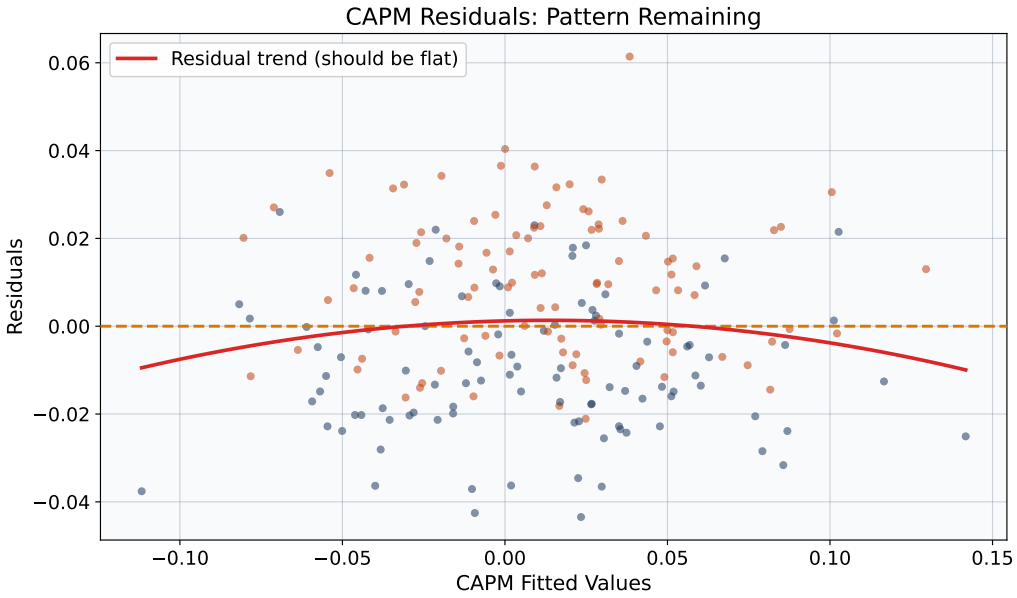


Figure 74: CAPM residuals for a small-cap value stock: persistent positive residuals indicate that CAPM underestimates the stock's expected return. A missing factor is at work.

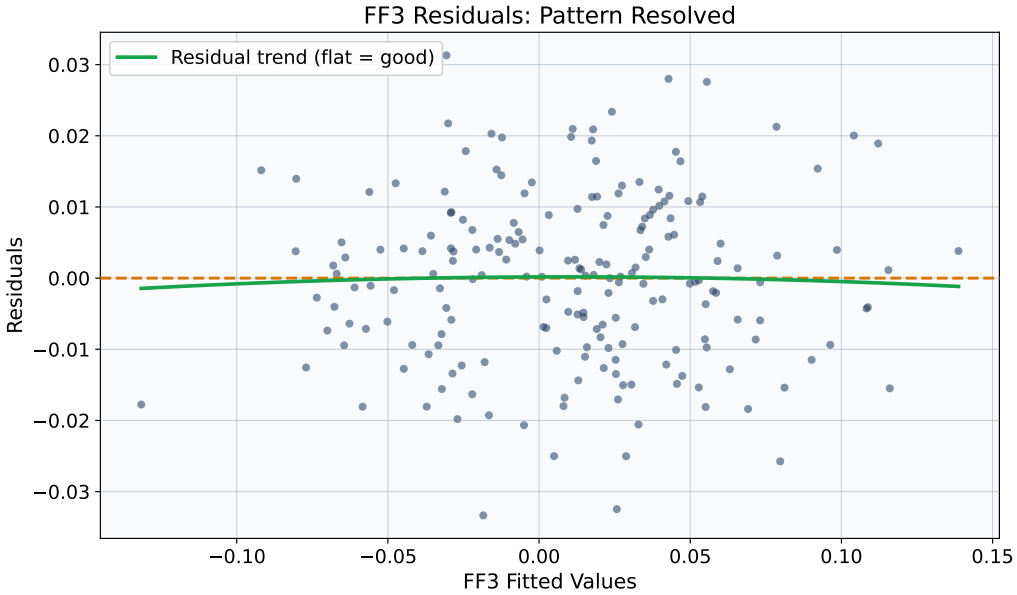


Figure 75: After adding the Fama-French size and value factors, the same stock's residuals look random. The pattern is resolved—the missing factors have been found.

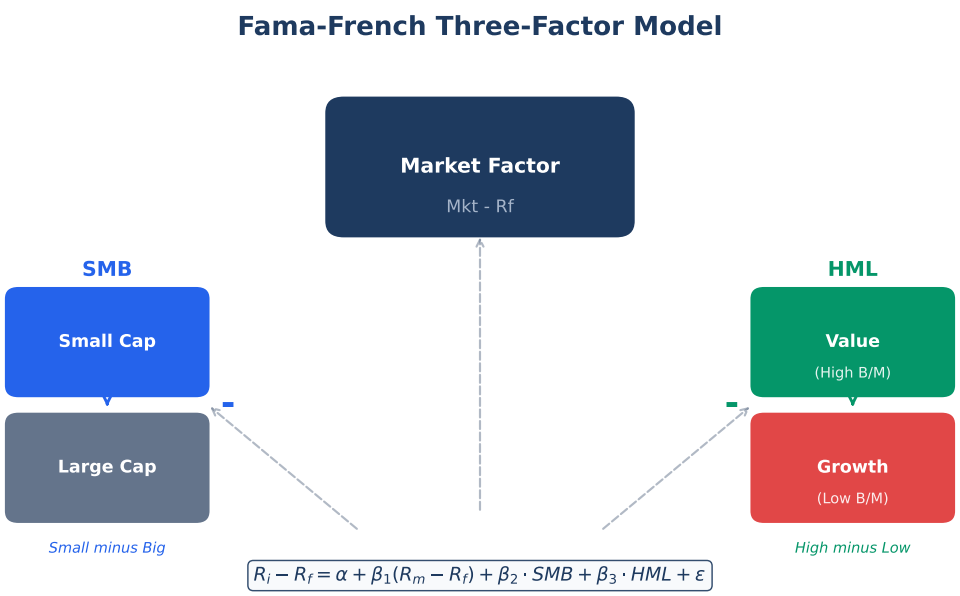


Figure 76: The Fama-French concept: one factor (market) leaves systematic patterns in the residuals. Adding size and value factors captures those patterns.

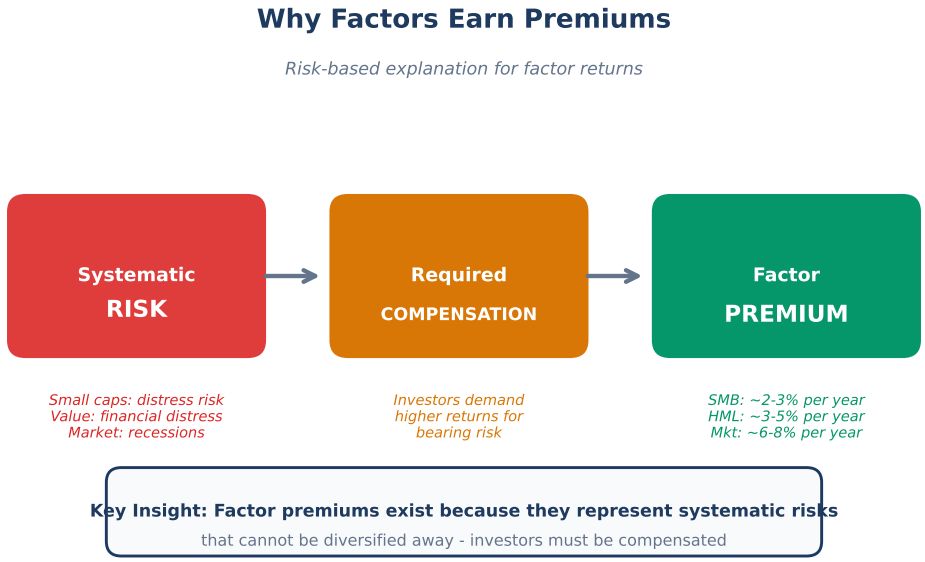


Figure 77: Factor premium logic: size and value premia exist because small and value stocks are riskier. Investors demand higher returns for bearing that risk.

Historical Background: William Sharpe and the CAPM (1964)

William Sharpe was a young economist at the RAND Corporation when he published "Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk" in 1964. The paper proposed a startlingly simple idea: in equilibrium, the expected excess return of any asset is proportional to its beta—its sensitivity to the market portfolio.

The implication was radical. It meant that the only risk investors should be compensated for is market risk. Stock-specific risk can be diversified away by holding a portfolio. Therefore, a stock with high idiosyncratic volatility but low beta should earn the same expected return as a stock with low idiosyncratic volatility and the same beta.

Sharpe shared the 1990 Nobel Prize in Economics with Harry Markowitz and Merton Miller. His model, despite its known empirical shortcomings (the anomalies that Fama and French later documented), remains the foundation of asset pricing. Every finance student learns the CAPM equation. Every portfolio manager calculates beta. And every quantitative analyst runs the CAPM regression as a starting point—even if they plan to add more factors.

The CAPM's lasting contribution is not that it is perfectly correct, but that it provides a framework for thinking about risk and return in terms of exposure to systematic factors.

Problem 7.1 (Easy)

A scatter plot of stock excess returns vs. market excess returns shows a clear upward trend with slope approximately 1.3. The intercept is close to zero.

(a) What is this stock's approximate beta?

(b) If the market rises 2% above the risk-free rate, what does CAPM predict for this stock's excess return?

(c) Is this stock aggressive or defensive?

Solution: see Appendix.

Problem 7.2 (Easy)

Interpret a beta of 1.3 for a technology stock in plain English. Cover three scenarios: (a) the market rises 3%, (b) the market falls 2%, (c) the market is flat. What is the expected stock excess return in each case if $\alpha = 0$?

