Lecture 19: Causal Wrap-Up + Final Review

MS&E 125 — Applied Statistics

Today's structure

Part 1: Causal wrap-up

DAG patterns

RCT / A-B tests

Natural experiments

Difference-in-differences

Part 2: Cumulative review

Recap

Practice questions

Causal inference wrap-up

The causal question

Prediction asks:

What is likely to happen?

Causal inference asks:

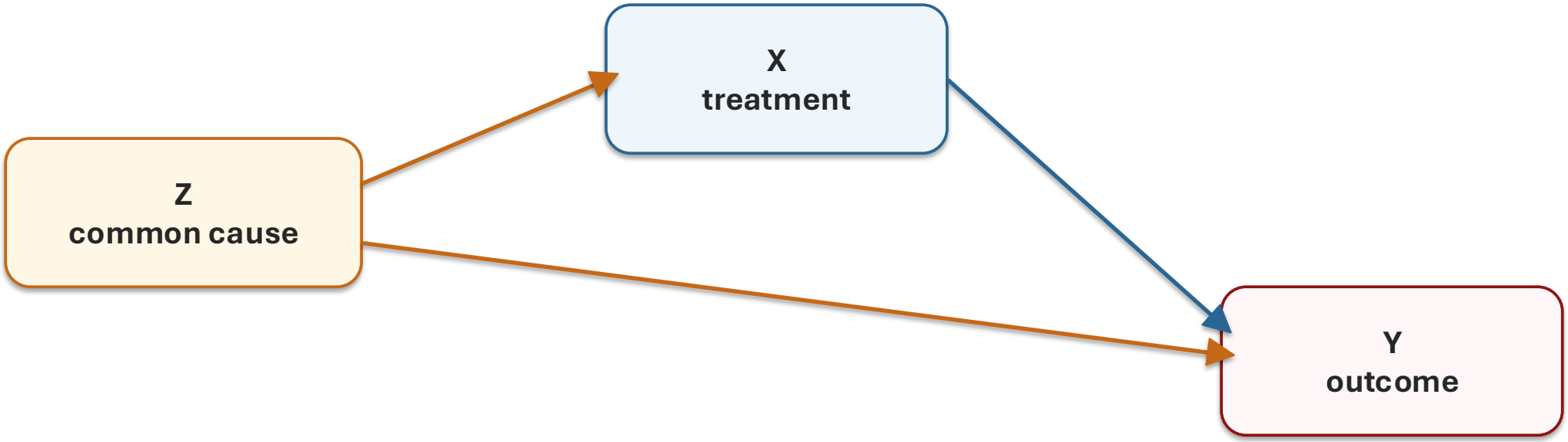
What would happen if we changed something?

Fundamental problem

For each unit, we want both potential outcomes: Y(1) and Y(0).

But we only observe one of them.

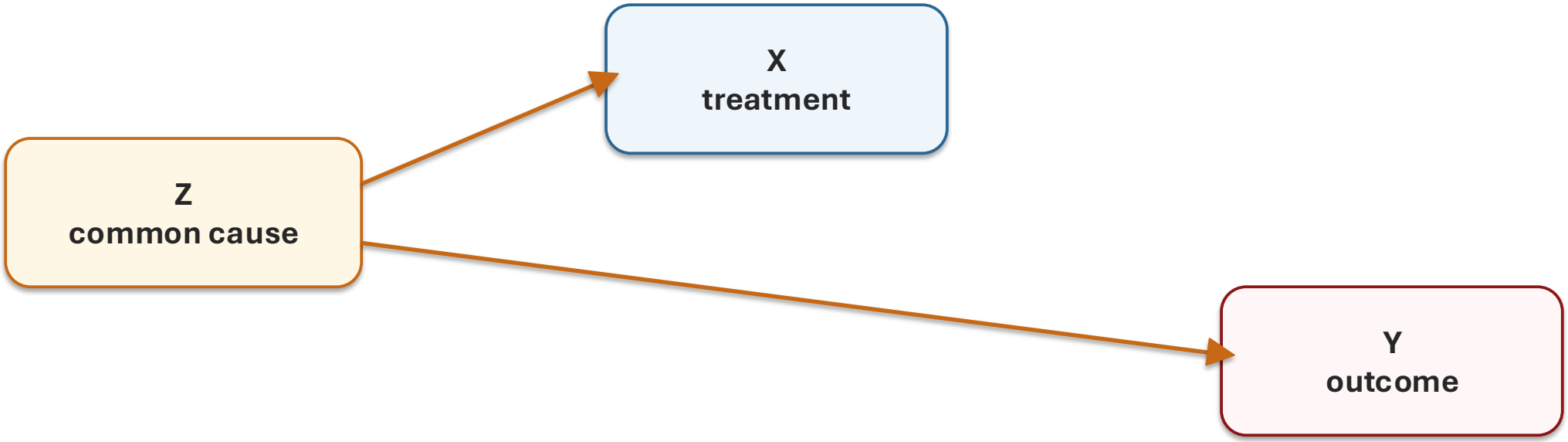
Motif 1: confounder



A confounder Z is a common cause for both treatment X and outcome Y.

Example: preparation affects both tutoring and exam scores. This is the pattern behind many misleading treatment comparisons.

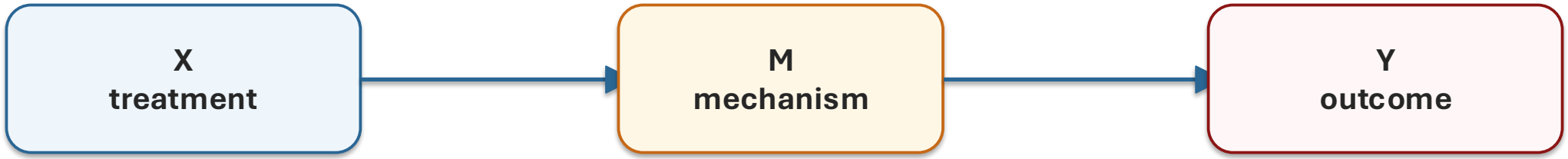
Motif 1: confounder



A confounder Z is a common cause for both treatment X and outcome Y.

Example: preparation affects both tutoring and exam scores. This is the pattern behind many misleading treatment comparisons.

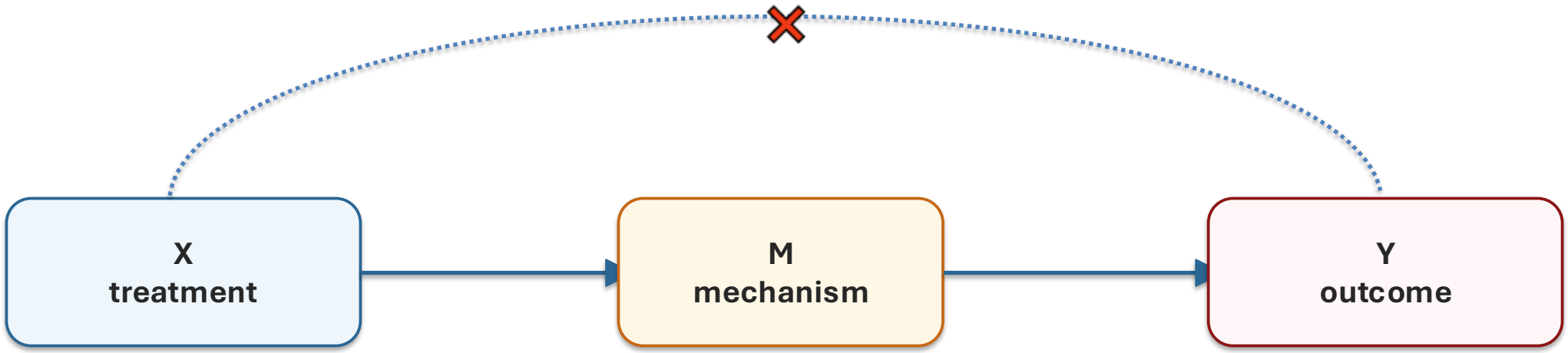
Motif 2: mediator



A mediator X is a captures how the treatment X effect outcome Y.

Example: college may affect alumni network, which may affect earnings. A mediator is part of the causal pathway, not a pre-treatment common cause.

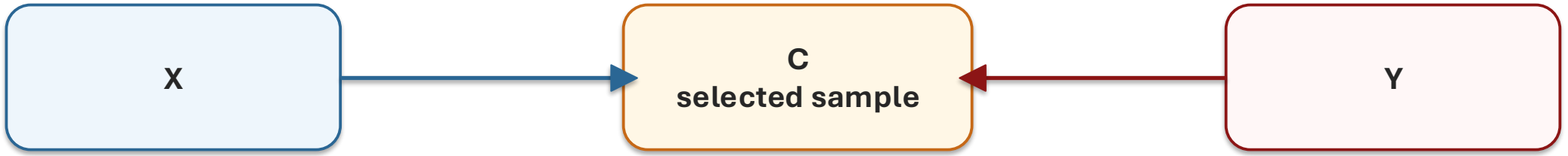
Motif 2: mediator



A mediator X is a captures how the treatment X effect outcome Y.