Solution: see Appendix.

Problem 7.3 (Medium)

The risk-free rate is 4% per year. The expected market return is 10% per year. A stock has $\beta = 0.8$.

(a) Using the CAPM, compute the stock's expected return.

(b) If the stock actually returns 9% this year, what is its realized alpha?

(c) Is this alpha likely to be statistically significant? Why or why not?

Solution: see Appendix.

Problem 7.4 (Medium)

A CAPM regression for a small-cap stock produces residuals with a clear positive bias—the stock consistently outperforms its CAPM prediction by about 0.3% per month.

(a) What does this pattern suggest about the adequacy of the CAPM for this stock?

(b) Name two factors that might explain this outperformance.

(c) How would adding those factors change the residual pattern?

Solution: see Appendix.

Problem 7.5 (Hard)

You are given 60 months of data for a stock and the market. Run the CAPM regression (by hand or describe the steps precisely):

(a) Write out the regression equation with symbols.

(b) Explain what each output (slope, intercept, R^2, residuals) tells you.

(c) If $R^2 = 0.55$, interpret: what fraction of the stock's return variance is explained by the market? What fraction is idiosyncratic?

(d) Explain why a low R^2 does not necessarily mean the CAPM is wrong—it means the stock has high idiosyncratic risk.

Solution: see Appendix.

Connecting Backward and Forward

The CAPM is a single-factor regression model. We estimated it using OLS (Section 1), checked its residuals (Section 2), and evaluated it with R^2 and walk-forward validation (Sections 5–6). For many stocks, the CAPM explains about 60% of return variance. That leaves 40% unexplained.

Some of that 40% is genuinely random—stock-specific news that no model can predict. But some of it is *systematic*: small stocks behave differently from large stocks, and cheap "value" stocks behave differently from expensive "growth" stocks. These patterns appear as residual patterns in the CAPM regression—exactly the kind of diagnostic failure we learned to detect in Section 2.

In 1993, Eugene Fama and Kenneth French proposed a solution: add two more factors. The three-factor model captures market risk, size risk, and value risk in a single multiple regression. Section 8 builds that model and shows how it transforms the way we measure alpha and attribute returns.

Key Takeaway: Beta is not a measure of total risk but of market sensitivity—a stock's beta tells you how much of its movement is explained by the market.

8 One Factor Is Never Enough – Fama-French and Multi-Factor Regression

Opening Problem: The Disappearing Alpha

A hedge fund manager reports to his investors: "Our annual alpha is 3% using the CAPM as our benchmark. We outperform the market consistently." The fund charges a 2% management fee for this skill.

An independent analyst re-runs the numbers. She uses the Fama-French 3-factor model instead of the CAPM. Under this model, the fund's alpha is −1%. The manager has not outperformed—he has underperformed, once you account for the fact that his portfolio tilts toward small-cap and value stocks.

The manager's "skill" was actually exposure to two well-known risk premiums: small stocks tend to outperform large stocks, and value stocks tend to outperform growth stocks. The CAPM did not account for these tilts, so their premium showed up in the intercept (alpha). The Fama-French model absorbs those premiums into the SMB and HML factors, leaving only the true unexplained residual as alpha.

The fund's actual performance has not changed. The definition of what counts as "alpha" has changed—because the benchmark has changed. This section shows you how to build multi-factor models and why the choice of factors determines everything.

Discovery Question

A hedge fund manager reports annual alpha of 3% using CAPM as the benchmark. When an analyst re-runs the numbers using the Fama-French 3-factor model, the alpha drops to −1%. Has the manager's actual performance changed? What has changed?

Why One Factor Falls Short

In Section 7 we saw that the CAPM explains about 60% of a typical stock's return variance. The remaining 40% is treated as noise. But look more carefully at that "noise." Sort stocks into groups and examine the residuals:

- *Small* stocks consistently have positive CAPM residuals: they earn more than their beta predicts.
- *Value* stocks (high book-to-market ratio, "cheap" stocks) also have positive CAPM residuals.
- *Large growth* stocks tend to have slightly negative CAPM residuals.

These are not random patterns—they are *systematic* departures from the CAPM's predictions. The CAPM residuals contain structure. And as Section 2 taught us, structured residuals mean the model is missing a variable.

SMB (Small Minus Big): A portfolio that goes long small-cap stocks and short large-cap stocks. The return of SMB captures the size premium: the historical tendency of small stocks to outperform large stocks. Positive SMB loading means a stock behaves like a small cap.

HML (High Minus Low): A portfolio that goes long high book-to-market ("value") stocks and short low book-to-market ("growth") stocks. The return of HML captures the value premium: the historical tendency of cheap stocks to outperform expensive ones. Positive HML loading means a stock behaves like a value stock.

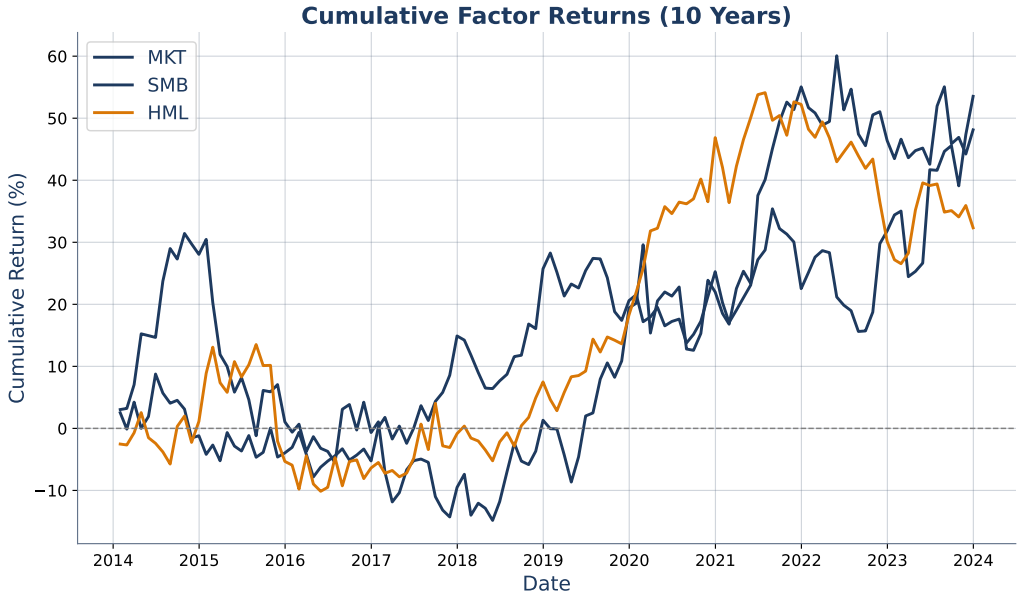


Figure 78: Cumulative returns of the three Fama-French factors over several decades. All three premia are positive in the long run, though each has periods of underperformance.

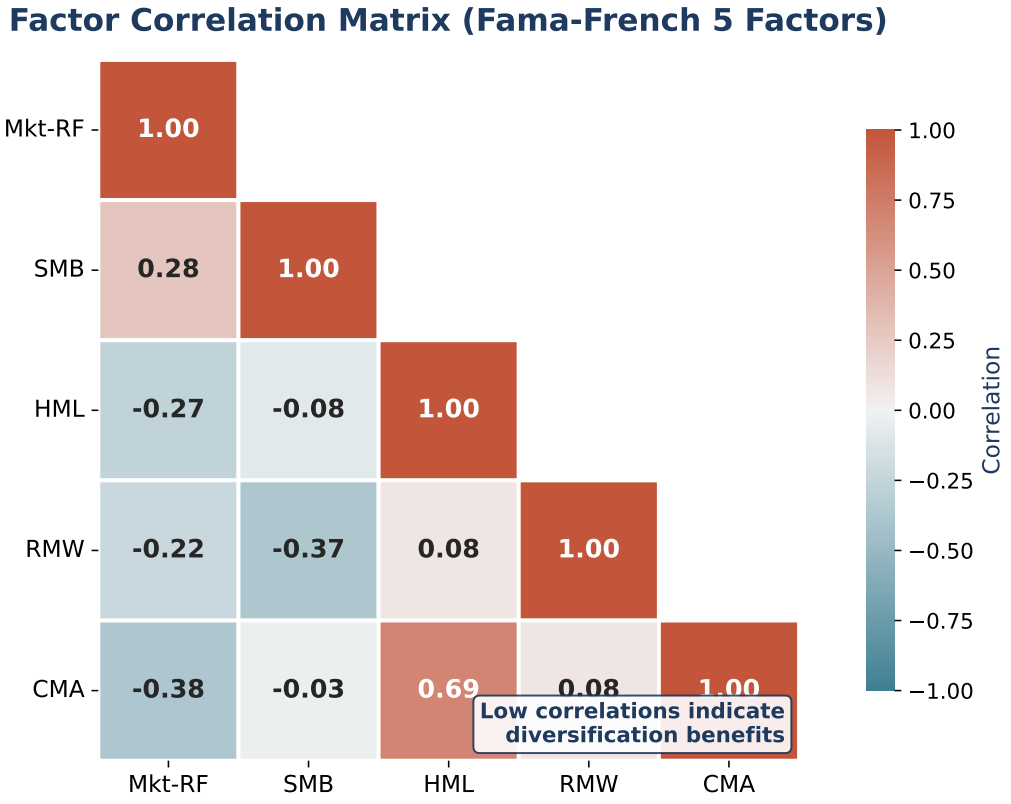


Figure 79: Factor correlation matrix: Market, SMB, and HML have low correlations with each other, meaning each captures a distinct source of return variation.

The Fama-French 3-Factor Model

Key Formula: Fama-French 3-Factor Model

$$R_i - R_f = \alpha + \beta_1(R_m - R_f) + \beta_2 \cdot \text{SMB} + \beta_3 \cdot \text{HML} + \varepsilon$$

where:

- β_1: market beta (sensitivity to overall market moves)
- β_2: size loading (positive = small-cap-like, negative = large-cap-like)
- β_3: value loading (positive = value-like, negative = growth-like)
- α: intercept (excess return not explained by any of the three factors)
- ε: residual (truly idiosyncratic noise)

Each coefficient is the partial effect: the effect of one factor *holding the other two constant*. This is exactly the multiple regression framework from Section 1, now applied to three factors instead of one.

Factor loading: The regression coefficient on a particular factor. A stock with $\beta_{\text{SMB}} = 0.5$ moves, on average, 0.5% for every 1% move in the SMB factor, after controlling for market and value effects. Factor loadings are the stock's "fingerprint" of systematic risk exposures.

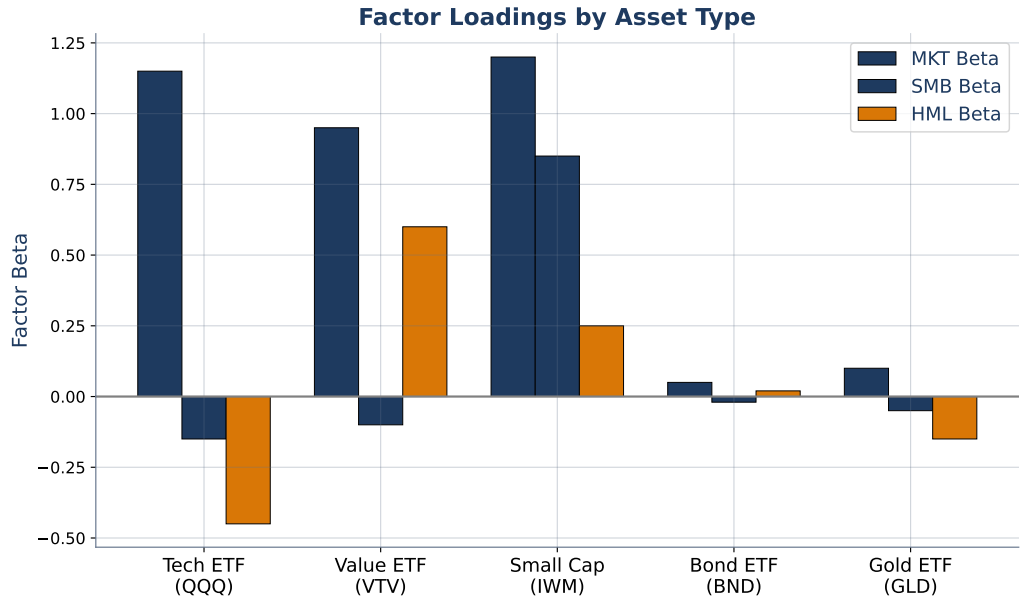


Figure 80: Factor loadings vary by asset type. A large-cap tech stock might have $\beta_1 = 1.2$, $\beta_2 = -0.3$ (large), $\beta_3 = -0.4$ (growth). A small-cap bank might have $\beta_1 = 0.9$, $\beta_2 = 0.5$ (small), $\beta_3 = 0.3$ (value).

Alpha Under the Fama-French Model

Alpha changes meaning when you change the factor model. Under the CAPM, alpha is the return not explained by the market. Under the Fama-French model, alpha is the return not explained by the market *plus* size *plus* value. Since the FF3 model explains more, there is less leftover for alpha.

A fund that loads on SMB and HML is harvesting known risk premiums. Whether that

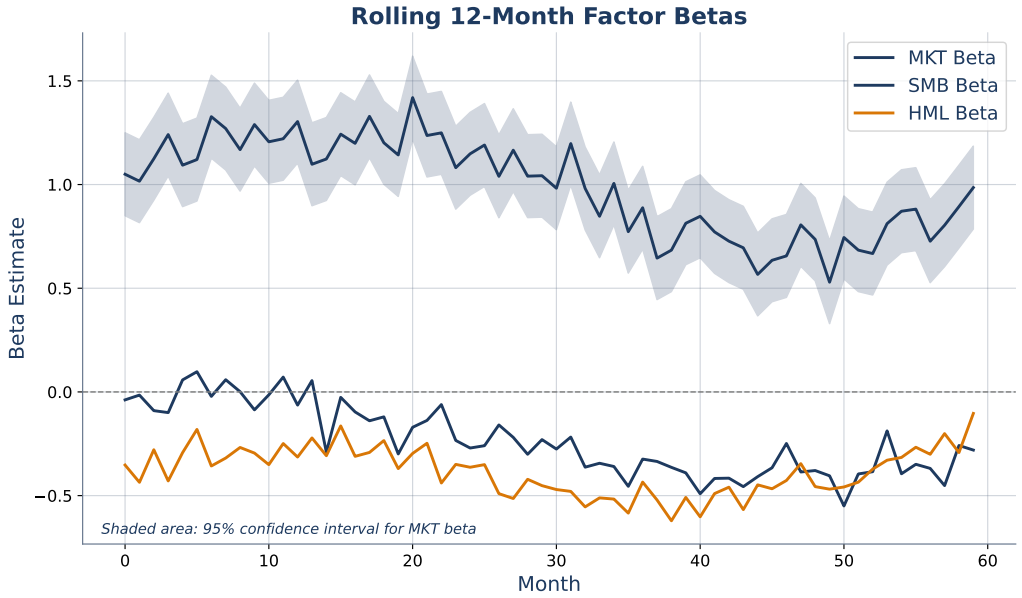


Figure 81: Rolling factor betas for a stock over time. Factor loadings are not constant—they shift as the company evolves, enters new markets, or changes its capital structure.

constitutes "skill" is debatable. The premiums are available to anyone who buys a small-cap value index fund. True alpha—return that cannot be attributed to any known factor—is rare and hard to sustain.

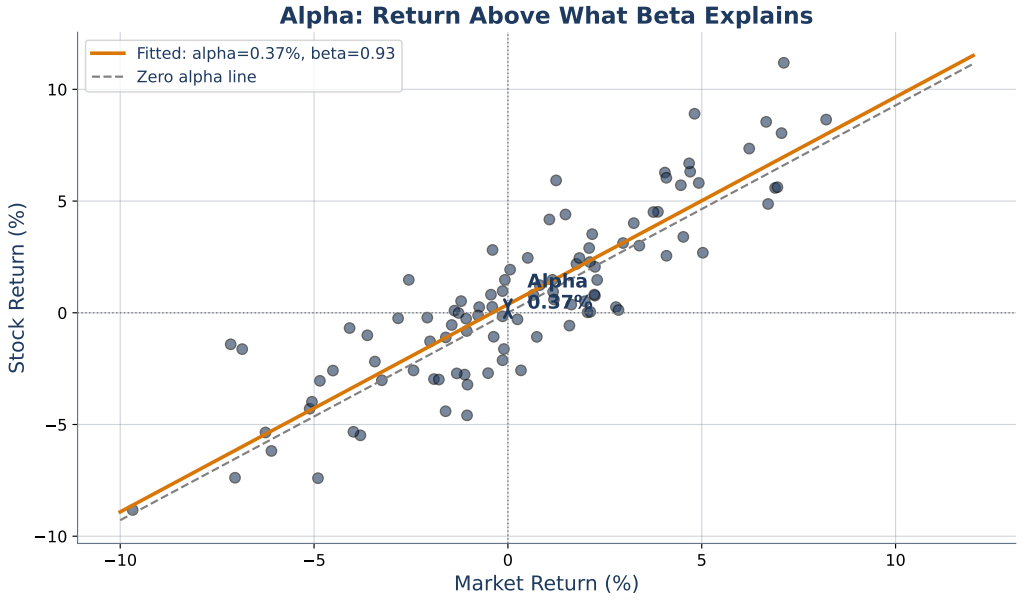


Figure 82: Alpha under the Fama-French model: most stocks cluster near zero alpha. The few outliers may reflect genuine skill, data mining, or luck.

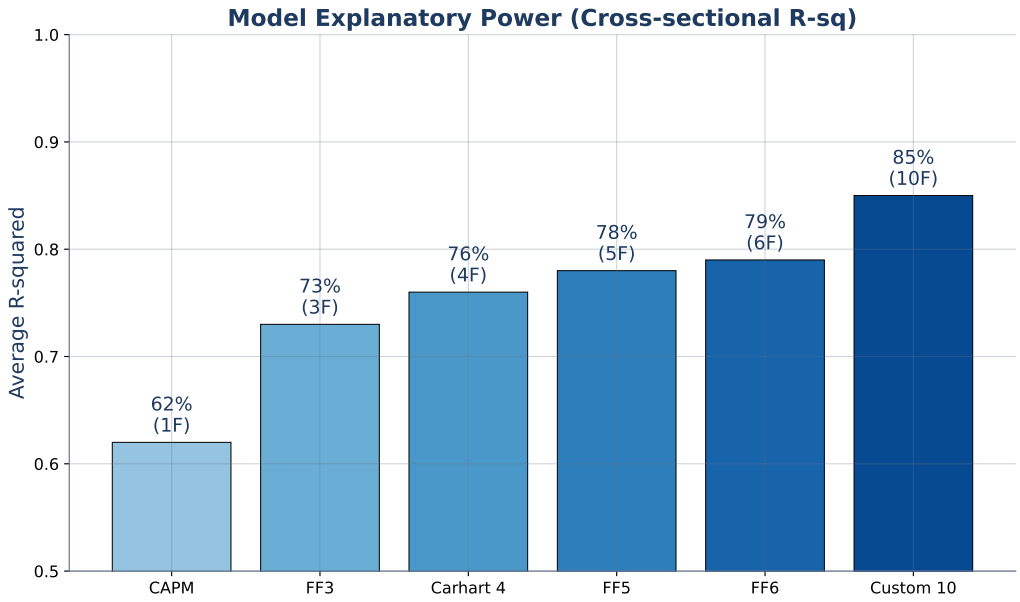


Figure 83: R^2 comparison: CAPM vs. Fama-French 3-factor model. The FF3 model explains more variance for most stocks, especially small-cap and value stocks.