Example: college may affect alumni network, which may affect earnings. A mediator is part of the causal pathway, not a pre-treatment common cause.

Motif 3: collider



A collider measures the common effect of X and Y.

Example: if we only look at people hired by a top firm, talent and connections may become related inside that selected group even if they were unrelated in the full population.

Causal designs: what creates the comparison?

RCT / A-B test

Researcher randomizes treatment.

Best defense against confounding.

Natural experiment

External process creates near-random treatment variation.

Example: different water companies in Snow's cholera study.

Difference-in-differences

Compare before/after changes in treated and control groups.

Key assumption: parallel trends.

In an RCT, the comparison is built by design

Group	Observed outcome	Why credible?
Treatment group	Y(1)	Randomization makes the group comparable to control before treatment
Control group	Y(0)	Control outcomes estimate the missing untreated outcomes for treated units

Average treatment effect ≈ mean(treated outcomes) – mean(control outcomes)

Why natural experiments?

Many causal questions cannot be randomized for ethical, legal, or practical reasons.

Sometimes the world creates variation that resembles random assignment.

The analyst's job is to argue why the treated and comparison groups are comparable.

Natural experiment logic: not "we controlled for everything," but "the assignment mechanism created a fair comparison."

Before/after alone is not enough

A hospital introduces a new discharge protocol. Readmissions fall from 20% to 15%.

Question:

Did the protocol cause the drop, or were readmissions falling everywhere?

We need a control group to estimate the background trend.

Treated/control after treatment is not enough either

Comparison	Problem
Before/after among treated units	Time trends may explain the change.
Treated/control after treatment	Groups may have differed before treatment.
Difference-in-differences	Uses both before/after and treated/control comparisons.

The 2×2 difference-in-differences table

Hospital	Before	After	Change
Treated hospital	20%	15%	−5 pp
Control hospital	18%	16%	−2 pp
Difference-in-differences			−3 pp

DiD = (15 − 20) − (16 − 18) = −3 percentage points

The 2×2 difference-in-differences table

Hospital	Before	After	Change
Treated hospital	20%	15%	−5 pp
Control hospital	18%	16%	−2 pp
Difference-in-differences			−3 pp

DiD = (15 − 20) − (16 − 18) = −3 percentage points

Parallel trends is the causal assumption

It does not require the treated and control groups to have the same level.

It requires that, absent treatment, their changes over time would have been similar.

Good sign	Warning sign
Pre-treatment trends look similar.	Trends diverge before treatment.
Control group faces same background shocks.	Control group affected by different shocks.
Treatment timing is clearly defined.	Treatment adoption is gradual or anticipatory.

Final review

Course map

Act 1: Build Models

1. Applied statistics
2. EDA
3. Data munging
4. Regression
5. Multiple regression
6. Validation
7. Classification

Act 2: Trust Models

8. Bootstrap / CLT
9. Permutation tests
10. Hypothesis tests
11. Multiple testing
12. Regression inference + classification inference

Act 3: See Further

13. Trees / forests
14. PCA
15. Clustering
16. Deployment validation
17. Working with AI
18–19. Causal inference

About practice final

Tool literacy

Choose the right method.

Compute some quantities.

Know formulas and definitions.

Interpretation / EDA

Read plots.

Interpret coefficients.

Critique missingness, confounding, and design.

Diagnose + supervise

Read/Audit AI code.

Spot leakage, shift, metric gaming.

State what to check next.

Act 1: Build Models

Data, features, regression, validation, classification

Ch 1: Introduction to applied statistics

Keep in mind

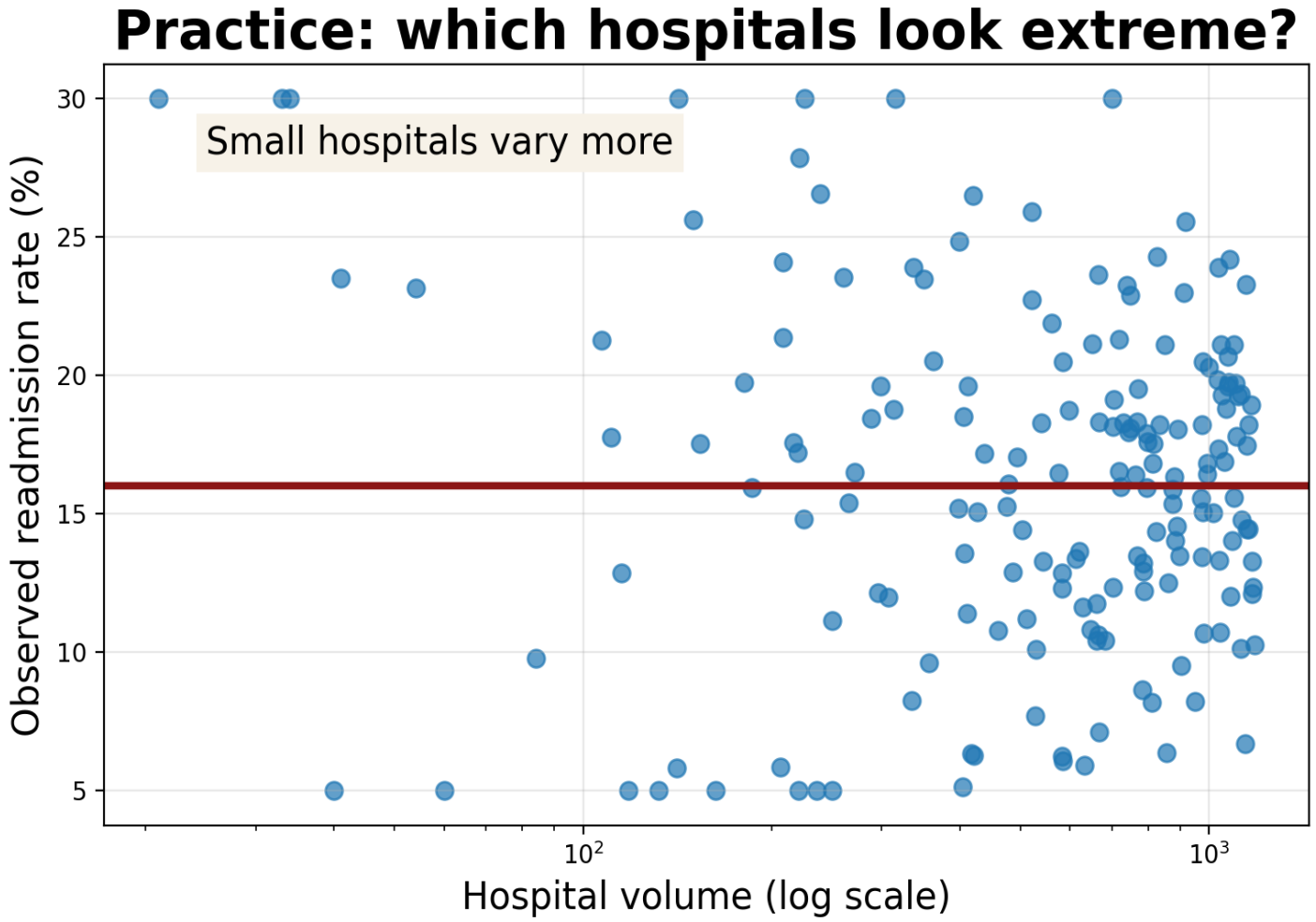
- **Summary**: describe what happened
- **Prediction**: forecast what will happen
- **Inference**: quantify uncertainty about a conclusion
- **Causation**: determine what would happen under an intervention

Ch 1: Introduction to applied statistics

Computational tools

- `pd.read_csv()` — load a CSV file into a DataFrame
- `df['column_name']` — select a single column (returns a Series)
- `.head()` — peek at the first few rows
- `.describe()` — summary statistics (mean, std, min, max, quartiles)
- `.value_counts()` — count unique values in a column
- `sns.histplot()` — plot a histogram
- `df[df['col'] == value]` — filter rows by a condition (boolean indexing)

Ch 1: Introduction to applied statistics



Question

A small hospital has the highest observed readmission rate.

Should we conclude the worst hospital?

Practice 1 answer

Answer

No. Small denominators create noisy rates.

The graph shows wider variation among low-volume hospitals.

Before ranking, quantify uncertainty and check whether the apparent extreme could be sampling noise.

Ch 2: Exploratory Data Analysis

Main idea

Before modeling, understand what the data looks like.

- Distributions shift?
- Missingness?
- Outliers?
- Skewness?
- Other patterns?