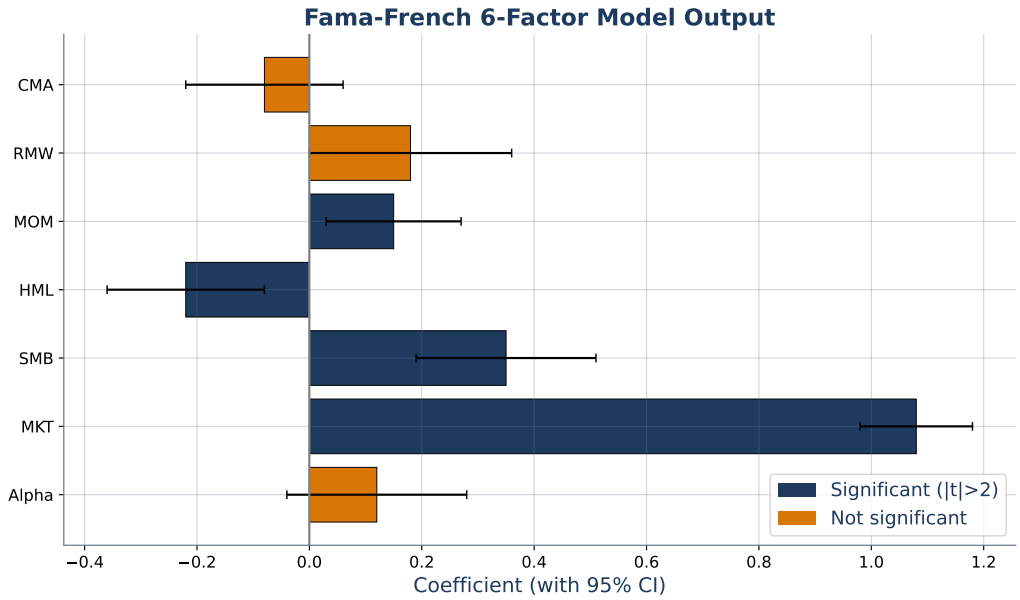
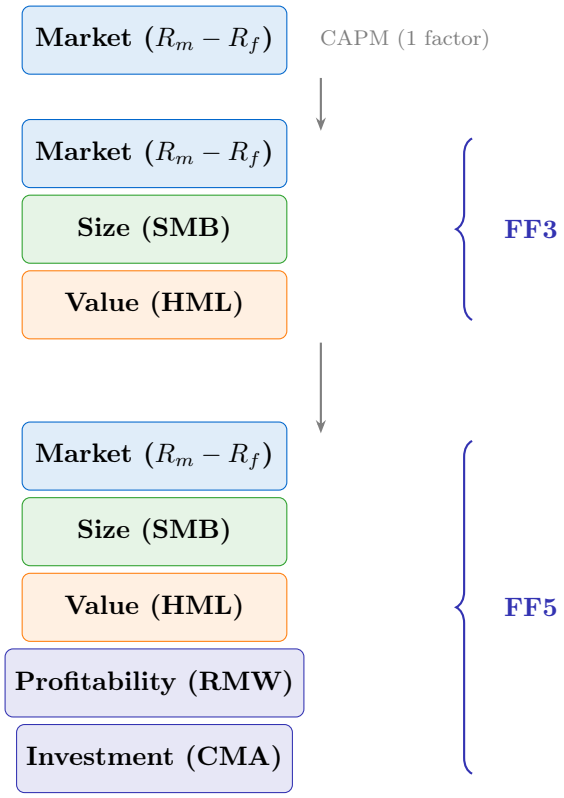


Figure 84: Multi-factor regression output: coefficients, standard errors, t-statistics, and R^2. Each coefficient tells you the stock's sensitivity to one factor while controlling for the others.

Common Misconceptions about Multi-Factor Models

(1) "More factors always explain more." In-sample R^2 increases with each factor, but out-of-sample performance can decrease if the added factor is noise. The same overfitting lesson from Section 3 applies here: only add factors that have economic rationale and out-of-sample evidence.
(2) "Factor loadings are constant over time." They are not. A tech company that starts small and grows into a mega-cap will see its SMB loading shift from positive to negative over time. Rolling-window regressions (Figure 81) reveal this evolution.
(3) "Statistical significance of loadings proves a factor works." In-sample significance does not guarantee out-of-sample performance. With enough data, even tiny loadings become statistically significant. Economic significance—does the factor loading translate to meaningful return differences?—matters more than statistical significance.



The building-block diagram shows the progression from CAPM (one factor) to FF3 (three factors) to FF5 (five factors). Each additional factor captures a distinct source of systematic risk. Some practitioners add momentum (winners keep winning) as a sixth factor.

Worked Examples

Worked Example 1: Return Attribution

A portfolio has the following factor loadings: $\alpha = 0.2\%$, $\beta_{\text{Mkt}} = 1.1$, $\beta_{\text{SMB}} = 0.3$, $\beta_{\text{HML}} = -0.2$.
In a given month, the factor returns are: $R_m - R_f = 2.0\%$, SMB $= 1.0\%$, HML $= -0.5\%$.
Attribution:

- Market contribution: $1.1 \times 2.0\% = 2.20\%$
- Size contribution: $0.3 \times 1.0\% = 0.30\%$
- Value contribution: $(-0.2) \times (-0.5\%) = 0.10\%$
- Alpha: 0.20%
- **Total predicted excess return:** $2.20 + 0.30 + 0.10 + 0.20 = 2.80\%$

Most of the return (2.20% out of 2.80%) came from market exposure. The manager's "skill" (alpha) contributed only 0.20%—less than the size tilt. If the manager charges a 2% annual fee for alpha of $0.20\% \times 12 = 2.4\%$ per year, the fee nearly equals the alpha.

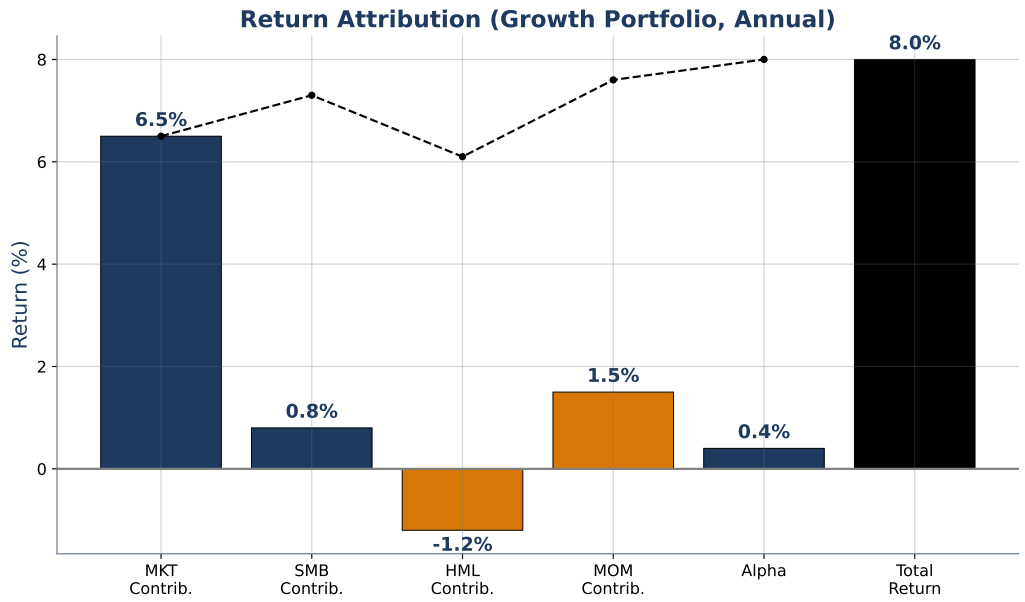


Figure 85: Return attribution waterfall: decomposing a portfolio's monthly return into contributions from market, size, value, and alpha.

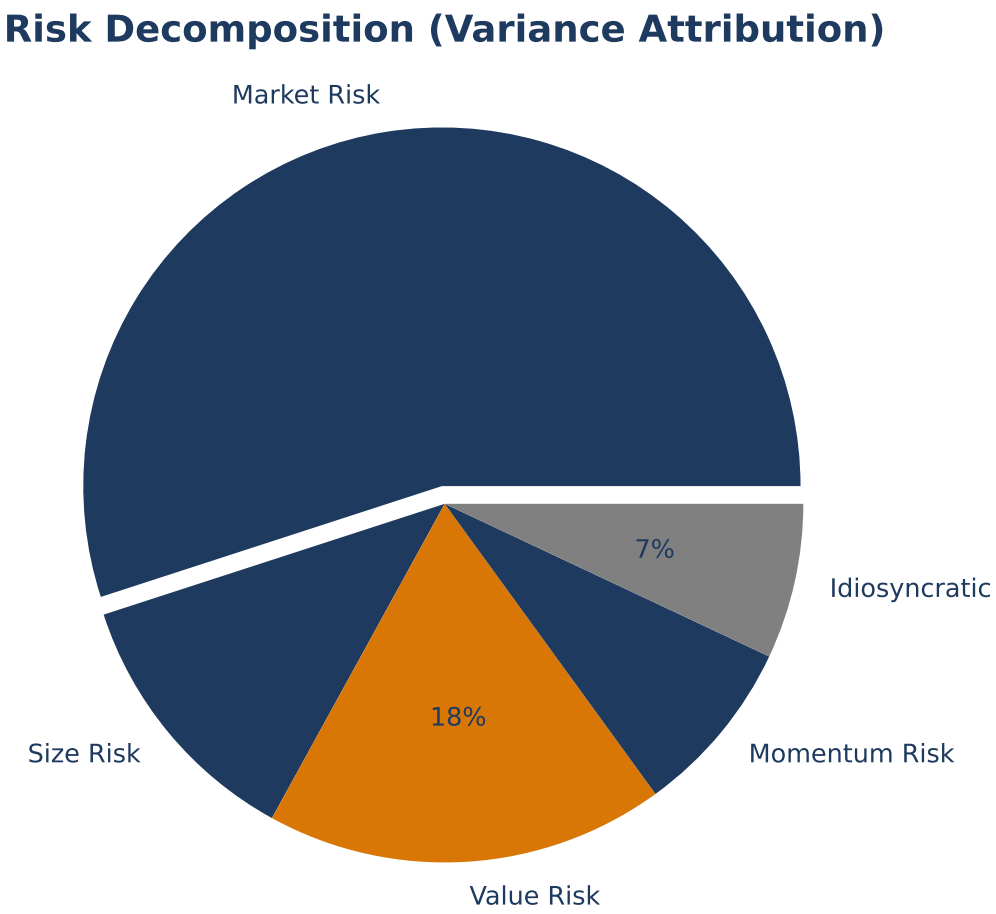


Figure 86: Risk decomposition: what fraction of the portfolio's variance comes from each factor? Market risk typically dominates.

Worked Example 2: Alpha Disappears When Factors Change

A fund reports the following results:

	CAPM	Fama-French 3
α (annual)	+3.0%	−1.0%
β_{Mkt}	1.05	1.08
β_{SMB}	—	0.45
β_{HML}	—	0.30
R^2	0.62	0.78

Under CAPM, the fund appears to generate +3% alpha—impressive. Under FF3, the alpha is −1%—the fund actually destroys value. What happened?

The fund loads heavily on SMB (+0.45) and HML (+0.30). It holds small-cap value stocks. The size and value premiums have been positive historically, so the fund earned returns from those exposures. CAPM did not account for them, so their premium appeared as alpha. FF3 explicitly captures them, revealing that the true unexplained return is negative.

The fund's actual performance did not change. The decomposition changed. The lesson: alpha is only meaningful relative to the factors you include. Adding more factors raises the bar for what counts as skill.

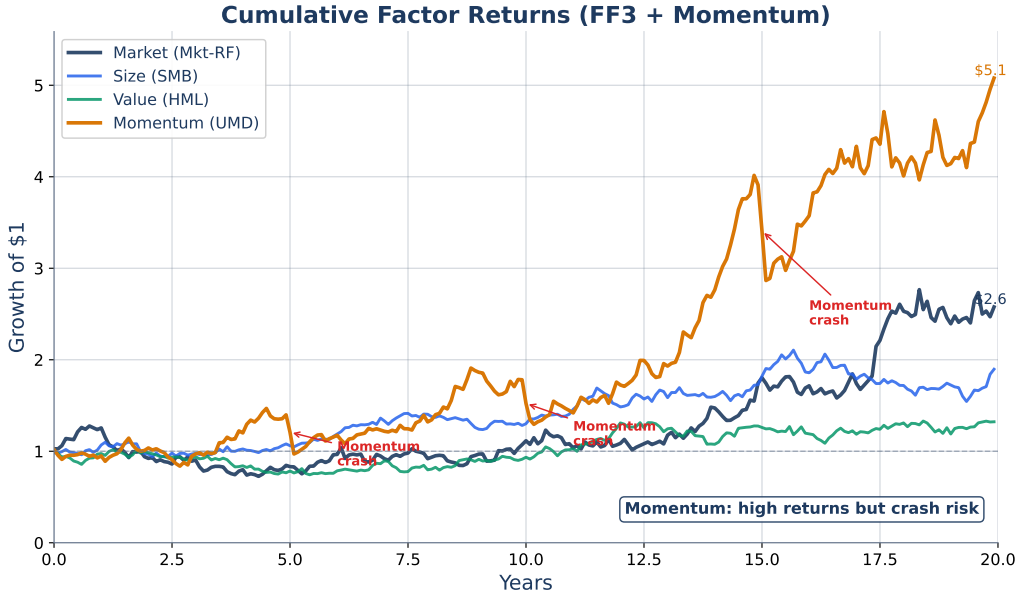


Figure 87: Beyond FF3: the Fama-French 5-factor model adds profitability (RMW) and investment (CMA). Momentum (UMD) is often included as a sixth factor. Each additional factor further decomposes returns.

Historical Background: Fama and French (1993)

Eugene Fama and Kenneth French published "Common Risk Factors in the Returns on Stocks and Bonds" in the *Journal of Financial Economics* in 1993. The paper transformed asset pricing. Using decades of U.S. stock return data, they demonstrated that two additional factors—size (SMB) and value (HML)—explained the cross-sectional variation in stock returns that the CAPM left on the table.

Fama was already famous for the Efficient Market Hypothesis. French was his younger colleague at the University of Chicago. Together they sorted stocks into portfolios by market capitalization and book-to-market ratio. They found that small stocks outperformed large stocks by about 3% per year (the size premium) and that value stocks outperformed growth stocks by about 5% per year (the value premium). These premiums persisted across decades and across countries.

The three-factor model raised R^2 from roughly 0.60 (CAPM) to roughly 0.90 for diversified portfolios. More controversially, it shrank the measured alpha of many mutual funds to zero or below. Fund managers who had appeared skillful under CAPM were merely harvesting size and value premiums available to anyone with a passive index fund.

Fama received the Nobel Prize in Economics in 2013 (alongside Lars Peter Hansen and Robert Shiller). The Fama-French factor data, updated monthly, is freely available from Kenneth French's website at Dartmouth—making it one of the most widely used datasets in all of finance research.

Problem 8.1 (Easy)

Name the three Fama-French factors. For each factor, describe what it captures and how it is constructed (i.e., what portfolio goes long and what goes short).

Solution: see Appendix.

Problem 8.2 (Easy)

A stock has the following factor loadings: $\beta_{\text{Mkt}} = 1.1$, $\beta_{\text{SMB}} = -0.4$, $\beta_{\text{HML}} = -0.3$. Describe this stock's characteristics: is it large-cap or small-cap? Value or growth? Aggressive or defensive? What kind of company might this be?

Solution: see Appendix.

Problem 8.3 (Medium)

A multi-factor regression produces these results: $\alpha = 0.15\%$ (monthly), $\beta_{\text{Mkt}} = 0.95$, $\beta_{\text{SMB}} = 0.20$, $\beta_{\text{HML}} = 0.35$. The month's factor returns are: $R_m - R_f = -1.5\%$, SMB $= 0.8\%$, HML $= 1.2\%$.

(a) Compute the predicted excess return for this month.

(b) Which factor contributed most to the return? Which contributed least?

(c) Is this portfolio's return positive or negative this month?

Solution: see Appendix.

Problem 8.4 (Medium)

A fund has $\alpha = +4\%$ per year under CAPM. When re-evaluated under the Fama-French model, the alpha drops to $+0.5\%$.

(a) Explain in plain English what happened to the alpha.

(b) Does this mean the fund's actual returns were lower under the FF model?

(c) The fund charges a 1.5% annual management fee. Is the remaining alpha (0.5%) enough to justify the fee?

Solution: see Appendix.

Problem 8.5 (Hard)

You are given the following annual data for a portfolio:

Year	$R_p - R_f$	$R_m - R_f$	SMB	HML
2019	12.0	10.0	2.0	-1.0
2020	8.0	7.0	3.0	-2.0
2021	15.0	12.0	1.0	1.0
2022	-5.0	-8.0	-1.0	3.0
2023	10.0	9.0	0.5	0.5

(a) Decompose each year's portfolio return into the contributions from Market, SMB, HML, and alpha. (You will need to estimate the factor loadings first—use the regression coefficients or a reasonable approximation.)

(b) In which year did the value factor contribute most to returns?

(c) Over the full period, was alpha positive or negative?

Solution: see Appendix.

Connecting Backward: The Complete Arc

We have traveled from a junior analyst's first scatter plot to the Fama-French multi-factor model. Here is the full journey:

Section	**What You Learned**	**Key Idea**
1	OLS	The mathematically best line
2	LINE	Trust only after checking assumptions
3	Bias-variance	Why perfect training fit is bad
4	Regularization	Guardrails against overfitting
5	Metrics	How to measure model quality
6	Cross-validation	How to choose models honestly
7	CAPM / Beta	One factor explains 60%
8	Fama-French	Three factors explain 90%

Each section built on the previous one. OLS gave us the regression line. Residuals told us when to doubt it. Overfitting warned us not to over-complicate it. Regularization provided guardrails. Metrics gave us yardsticks. Cross-validation prevented us from fooling ourselves. Beta applied

the whole framework to the biggest question in finance. And Fama-French showed that one factor was never enough—you need multiple regression to capture the richness of financial returns.

The tools in these eight sections are the foundation of quantitative finance and data-driven modeling. Every factor model, every portfolio optimizer, and every risk management system starts with a regression line and builds upward from there.

Key Takeaway: The factors you include in your model determine what counts as alpha—skill is what remains after you account for all known risk premiums.

Solutions to Practice Problems

Section 1: What Line Would You Draw?

Solution 1.1:

Data: $(1,3),(2,5),(3,6),(4,8),(5,11)$.