Core methods

- Histograms,
- Boxplots,
- Scatter plots,
- Density plots,

- Summary statistics,

Keep in mind

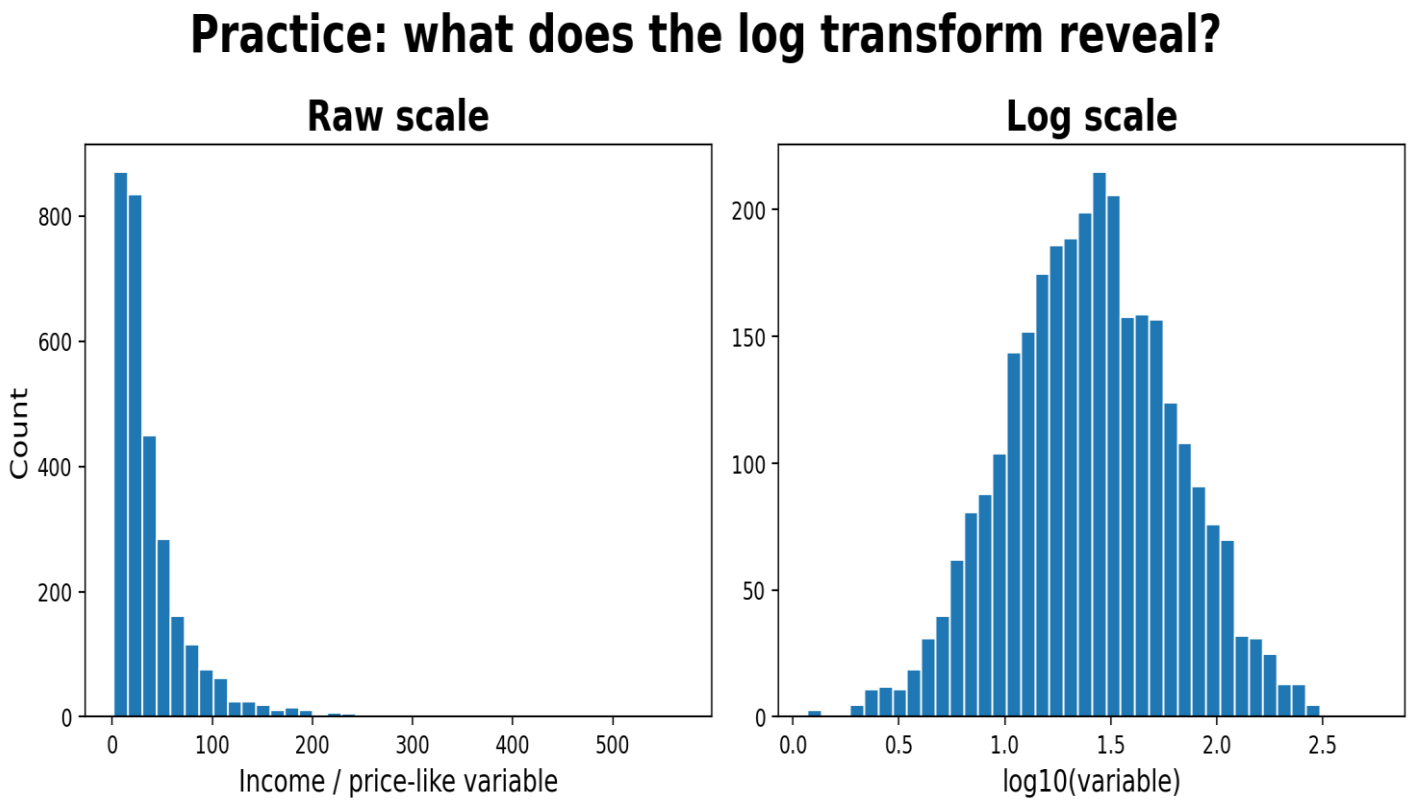
Plot choice depends on variable types. Always check what the plotting function dropped.

Ch 2: Exploratory Data Analysis

Computational tools

- `df.shape`, `df.head()`, `df.info()`, `df.describe()` — first moves in any EDA
- `df['col'].isna().sum()` — count missing values
- `df['col'].value_counts()` — frequency table for categorical data
- `sns.histplot()` — histogram of one numeric variable
- `sns.boxplot(data=df, x='cat', y='num')` — compare distributions across groups
- `sns.violinplot(data=df, x='cat', y='num')` — compare full distribution shapes across groups
- `sns.displot(col='group')` — faceted histograms for comparing distributions
- `df.groupby([col1, col2])[val].median().unstack()` — contingency table of summaries
- `pd.crosstab()` — contingency table of counts for two categorical variables
- `ax.set_xscale('log')` — set axis to log scale
- `df.plot.bar(stacked=True)` — stacked bar chart

Practice 2: skew and log scale



Question

What changes when we plot the variable on a log scale?

What would be misleading on the raw scale?

Practice 2 answer

Answer

The raw-scale histogram hides most observations near zero and exaggerates a long right tail.

The log scale reveals the body of the distribution and makes multiplicative differences easier to see.

Ch 3: Data munging

Keep in mind

Data munging:
The process of cleaning, transforming, and preparing raw data for analysis. Every cleaning choice is a decision that changes your conclusions — there is no “neutral” default.

Missingness encodings:
- **MCAR** (unrelated to any variable),
- **MAR** (depends on observed variables only),
- **MNAR** (depends on the unobserved value itself)

Missingness can be informative.

How to deal with missingness?

Ch 3: Data munging

> ⓘ **Data munging checklist**
>
> 1. **Check provenance.** Where did the data come from? Is there a data dictionary? A methodology document? If not, proceed with caution.
> 2. **Check types.** Run `df.dtypes`. Any `object` columns that should be numeric? Any numeric columns that are really categories?
> 3. **Find the missing data.** Check `df.isna().sum()`, but also look for sentinel values (`-999`, `0`, `99`), string placeholders (`"PrivacySuppressed"`, `"PS"`, `"Not Available"`), and empty strings.
> 4. **Understand *why* data is missing.** Is the missingness random (MCAR), related to observed variables (MAR), or related to the missing value itself (MNAR)?
> 5. **Check joins.** After every merge, compare row counts. Did you gain or lose rows? Which records were dropped or duplicated? Add an `assert`.
> 6. **Standardize strings.** Lowercase, strip whitespace, collapse variants. Document each grouping decision.
> 7. **Deduplicate deliberately.** Check `df.duplicated()`. Decide what columns define "the same record" — and document that choice.
> 8. **Verify type conversions.** After `pd.to_numeric(errors='coerce')`, check how many values became NaN. Know what you lost.
> 9. **Compare groups.** Is the missingness evenly distributed across the groups you care about? If not, group comparisons are biased.

Ch 3: Data munging

Computational tools

- `pd.merge(left, right, on=..., how=...)` — join two DataFrames; `how` controls inner/left/right/outer
- `pd.to_numeric(series, errors='coerce')` — convert strings to numbers, turning failures into NaN
- `pd.cut(series, bins, labels)` — bin continuous values into discrete intervals
- `.is_unique` — True if all values in a Series are distinct (useful for checking join keys)
- `.dropna(subset=[...])` — drop rows with NaN in specified columns
- `.fillna(value)` — fill NaN with a specified value
- `.isna()` / `.notna()` — boolean mask for missing / non-missing values
- `.value_counts()` — count unique values (useful for spotting unexpected entries like "PrivacySuppressed")
- `.str.lower()`, `.str.strip()`, `.str.replace(pat, repl, regex=True)` — string standardization
- `.duplicated(subset=[...])` / `.drop_duplicates()` — find and remove duplicate rows
- `.groupby(col).agg(...)` — split data by groups and compute summary statistics
- `.astype(dtype)` — cast a Series to a specified type (e.g., `float`, `int`, `str`, `'category'`)
- `assert condition, message` — defensive check; raises an error if the condition is false

Practice 3: missingness audit

Question

A dataset has 10,000 schools. Only 1,200 have SAT and earnings data.

An analyst drops the other 8,800 rows and fits a model.

What should you ask before trusting the result?

Think about

Are the missing rows random?

Which school types disappear?

Does the target population change?

Practice 3 answer

Answer

Ask whether complete cases represent the population you care about.

If missingness excludes community colleges, for-profit schools, or less selective schools, the model describes a selected subset, not all institutions.

Ch 4: From the mean to simple regression

Main idea

Regression projects y onto a linear space built from x. The residuals are what the line misses.