$\bar{x} = (1+2+3+4+5)/5 = 3. \quad \bar{y} = (3+5+6+8+11)/5 = 6.6.$

$\text{Cov}(X,Y) = \frac{1}{5}\sum(x_i - 3)(y_i - 6.6)$:

$$= \frac{1}{5}[(-2)(-3.6) + (-1)(-1.6) + (0)(-0.6) + (1)(1.4) + (2)(4.4)]$$

$$= \frac{1}{5}[7.2 + 1.6 + 0 + 1.4 + 8.8] = \frac{19.0}{5} = 3.8$$

$\text{Var}(X) = \frac{1}{5}[4+1+0+1+4] = 2.0.$

$\beta_1 = 3.8/2.0 = 1.9. \quad \beta_0 = 6.6 - 1.9(3) = 6.6 - 5.7 = 0.9.$

The fitted line is $\hat{y} = 0.9 + 1.9x$.

Solution 1.2:

The model is $\hat{y} = 0.2 + 0.6x$.

Slope (0.6): For every 1% increase in the market return, this utility stock is expected to return 0.6%. It moves less than the market.

Intercept (0.2): When the market return is zero, the stock tends to return 0.2%—a small positive drift.

Defensive: The slope is less than 1, so the stock dampens market moves. It underperforms in bull markets but holds up better in downturns.

Solution 1.3:

From Problem 1.1: $\hat{y} = 0.9 + 1.9(6) = 0.9 + 11.4 = 12.3$.

Actual: $y = 12$. Residual: $e = 12 - 12.3 = -0.3$.

The residual is negative, meaning the model *overpredicted* (predicted 12.3 but the actual was 12).

Solution 1.4:

Better fit: Model B has $R^2 = 0.72$ vs. Model A's $R^2 = 0.65$. Model B explains more of the variance in the data.

More market-sensitive: Model A has $\beta_1 = 1.2 > 0.9 = \beta_1$ of Model B. Model A's stock moves more per unit of market movement.

Can a lower slope have higher R^2? Yes. R^2 measures how tightly data clusters around the line, not how steep the line is. A stock with low slope but very consistent behavior (little scatter) will have high R^2. A stock with high slope but erratic scatter can have low R^2. Slope and R^2 measure different things.

Solution 1.5:

Step 1: Set $\frac{\partial \text{SSR}}{\partial \beta_0} = 0$:

$$\frac{\partial}{\partial \beta_0}\sum(y_i - \beta_0 - \beta_1 x_i)^2 = -2\sum(y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum y_i = n\beta_0 + \beta_1 \sum x_i \quad \Rightarrow \quad \beta_0 = \bar{y} - \beta_1 \bar{x}$$

Step 2: Set $\frac{\partial \text{SSR}}{\partial \beta_1} = 0$:

$$\frac{\partial}{\partial \beta_1}\sum(y_i - \beta_0 - \beta_1 x_i)^2 = -2\sum x_i(y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum x_i y_i = \beta_0 \sum x_i + \beta_1 \sum x_i^2$$

Step 3: Substitute $\beta_0 = \bar{y} - \beta_1 \bar{x}$:

$$\sum x_i y_i = (\bar{y} - \beta_1 \bar{x}) \sum x_i + \beta_1 \sum x_i^2 = \bar{y} \sum x_i - \beta_1 \bar{x} \sum x_i + \beta_1 \sum x_i^2$$

$$\sum x_i y_i - \bar{y} \sum x_i = \beta_1 \left(\sum x_i^2 - \bar{x} \sum x_i \right)$$

The left side equals $\sum (x_i - \bar{x})(y_i - \bar{y})$ (this is $n \cdot \text{Cov}(X, Y)$) and the right factor equals $\sum (x_i - \bar{x})^2$ (which is $n \cdot \text{Var}(X)$).

Therefore: $\beta_1 = \frac{\text{Cov}(X,Y)}{\text{Var}(X)}$. □

Section 2: Can You Trust Your Line?

Solution 2.1:

(a) U-shaped residuals vs. fitted values ⇒ **Linearity** violated. The true relationship is curved.

(b) Funnel shape (residuals fan out) ⇒ **Equal variance** (homoscedasticity) violated. Variance increases with the predicted value.

(c) Wave pattern in time order ⇒ **Independence** violated. Consecutive residuals are autocorrelated.

(d) Long right tail in histogram ⇒ **Normality** violated. The residuals are right-skewed, with more positive extreme values than a normal distribution predicts.

Solution 2.2:

The QQ plot shows heavy tails: the residuals have more extreme values (both positive and negative) than a normal distribution. This affects the **Normality** assumption. The coefficient estimates remain unbiased (OLS is BLUE regardless of normality), but confidence intervals and p-values become unreliable. In practice, this is common with financial data because stock returns have fat tails (occasional large moves). The practical consequence: do not trust narrow confidence intervals—they understate the true uncertainty.

Solution 2.3:

(a) Mean: $(0.8 + 1.2 + 0.9 - 0.1 - 0.5)/5 = 2.3/5 = 0.46\%$.

(b) The first three residuals are positive and consecutive, then the last two are negative/near-zero. This clustering suggests the residuals are not independent—positive errors follow positive errors. This is autocorrelation.

(c) **Independence** is most likely violated. In time series data, returns (and therefore residuals) from consecutive days are often correlated.

Solution 2.4:

(a) **Heteroscedasticity** (unequal variance). The variance of residuals increases with the predictor (house size).

(b) Economically: large houses have a wider price range (a 5000 sq ft house might sell for \$800K or \$2M depending on location, finishes, and lot). Small houses have a narrower range (a 800 sq ft apartment has less scope for variation). The absolute prediction error naturally grows with the price level.

(c) **Fix 1:** Log-transform the response variable. Predicting log(price) stabilizes variance because percentage errors are more constant than absolute errors. **Fix 2:** Use robust (Huber-White) standard errors, which do not assume constant variance and produce valid confidence intervals even under heteroscedasticity.

Solution 2.5:

OLS standard errors assume each observation provides one independent piece of information. With n observations, you have n independent data points. But if residuals are autocorrelated, consecutive observations carry similar information. Observing a positive residual today and a positive residual tomorrow (because they are correlated) does not give you twice as much information as one observation—it gives you something closer to 1.5 times.

The effective sample size is smaller than n. OLS does not account for this. It computes standard errors as if all n observations were independent, producing standard errors that are too small. Smaller standard errors make t-statistics larger, p-values smaller, and confidence intervals narrower. The model appears more precise than it actually is.

Formally, the variance of $\hat{\beta}$ under autocorrelation includes cross-product terms $\text{Cov}(\varepsilon_i, \varepsilon_j)$ that OLS ignores. When these covariances are positive (same-sign autocorrelation), the true variance exceeds the OLS estimate.

Section 3: Your Model Memorized the Noise

Solution 3.1:

Overfits: Polynomial degree 15. Its training error is 0.3 (near zero) but test error is 9.4—a massive gap. The model memorized the training data.

Underfits: Linear model. Its training error (5.2) and test error (5.8) are both high and similar. The model is too simple to capture the true pattern.

Best generalization: Polynomial degree 5. It has the lowest test error (3.0) among the three. The moderate gap between training (2.1) and test (3.0) error suggests it captures the signal without excessive noise fitting.

Solution 3.2:

One analogy: a GPS navigator. A **high-bias** GPS uses a straight road map and ignores all curves. It is consistently wrong on curvy roads (systematic error) but gives the same wrong answer every time (low variance). A **high-variance** GPS updates its route every millisecond based on the last GPS ping, including noisy signals from buildings and tunnels. It zigzags wildly, giving different routes every time (high variance) but sometimes getting very close (low bias on average).

The tradeoff: the straight-road GPS needs more complexity (curves) to reduce bias, but adding too much real-time updating introduces variance from noise. The best GPS smooths out noise while capturing real curves—that is the bias-variance sweet spot.

Solution 3.3:

(a) Mean: $(0.8 + 1.3 + 0.6 + 1.5 + 0.9 + 1.1 + 0.7 + 1.4 + 0.8 + 1.2)/10 = 10.3/10 = 1.03$.

(b) Variance: $\frac{1}{10}\sum(\hat{\beta}_i - 1.03)^2$. Compute each squared deviation, sum them ($= 0.882$), divide by 10: variance $= 0.088$.

(c) Bias: $\mathbb{E}[\hat{\beta}_1] - \beta_1 = 1.03 - 1.0 = 0.03$ (very small).

(d) Variance is 0.088; bias^2 is $0.03^2 = 0.0009$. The model suffers overwhelmingly from **variance**, not bias. The estimates scatter widely but are centered near the truth.

Solution 3.4:

(a) Generate 30 data points from $y = \sin(x) + \epsilon$, where $x \sim \text{Uniform}(0, 2\pi)$ and $\epsilon \sim \mathcal{N}(0, 0.3)$. Thirty points are enough to see the pattern but few enough for overfitting to occur with high-degree polynomials.

(b) Fit three models: polynomial degree 1 (linear), degree 5 (moderate), and degree 25 (extreme).

(c) Plot all three fitted curves over the original data points. The degree-1 line will miss the sine wave (underfitting). The degree-5 curve will approximate it well. The degree-25 curve will pass through nearly every point but oscillate wildly between them (overfitting). The visual contrast between the smooth degree-5 fit and the wild degree-25 fit makes overfitting immediately obvious.