Assumptions behind the linear regression ?
(more in regression diagnosis)

Core methods

Correlation,

least squares,

residuals,

R^2,

projection,

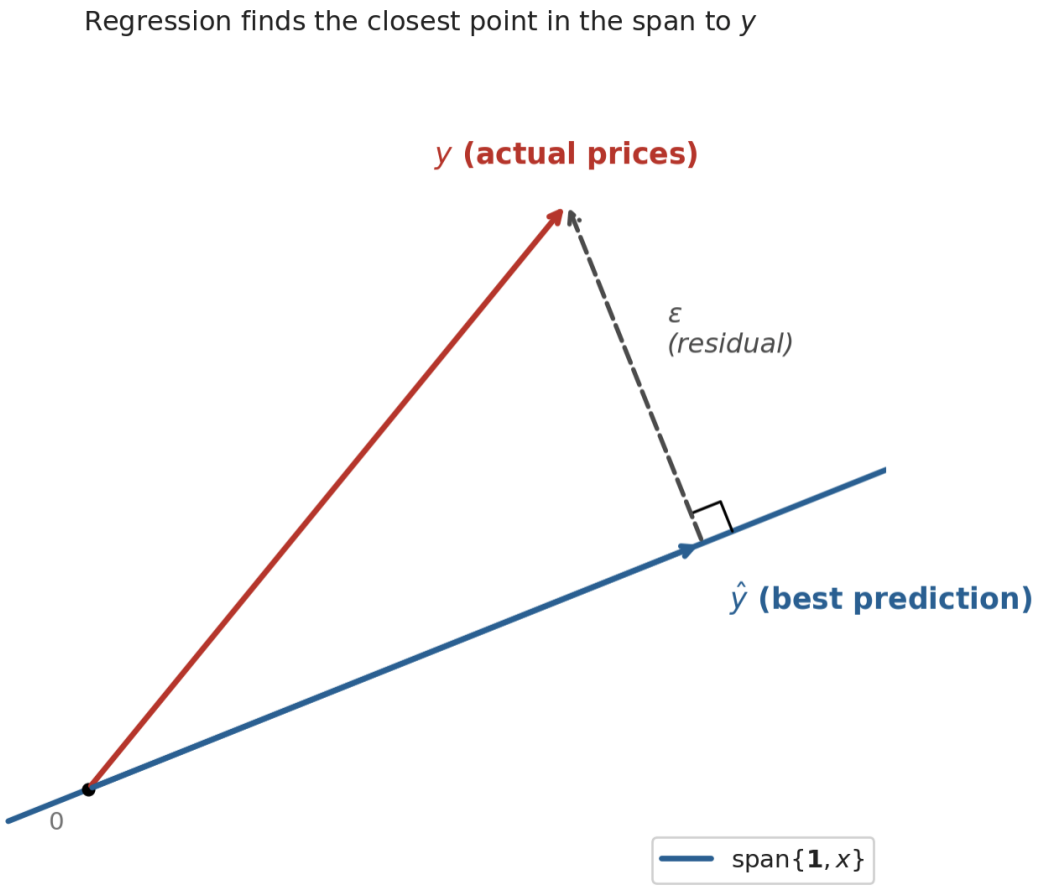
regression to the mean.

Keep in mind

A slope is an association unless the design justifies causality

Ch 4: From the mean to simple regression

Geometric interpretation of linear regression?



Ch 4: From the mean to simple regression

Geometric interpretation of linear regression?

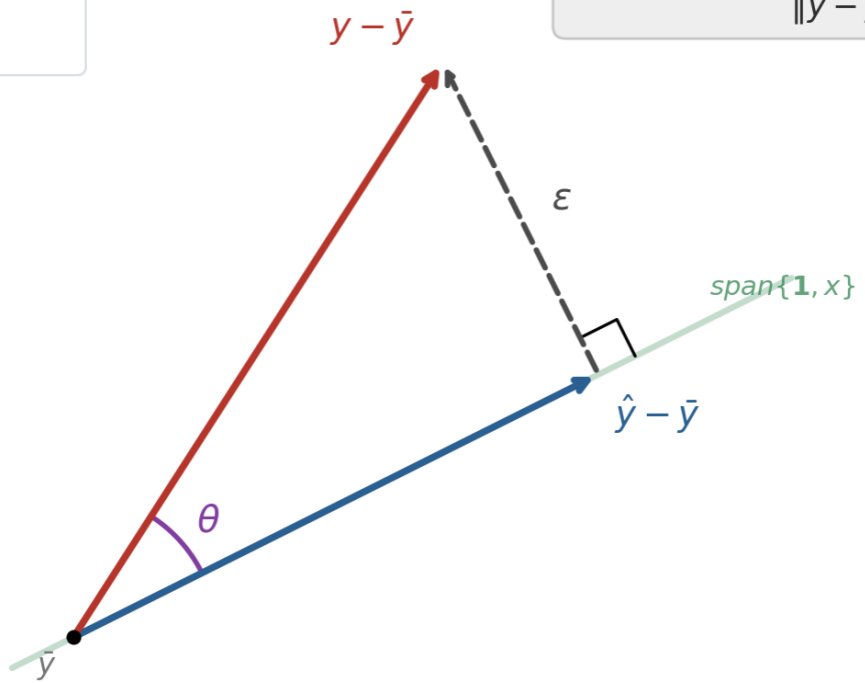
Definition: R^2 (coefficient of determination)

$$R^2 = 1 - \frac{\|\epsilon\|^2}{\|y - \bar{y}\|^2} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

R^2 measures the fraction of variance in y explained by the model. A value of 0 means the model does no better than predicting $\bar{y}$ for every listing. A value of 1 means a perfect fit.

R^2 measures how close y is to the span

$$R^2 = r(y, \hat{y})^2 = \frac{\|\hat{y} - \bar{y}\|^2}{\|y - \bar{y}\|^2}$$



After centering y (subtracting $\bar{y}$), the Pythagorean theorem gives an exact decomposition:

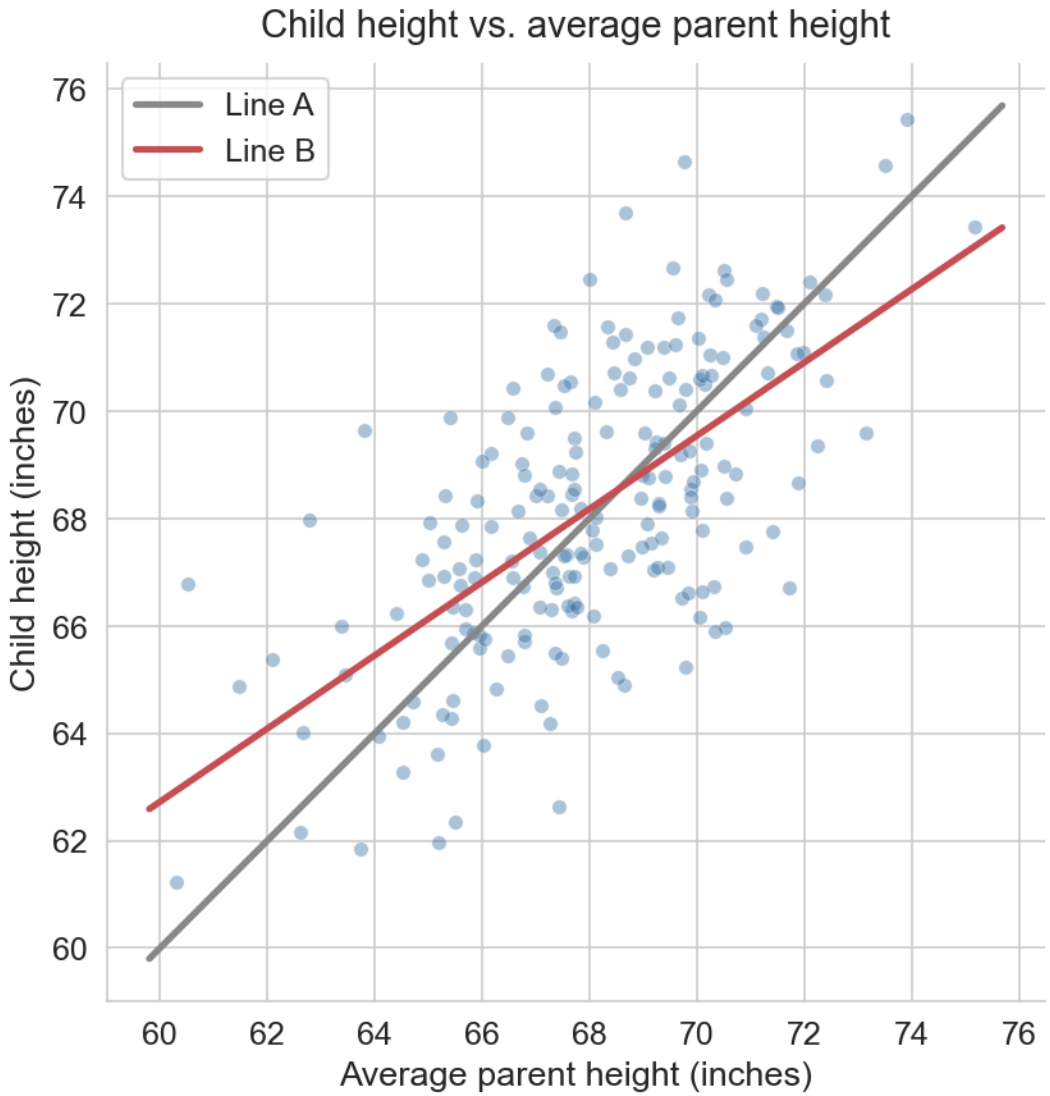
$$\|y - \bar{y}\|^2 = \|\hat{y} - \bar{y}\|^2 + \|\epsilon\|^2$$

The centered prediction and the residual form the two legs of a right triangle, with the centered response as the hypotenuse. Dividing through:

$$1 = \frac{\|\hat{y} - \bar{y}\|^2}{\|y - \bar{y}\|^2} + \frac{\|\epsilon\|^2}{\|y - \bar{y}\|^2} = R^2 + (1 - R^2)$$

The first ratio has a familiar name. By the definition of correlation, $\|\hat{y} - \bar{y}\| / \|y - \bar{y}\|$ is exactly $r(y, \hat{y})$ — the correlation between the actual and predicted prices. So $R^2 = r(y, \hat{y})^2$: it measures how closely the centered prediction vector aligns with the centered response, with perfect alignment ($r = \pm 1$) giving $R^2 = 1$ and orthogonal vectors ($r = 0$) giving $R^2 = 0$.

Practice 4: regression to the mean



Question

The plot below shows child height vs. average parent height for a sample of families. Two lines are overlaid: **Line A** (gray) and **Line B** (red).

Which line is the regression line of child on parent, and which is y = x ?

What does the relative slope of the two lines reveal about heights across generations?

Practice 4 answer

Solution

(a) Line B (red, flatter) is the regression line. Line A is $y = x$. The regression line always has slope $r \cdot \mathrm{SD}_y / \mathrm{SD}_x$, and since $|r| < 1$, the slope is shallower than the 45-degree line.

(b) The flatter regression line shows **regression to the mean**: children of tall parents tend to be tall, but less tall than their parents on average; children of short parents are short but less short. This is purely a statistical consequence of imperfect parent-child correlation, not a biological mechanism.

Ch 5: Multiple regression + feature engineering

Main idea

Coefficients are conditional: “holding the other model variables fixed.”

Core methods

- One-hot encoding,
- Interactions,
- Transformations,
- log outcomes,
- adjusted R^2.

Keep in mind

Feature engineering changes the question. Interactions say one effect depends on another variable.

Ch 5: Multiple regression + feature engineering

Transformation:

The four transform combinations

Depending on whether you log-transform y, x, or both, the coefficient takes on a different interpretation.

Interpreting coefficients with log transforms

Model	Coefficient β_1 means
$y \sim x$	a 1-unit increase in x adds β_1 to y
$\log(y) \sim x$	a 1-unit increase in x multiplies y by e^{β_1}
$y \sim \log(x)$	a 1% increase in x adds about $\beta_1/100$ to y
$\log(y) \sim \log(x)$	a 1% increase in x gives about a β_1% change in y

Ch 5: Multiple regression + feature engineering

Adding more features?

- What can go wrong if we ignore the intercept?
- Adjusted R square?

Adjusted R^2

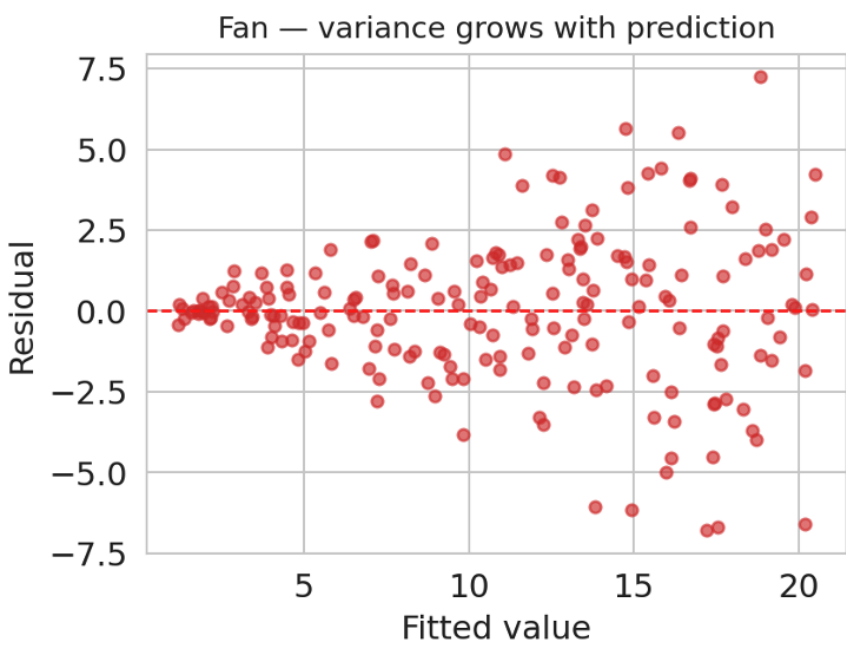
Training R^2 never decreases when you add a feature — even a column of random noise. **Adjusted R^2** penalizes for the number of features:

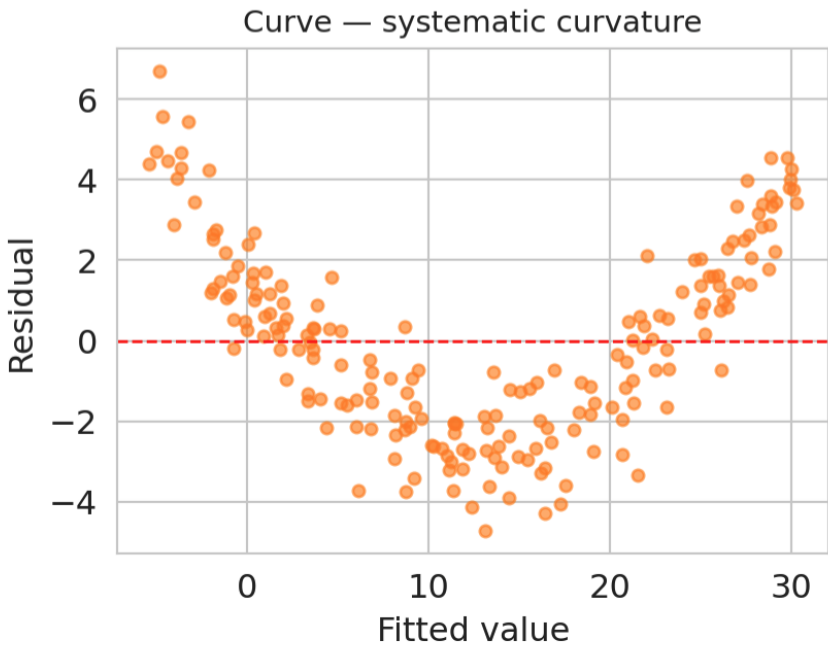
$$R^2_{\text{adj}} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

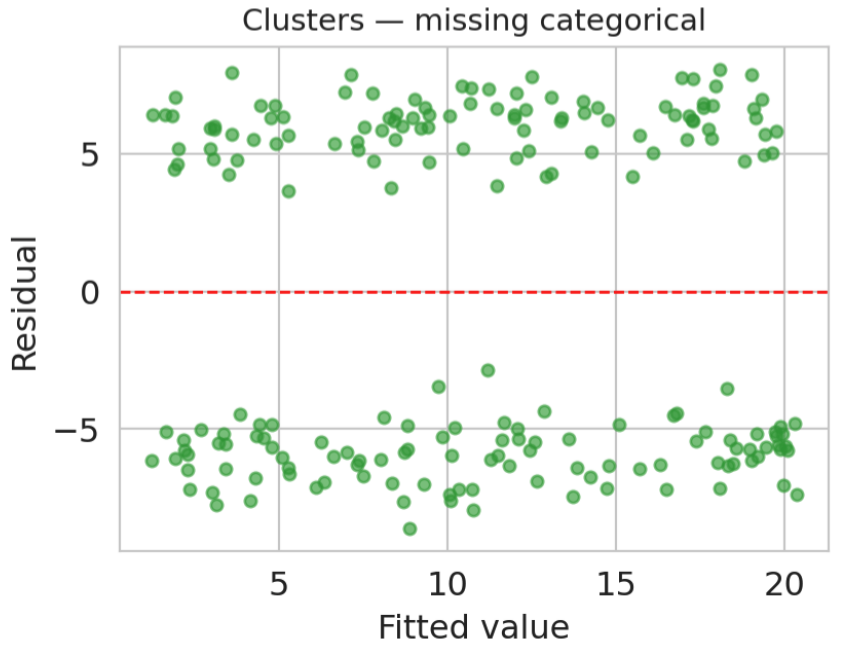
where n is the number of observations and p is the number of predictors. A useless feature increases p without sufficiently reducing residual variance, so adjusted R^2 drops.

Ch 5: Multiple regression + feature engineering

Residual diagnosis?