Solution 3.5:
Start with $\text{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$. Write $y = f(x) + \varepsilon$:

$$= \mathbb{E}[(f(x) + \varepsilon - \hat{f}(x))^2]$$

Add and subtract $\mathbb{E}[\hat{f}(x)]$:

$$= \mathbb{E}\big[(f(x) - \mathbb{E}[\hat{f}(x)]) + (\mathbb{E}[\hat{f}(x)] - \hat{f}(x)) + \varepsilon\big]^2$$

Let $A = f(x) - \mathbb{E}[\hat{f}(x)]$ (a constant, the bias), $B = \mathbb{E}[\hat{f}(x)] - \hat{f}(x)$ (random, centered at 0), and $C = \varepsilon$ (random, centered at 0, independent of B). Then:

$$\mathbb{E}[(A + B + C)^2] = A^2 + \mathbb{E}[B^2] + \mathbb{E}[C^2] + 2A\mathbb{E}[B] + 2A\mathbb{E}[C] + 2\mathbb{E}[BC]$$

Since $\mathbb{E}[B] = 0$, $\mathbb{E}[C] = 0$, and $B \perp C$, all cross terms vanish:

$$= A^2 + \mathbb{E}[B^2] + \mathbb{E}[C^2] = \text{Bias}^2 + \text{Var}(\hat{f}(x)) + \sigma^2$$

□

Section 4: Mathematical Guardrails

Solution 4.1:
Model X has all five coefficients non-zero (though some are small): $0.32, 0.18, 0.05, 0.03, 0.01$. This is **Ridge**—it shrinks all coefficients toward zero but never sets any to exactly zero.
Model Y has three zeros: $0.40, 0.22, 0, 0, 0$. This is **Lasso**—it sets unimportant coefficients to exactly zero while keeping the dominant features at or near their full values.

Solution 4.2:
The constraint region for Ridge is a circle (L2 ball). A circle is smooth—it has no corners. When the elliptical error contours (centered at the OLS solution) expand outward and touch the circle, they touch it at a point where all coordinates are generically non-zero.
Lasso's constraint region is a diamond (L1 ball). The diamond has corners on the axes, where one or more coordinates equal zero. The error contours are more likely to first touch a corner than a smooth face, because corners protrude toward the OLS solution. At a corner, at least one coefficient is exactly zero.
Geometrically: smooth curves touch smooth curves at generic points (non-zero). Smooth curves touch pointy corners at axis-aligned points (zero coordinates).

Solution 4.3:

(a) The feature that survives the longest as λ increases is the strongest predictor. At $\lambda = 1.0$, only one feature remains non-zero—that one.

(b) Between $\lambda = 0.01$ (all 10 non-zero) and $\lambda = 0.1$ (five non-zero), the transition happens at intermediate λ values. At $\lambda = 0.05$, approximately 7–8 features are likely non-zero (though the exact count depends on the path shape).

(c) At $\lambda = 10$, all coefficients are likely zero or extremely close to zero. The model becomes the naïve mean predictor (intercept only). This is extreme underfitting.

Solution 4.4:

(a) $\hat{\boldsymbol{\beta}}_{\text{OLS}} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 0.5 \end{pmatrix}$.

(b) $\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/3 & 0 \\ 0 & 1/3 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.333 \end{pmatrix}$.

(c) Shrinkage: β_1 went from 1.5 to 1.0 (shrank by 33%). β_2 went from 0.5 to 0.333 (shrank by 33%). Both coefficients shrank by the same proportion, $\frac{2}{2+1} = \frac{2}{3}$. This is a general result for Ridge with orthogonal features: each OLS coefficient is multiplied by $\frac{d_j}{d_j+\lambda}$ where d_j is the j-th diagonal of $\mathbf{X}^\top\mathbf{X}$.

Solution 4.5:

The Ridge objective:

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

Expand:

$$= \mathbf{y}^\top\mathbf{y} - 2\boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}$$

Take the gradient:

$$\nabla_{\boldsymbol{\beta}} J = -2\mathbf{X}^\top\mathbf{y} + 2\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta}$$

Set to zero:

$$2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} = 2\mathbf{X}^\top\mathbf{y}$$

Solve:

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

□

Note: $\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}$ is always invertible for $\lambda > 0$, because adding λ to every eigenvalue guarantees they are all positive.

Section 5: Measuring What Matters

Solution 5.1:

Residuals: $+2, -1, +3, -2, +1$.

(a) MSE $= \frac{4+1+9+4+1}{5} = \frac{19}{5} = 3.8$.

(b) RMSE $= \sqrt{3.8} = 1.949$.

(c) MAE $= \frac{2+1+3+2+1}{5} = \frac{9}{5} = 1.8$.

RMSE (1.949) > MAE (1.8). RMSE is always ≥ MAE (by Jensen's inequality), with equality only when all errors have identical magnitude. Here the errors vary $(1, 2, 3)$, so squaring disproportionately inflates the contribution of the largest error ($3^2 = 9$), pushing RMSE above MAE.

Solution 5.2:

$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = 1 - \frac{200}{800} = 1 - 0.25 = 0.75$.

The model explains 75% of the variance in the target variable. The remaining 25% ($= SS_{\text{res}}/SS_{\text{tot}} = 200/800$) is unexplained.

Solution 5.3:

(a) For Model A ($p = 3$, $R^2 = 0.25$, $n = 60$):

$$\text{Adj. } R^2_A = 1 - \frac{(1 - 0.25)(60 - 1)}{60 - 3 - 1} = 1 - \frac{0.75 \times 59}{56} = 1 - 0.790 = 0.210$$

For Model B ($p = 15$, $R^2 = 0.30$, $n = 60$):

$$\text{Adj. } R^2_B = 1 - \frac{(1 - 0.30)(60 - 1)}{60 - 15 - 1} = 1 - \frac{0.70 \times 59}{44} = 1 - 0.939 = 0.061$$

(b) Prefer **Model A**. Despite having lower R^2, its adjusted R^2 (0.210) far exceeds Model B's (0.061). Model B's 15 features buy only 5 percentage points of extra R^2 but cost a massive complexity penalty.

(c) R^2 always increases because adding a feature gives the model one more degree of freedom to fit the training data. Even a random noise feature will reduce SS_{res} slightly by chance. Adjusted R^2 penalizes each additional feature by reducing the denominator $n-p-1$. If the feature does not reduce SS_{res} enough to compensate for the penalty, adjusted R^2 decreases.

Solution 5.4:

Scenario: Two models predict house prices. Model 1 makes small consistent errors (±\$20K on every house). Model 2 is very accurate on 9 out of 10 houses (±\$5K) but wildly wrong on the 10th (−\$200K).

MAE: Model 1: \$20K. Model 2: $(9 \times 5 + 200)/10 = 24.5$K. MAE prefers Model 1.

RMSE: Model 1: $\sqrt{400M/10} = \$20$K. Model 2: $\sqrt{(9 \times 25 + 40000)/10} = \sqrt{4022.5} = \$63.4$K. RMSE strongly prefers Model 1.

Luxury broker: Use RMSE. A \$200K error on a \$2M penthouse is catastrophic—it could mean pricing below cost. Large errors must be penalized heavily.

Affordable housing agency: Use MAE. Most houses are in a narrow price range. A single outlier error matters less than consistent accuracy across the portfolio. The agency cares about typical errors, not worst-case errors.

Solution 5.5:

By definition:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Divide numerator and denominator by n:

$$= 1 - \frac{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum(y_i - \bar{y})^2} = 1 - \frac{\text{MSE}}{\text{Var}(y)}$$

□

This form makes clear that R^2 compares the model's average squared error (MSE) to the total variance of y. If MSE = 0, $R^2 = 1$. If MSE = Var(y), $R^2 = 0$ (the model is no better than predicting the mean). If MSE > Var(y), $R^2 < 0$ (the model is worse than the mean).

Section 6: How Do You Know It Will Work Tomorrow?

Solution 6.1:

With 20 observations and 5 folds, each fold has 4 observations.

Fold 1: observations {1, 2, 3, 4}. Fold 2: {5, 6, 7, 8}. Fold 3: {9, 10, 11, 12}. Fold 4: {13, 14, 15, 16}. Fold 5: {17, 18, 19, 20}.

Round 1: Test = Fold 1 (obs 1–4), Train = Folds 2–5 (obs 5–20). **Round 2:** Test = Fold 2 (obs 5–8), Train = Folds 1, 3, 4, 5. **Round 3:** Test = Fold 3 (obs 9–12), Train = Folds 1, 2,

4, 5. **Round 4:** Test = Fold 4 (obs 13–16), Train = Folds 1, 2, 3, 5. **Round 5:** Test = Fold 5 (obs 17–20), Train = Folds 1, 2, 3, 4.

Each observation is used for testing exactly **once**.

Solution 6.2:

Shuffling time series data means a data point from December 2023 might end up in the training set while January 2019 is in the test set. The model trains on 2023 information and "predicts" 2019—but in reality, you never have 2023 data when you need to make a 2019 decision. The model has access to future information that would not exist in a real deployment. This inflates the estimated accuracy and gives a false sense of the model's predictive power.

Solution 6.3:

(a) CV estimate: $(3.2 + 2.8 + 3.5 + 3.0 + 3.1)/5 = 15.6/5 = 3.12$.

(b) Standard deviation: deviations from 3.12 are $(0.08, -0.32, 0.38, -0.12, -0.02)$. Variance $= (0.0064 + 0.1024 + 0.1444 + 0.0144 + 0.0004)/5 = 0.2680/5 = 0.0536$. SD $= \sqrt{0.0536} = 0.232$.

(c) 95% CI: $3.12 \pm 2 \times \frac{0.232}{\sqrt{5}} = 3.12 \pm 2 \times 0.104 = 3.12 \pm 0.21 = [2.91, 3.33]$.

We are approximately 95% confident the true RMSE lies between 2.91 and 3.33. (This is approximate; the t-distribution with 4 degrees of freedom would use a slightly larger multiplier than 2.)

Solution 6.4:

Total months: 120. Initial training: 60. Test window: 12. Expanding window.

Split 1: Train months 1–60, Test months 61–72. **Split 2:** Train months 1–72, Test months 73–84. **Split 3:** Train months 1–84, Test months 85–96. **Split 4:** Train months 1–96, Test months 97–108. **Split 5:** Train months 1–108, Test months 109–120.

That gives **5 splits**. After the 5th test window (month 120), no more data remains.