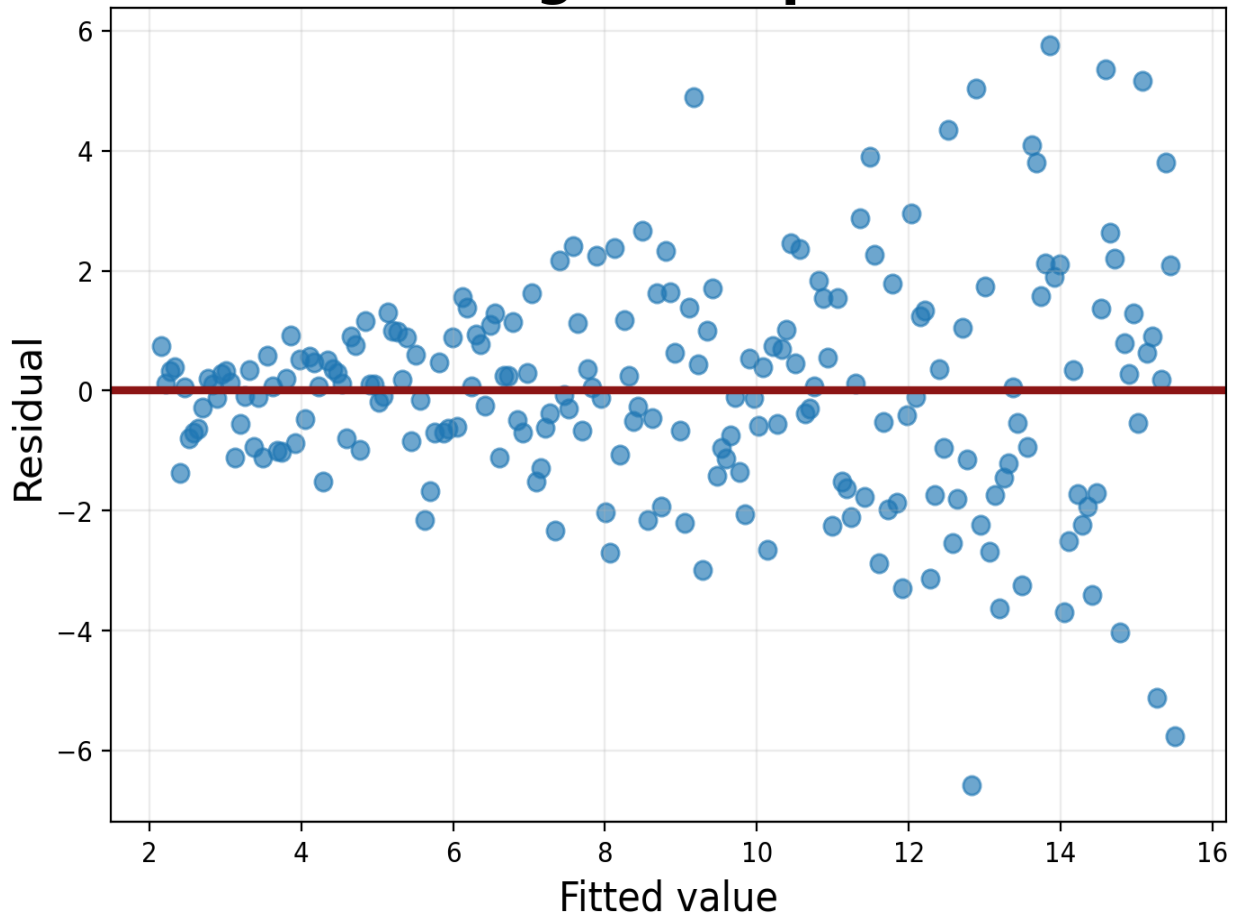
Ch 5: Multiple regression + feature engineering

Computational tools

- `np.linalg.solve(A, b)` — solve $Ax = b$ without forming A^{-1} (for the normal equations)
- `pd.get_dummies(df, drop_first=True)` — one-hot encode categorical variables, dropping the reference level
- `LinearRegression().fit(X, y)` — fit a linear model; `.coef_` for slopes, `.intercept_` for intercept
- `.score(X, y)` — compute R^2 on data
- `.dropna(subset=[...])` — drop rows with missing values in specified columns
- `.fillna(value)` — fill missing values (e.g., with the median for imputation)
- `.groupby(col)` — split a DataFrame by a categorical column for per-group computation

Practice 5: residual diagnostics

Practice: what diagnostic problem is visible?



Question

The residuals widen as fitted values grow.

Which assumption is questionable?

What might you try next?

Practice 5 answer

Answer

The equal-variance assumption is questionable: residual spread grows with the fitted value.

Possible fixes: transform the outcome, use robust standard errors, model variance, or check whether important subgroups are missing.

Ch 6: Validation and bias–variance

Main idea

Training performance is not deployment performance. Validation estimates generalization.

Core methods

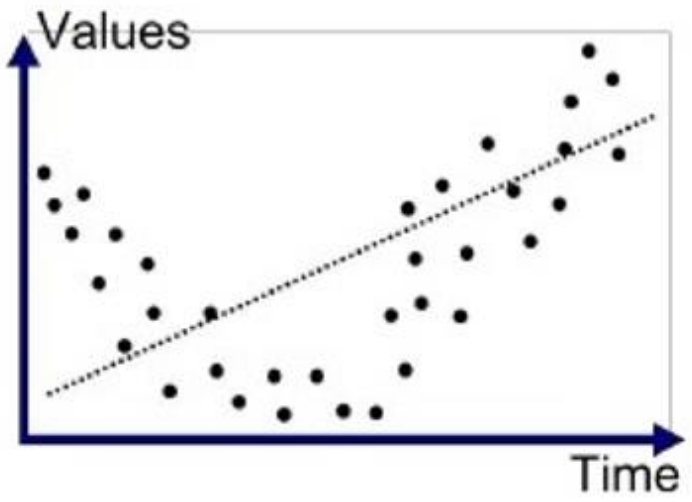
- Train/test split,
- validation sets,
- k-fold CV,
- regularization,
- ridge/lasso,
- bias–variance tradeoff.

Keep in mind

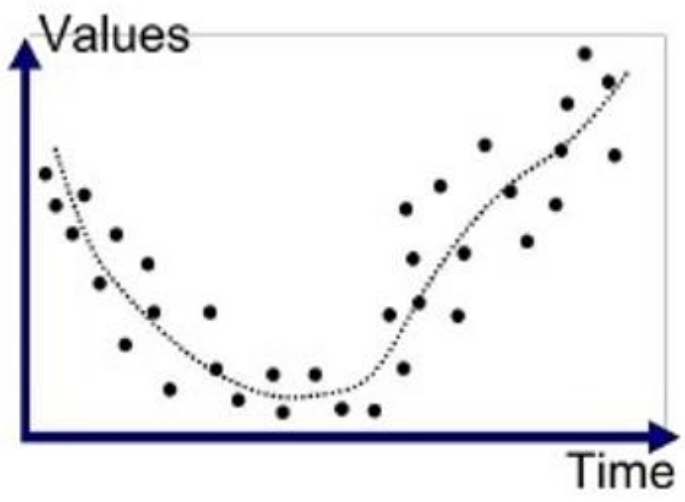
Never use the test set for tuning. Random splits do not protect against distribution shift.

Ch 6: Validation and bias–variance

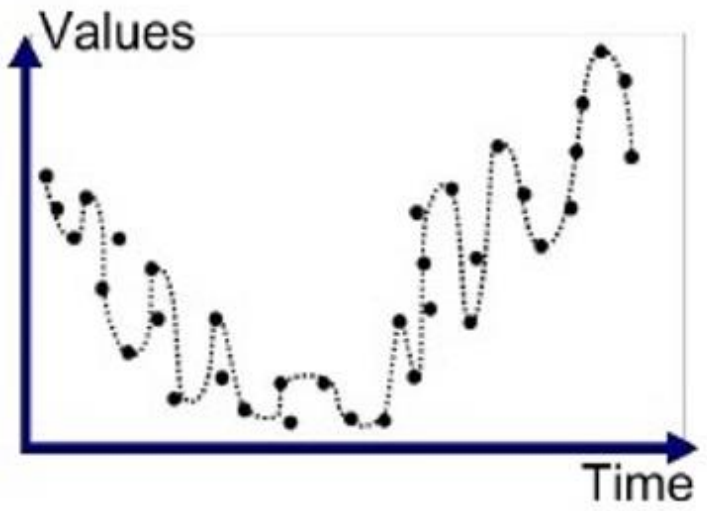
What is underfitting, and what is overfitting?



Underfitted



Good Fit/Robust



Overfitted

Ch 6: Validation and bias–variance

How to fix them?

Fixing overfitting and underfitting

Once you've diagnosed which way the model is failing, the fix is mechanical.

symptom	diagnosis	fix
training R^2 is low, test R^2 is close to it	**high bias** (underfitting)	add features, use a more flexible model class
training R^2 is high, test R^2 is much lower	**high variance** (overfitting)	reduce features, regularize (Lasso / Ridge), or **add more training data**
both R^2 are high and close	sweet spot	nothing to fix
both are low, gap is small	high bias **and** high noise	check for data quality, mismeasured features, wrong outcome

Ch 6: Validation and bias–variance

Train, Validation, Test set?

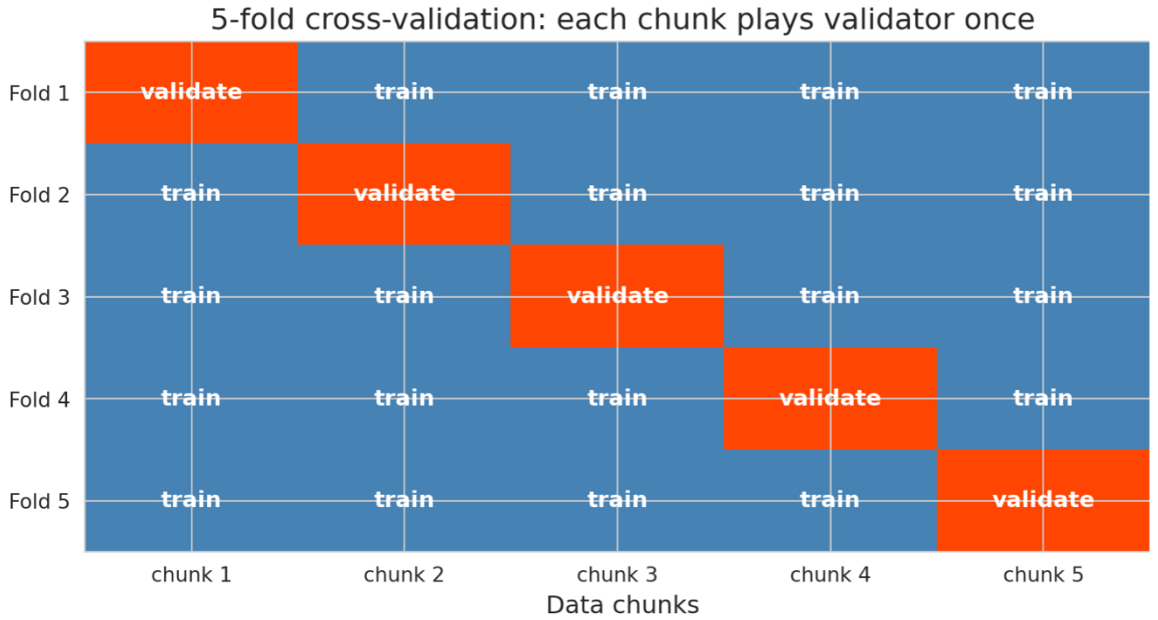
Definition: The three-way split

- **Training set** — the data the model learns from.
- **Validation set** — held-out data used to *choose* model complexity (number of features, regularization strength, tree depth). The modeler sees validation scores and adjusts accordingly.
- **Test set** — held-out data touched *only once*, at the very end, to report final performance. No modeling decisions depend on the test set.

Examples of data leakage?

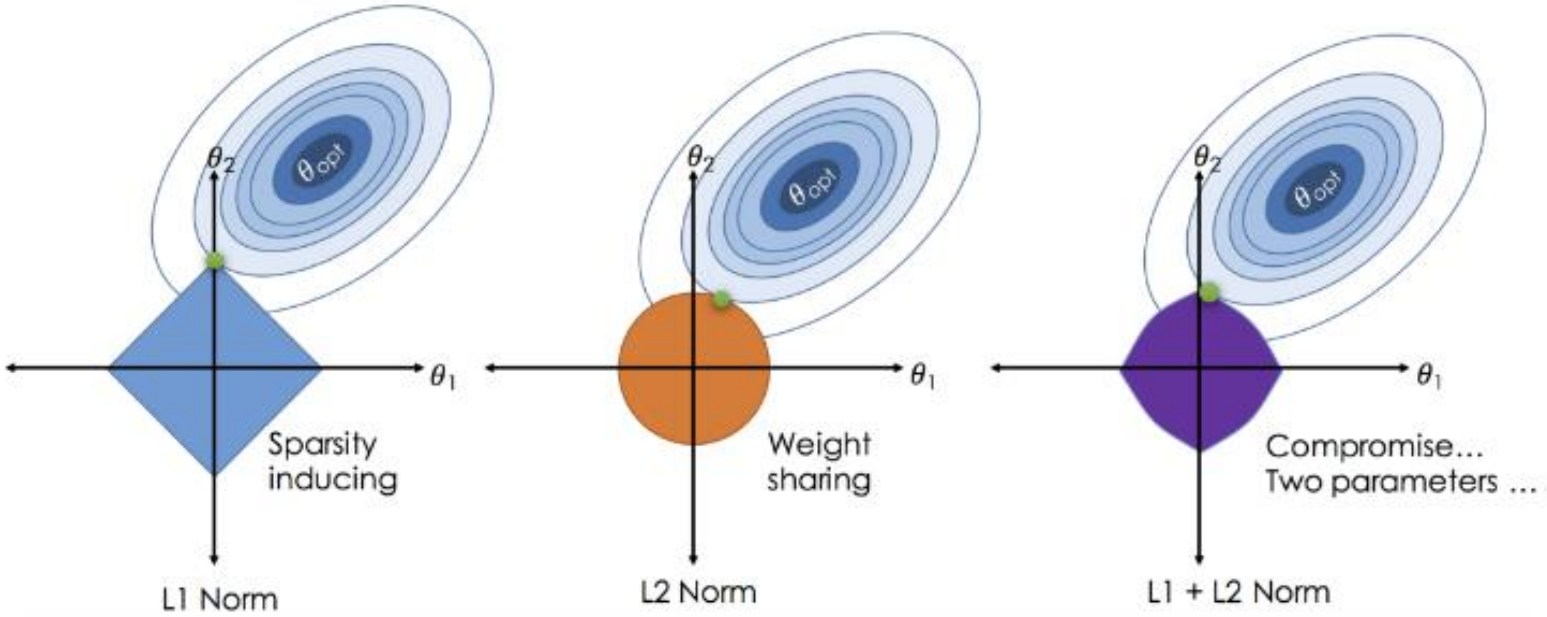
- Doing standardization?

Cross-validation, but why?



Ch 6: Validation and bias–variance

Regularization?