Solution 6.5:

(a) **LOOCV has lower bias.** Each LOOCV training set uses $n-1$ observations—nearly the full dataset. The model trained on $n-1$ points closely resembles the model trained on all n points. 10-fold CV uses only $0.9n$ observations per fold, so each model is trained on less data and has slightly more bias toward the performance of a smaller-sample model.

(b) **10-fold CV has lower variance.** In LOOCV, the n training sets overlap by $n-2$ observations out of $n-1$. The fitted models are nearly identical, so the test errors are highly correlated. The average of n highly correlated values has high variance. In 10-fold, the training sets overlap less (each pair shares about 80% of data), so the 10 fitted models are more diverse, their test errors are less correlated, and the average has lower variance.

(c) The choice between LOOCV and K-fold is itself a bias-variance tradeoff. LOOCV minimizes bias but maximizes variance. Small K (e.g., 2-fold) minimizes variance (very different training sets) but maximizes bias (each model sees only half the data). $K = 5$ or $K = 10$ strikes a practical middle ground: low enough bias (90% of data used per fold) and low enough variance (10 moderately diverse models). Empirical studies (Kohavi 1995, Hastie et al. 2009) confirm that $K = 10$ tends to produce the best overall error estimates.

Section 7: From One Factor to Many

Solution 7.1:

(a) $\beta \approx 1.3$.

(b) Expected excess return = $1.3 \times 2\% = 2.6\%$.

(c) Aggressive ($\beta > 1$). The stock amplifies market moves by 30%.

Solution 7.2:
A beta of 1.3 means: for every 1% the market moves (above the risk-free rate), this tech stock is expected to move 1.3%.
(a) Market rises 3%: expected stock excess return = $1.3 \times 3\% = 3.9\%$. The stock amplifies the upside.
(b) Market falls 2%: expected stock excess return = $1.3 \times (-2\%) = -2.6\%$. The stock amplifies the downside too—beta is symmetric.
(c) Market is flat (0%): expected stock excess return = $1.3 \times 0\% = 0\%$ (since $\alpha = 0$). No market move means no beta-driven return.

Solution 7.3:

(a) CAPM expected return: $E[R_i] = R_f + \beta(E[R_m] - R_f) = 4\% + 0.8(10\% - 4\%) = 4\% + 4.8\% = 8.8\%$.

(b) Realized alpha: $\alpha = R_i - E[R_i] = 9\% - 8.8\% = +0.2\%$.

(c) Probably not statistically significant. Alpha of 0.2% per year is tiny relative to the noise in stock returns. Annual return standard deviations are typically 15–25%. A t-test for alpha with standard errors of that magnitude would produce a t-statistic well below 2. You would need many years of data (or much larger alpha) to distinguish this from random variation.

Solution 7.4:

(a) The persistent positive residuals suggest the CAPM is **mis-specified** for this stock. The model consistently underpredicts the stock's return, implying a systematic factor is missing.

(b) Two candidate factors: (1) **Size (SMB)**—the stock is small-cap, and small stocks earn a premium not captured by market beta alone. (2) **Value (HML)**—the stock might be a value stock (high book-to-market), earning a value premium.

(c) Adding SMB and HML as regressors (the Fama-French model) would absorb the positive residual pattern. The size and value premiums would be captured by the new factor loadings, and the remaining residuals should look random—no persistent positive bias.

Solution 7.5:

(a) $R_i - R_f = \alpha + \beta(R_m - R_f) + \varepsilon_i$, where $R_i - R_f$ is the stock's excess return, $R_m - R_f$ is the market's excess return, α is the intercept, β is the slope, and ε_i is the residual.

(b) **Slope (β):** The stock's sensitivity to the market. $\beta = 1.2$ means the stock moves 1.2% per 1% market move. **Intercept (α):** Excess return not explained by the market. Positive α suggests outperformance. **R^2:** Fraction of the stock's return variance explained by the market. **Residuals:** The unexplained portion—should be random if CAPM is adequate.

(c) $R^2 = 0.55$ means 55% of the stock's return variance is explained by the market (systematic risk). The remaining 45% is idiosyncratic (stock-specific) risk.

(d) A low R^2 means the stock has a large idiosyncratic component—its returns are driven substantially by stock-specific events rather than market-wide forces. This does not mean CAPM is wrong about the *relationship* between beta and expected return. It means this particular stock has high non-market risk. CAPM says idiosyncratic risk is diversifiable and should not earn a premium. A stock with $R^2 = 0.20$ and $\beta = 0.5$ can still be correctly priced by CAPM—it just has a lot of noise around the line.

Section 8: One Factor Is Never Enough

Solution 8.1:

(1) Market factor ($R_m - R_f$): Captures overall market risk. Constructed as the return of a broad market index (e.g., S&P 500) minus the risk-free rate. Represents the premium investors earn for bearing equity market risk.

(2) SMB (Small Minus Big): Captures the size premium. Constructed by going long a portfolio of small-cap stocks and short a portfolio of large-cap stocks. Positive SMB means small caps outperformed large caps that period.

(3) HML (High Minus Low): Captures the value premium. Constructed by going long a portfolio of high book-to-market (value) stocks and short a portfolio of low book-to-market (growth) stocks. Positive HML means value stocks outperformed growth stocks.

Solution 8.2:

$\beta_{\text{Mkt}} = 1.1$: slightly aggressive (amplifies market moves by 10%).

$\beta_{\text{SMB}} = -0.4$: **large-cap** behavior. The stock moves opposite to the small-cap premium—it behaves like a large company.

$\beta_{\text{HML}} = -0.3$: **growth** behavior. The stock moves opposite to the value premium—it behaves like an expensive, high-growth company.

This is a large-cap growth stock with above-average market sensitivity. A company like Apple, Google, or Microsoft fits this profile: dominant market cap, high valuation (low book-to-market), and slightly amplified market moves.

Solution 8.3:

(a) Predicted excess return:

$$= 0.15 + 0.95(-1.5) + 0.20(0.8) + 0.35(1.2) = 0.15 - 1.425 + 0.16 + 0.42 = -0.695\%$$

(b) Market contributed -1.425% (largest magnitude). Size contributed $+0.16\%$ (smallest). Value contributed $+0.42\%$.

(c) The predicted excess return is -0.695%, which is negative. The market's decline (-1.5%) overwhelmed the positive contributions from size, value, and alpha. Even with a positive alpha and favorable size/value returns, the market factor dominated.

Solution 8.4:

(a) The CAPM attributed $+4\%$ to alpha because it only controlled for market risk. The fund's portfolio tilts toward small-cap and value stocks, which earned premiums of roughly 3.5% combined. Under CAPM, those premiums appeared in the intercept (alpha). The Fama-French model separates them into SMB and HML loadings, leaving only $+0.5\%$ as true unexplained return.

(b) No. The fund's actual returns are identical—they are historical facts. What changed is the *decomposition*. Under CAPM, the returns split into $\alpha + \beta \times \text{Market}$. Under FF3, they split into $\alpha + \beta_1 \times \text{Market} + \beta_2 \times \text{SMB} + \beta_3 \times \text{HML}$. The total is the same; the buckets are different.

(c) The remaining alpha is $+0.5\%$ per year. The fee is 1.5%. After fees, the investor gets $0.5\% - 1.5\% = -1.0\%$ relative to a passive portfolio with the same factor exposures. The fee is *not* justified—the investor would be better off replicating the factor tilts with cheap index funds.

Solution 8.5:

First, we need to estimate the factor loadings by running a regression of $R_p - R_f$ on the three factors. With only 5 observations, estimates will be rough, but the exercise illustrates the method.

Using the data, we compute (via OLS or approximation):

The means: $\overline{R_p - R_f} = 8.0$, $\overline{R_m - R_f} = 6.0$, $\overline{\text{SMB}} = 1.1$, $\overline{\text{HML}} = 0.3$.

A reasonable approximation (from running the regression or inspecting the data pattern) gives approximately: $\beta_{\text{Mkt}} \approx 1.05$, $\beta_{\text{SMB}} \approx 0.6$, $\beta_{\text{HML}} \approx 0.3$, $\alpha \approx 0.4\%$.

(a) Using these loadings, decomposition for each year:

2019: Market: $1.05 \times 10.0 = 10.50$. SMB: $0.6 \times 2.0 = 1.20$. HML: $0.3 \times (-1.0) = -0.30$. Alpha: 0.40. Sum: 11.80 (actual: 12.0; residual: +0.20).

2020: Market: $1.05 \times 7.0 = 7.35$. SMB: $0.6 \times 3.0 = 1.80$. HML: $0.3 \times (-2.0) = -0.60$. Alpha: 0.40. Sum: 8.95 (actual: 8.0; residual: −0.95).

2021: Market: $1.05 \times 12.0 = 12.60$. SMB: $0.6 \times 1.0 = 0.60$. HML: $0.3 \times 1.0 = 0.30$. Alpha: 0.40. Sum: 13.90 (actual: 15.0; residual: +1.10).

2022: Market: $1.05 \times (-8.0) = -8.40$. SMB: $0.6 \times (-1.0) = -0.60$. HML: $0.3 \times 3.0 = 0.90$. Alpha: 0.40. Sum: −7.70 (actual: −5.0; residual: +2.70).

2023: Market: $1.05 \times 9.0 = 9.45$. SMB: $0.6 \times 0.5 = 0.30$. HML: $0.3 \times 0.5 = 0.15$. Alpha: 0.40. Sum: 10.30 (actual: 10.0; residual: −0.30).

(b) The value factor (HML) contributed most in **2022**: +0.90%. This makes sense—2022 saw a value rotation as rising interest rates punished growth stocks and rewarded value stocks.

(c) The estimated alpha is approximately +0.4% per year. Given only 5 data points and non-trivial residuals, this is not statistically significant, but the point estimate is slightly positive.

Note: with 5 data points, these estimates are imprecise. The purpose of the exercise is to practice the decomposition methodology, not to draw firm statistical conclusions.