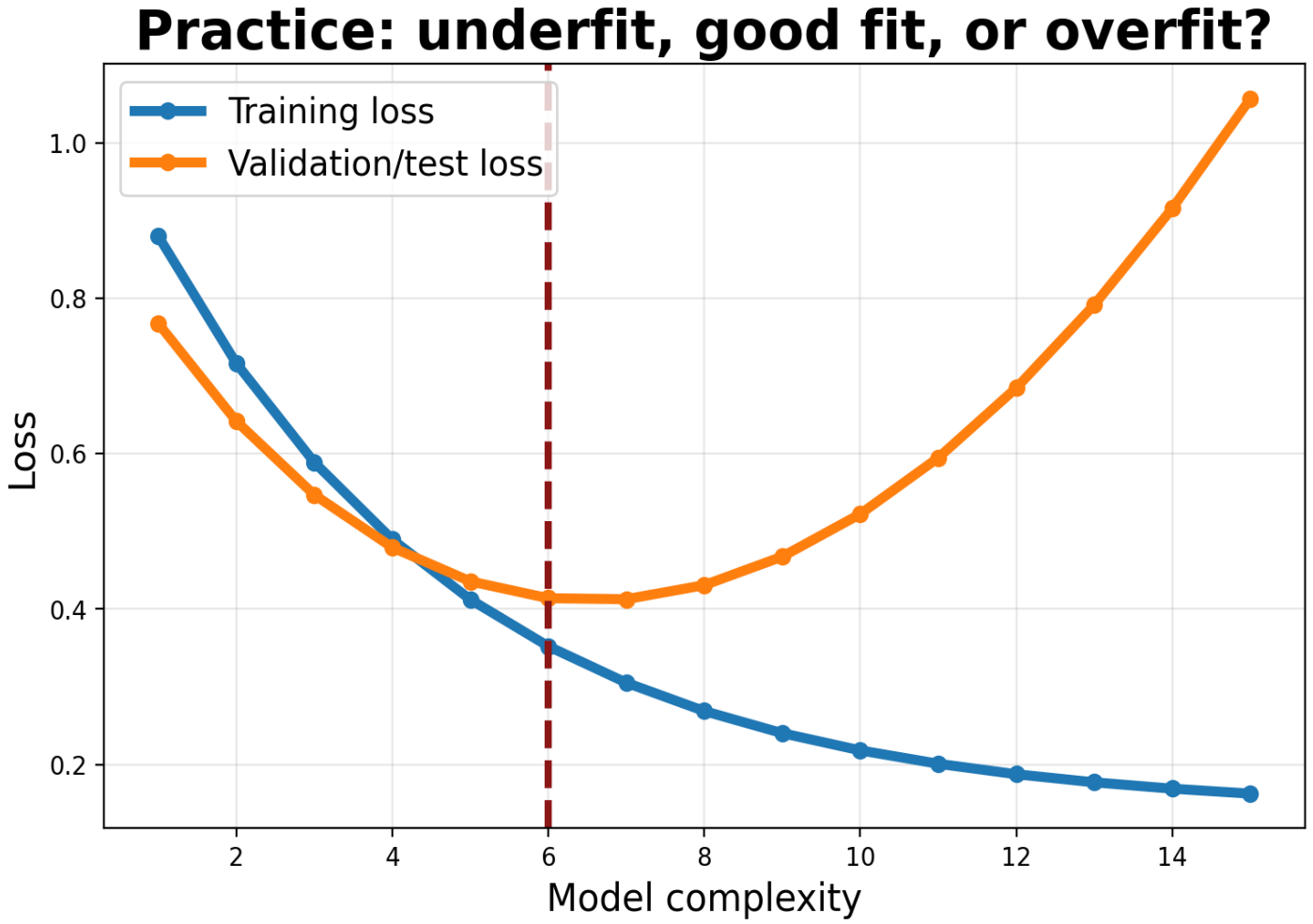


Ch 6: Validation and bias–variance

Computational tools

- `train_test_split(X, y, test_size=0.3)` — splits data into training and test sets.
- `LinearRegression().fit(X, y)` — ordinary least squares (OLS).
- `Ridge(alpha=1.0).fit(X, y)` — L2-regularized linear regression.
- `Lasso(alpha=1.0).fit(X, y)` — L1-regularized linear regression.
- `LassoCV(cv=5).fit(X, y)` — fits Lasso at many α values and selects the best by cross-validation.
- `KFold(n_splits=5, shuffle=True, random_state=...)` — cross-validation splitter; use with `cross_val_score`.
- `cross_val_score(model, X, y, cv=5)` — returns k fold-wise scores for averaging.
- `r2_score(y_true, y_pred)` — computes R^2 on any set of true/predicted values.
- `StandardScaler().fit_transform(X_train)` — standardizes features to mean 0, std 1; use `.transform(X_test)` on test data.
- `PolynomialFeatures(degree=d)` — generates all polynomial terms up to degree d.

Practice 6: validation curve



Question

Where is the model underfitting?

Where is it overfitting?

Which complexity would you choose?

Practice 6 answer

Answer

Low complexity underfits: both losses are high.

High complexity overfits: training loss keeps falling while validation loss rises.

Choose near the minimum validation loss, not the minimum training loss.

Ch 7: Classification

Main idea

Classification converts features into probabilities, then decisions depend on thresholds and costs.

Core methods

- Logistic regression,
- odds ratios,
- confusion matrix,
- precision/recall,
- ROC/AUC,
- calibration.

Keep in mind

Accuracy can be misleading under imbalance. Threshold choice is a policy decision.

Ch 7: Classification

- What is a classification task, why this is different from regression?

- What is logistic regression?

Log-odds can be any real number — just like a linear regression prediction. So let's model the log-odds as a linear function of the features:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

This is **logistic regression**. The name reflects the core assumption: the **logit** (log-odds) is a linear function of the features.

- What is the problem of logistic regression when the data is perfectly linear separable?

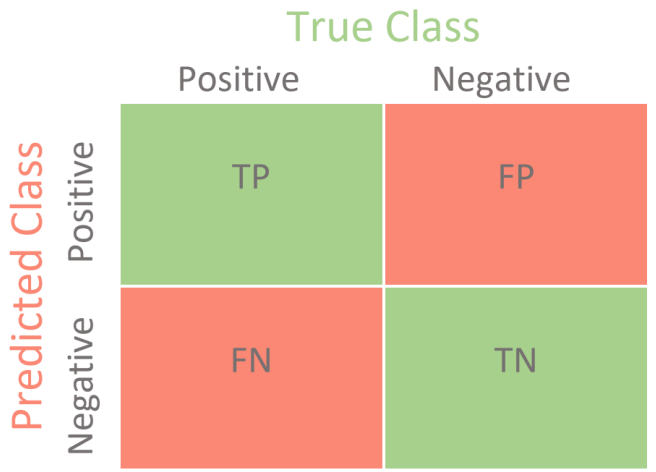
- How to solve logistic regression, why can't just solve like linear regression?

- How to interpret the coefficient from the logistic regression?

Ch 7: Classification

- What is the problem of imbalance class?
- Any metric to use when imbalance class exists?

Metric	Formula	What it measures
Accuracy	(TP + TN) / N	Overall correctness (misleading with imbalance)
Precision	TP / (TP + FP)	Of those flagged, how many are real?
Recall	TP / (TP + FN)	Of real cases, how many caught?
F1 score	2 * Prec * Rec / (Prec + Rec)	Harmonic mean of precision and recall
AUC	Area under ROC curve	Overall ranking ability (threshold-free)



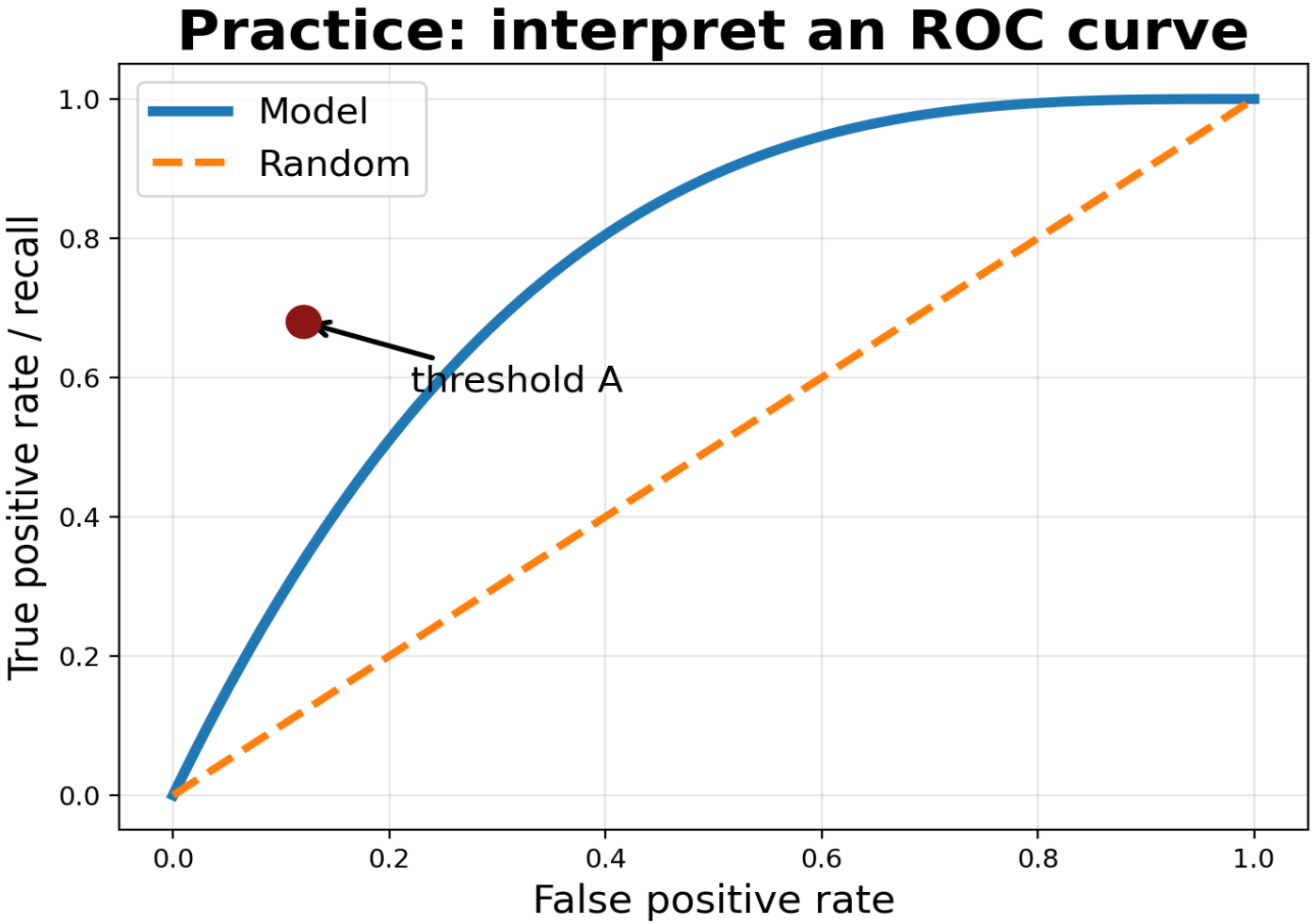
- How to solve class imbalance?

Ch 7: Classification

Computational tools

- `LogisticRegression()` — fit a logistic regression model
- `model.predict_proba()` — get predicted probabilities (not just 0/1 predictions)
- `confusion_matrix()` — compute the 2x2 confusion matrix
- `ConfusionMatrixDisplay()` — plot the confusion matrix
- `roc_curve()`, `auc()` — compute the ROC curve and its area
- `precision_recall_curve()` — compute the precision-recall curve
- `calibration_curve(y_true, y_prob, n_bins=10, strategy='quantile')` — observed frequency vs. mean predicted probability per bin

Practice 7: ROC interpretation



Question

At threshold A, what is high and what is low?

Why might a hospital prefer a different threshold than a marketing team?

Practice 7 answer

Answer

Threshold A has high recall but nonzero false-positive rate.

Different contexts have different costs: missing cancer may be worse than extra follow-up, while spamming customers may be costly.

Act 2: Trust Models

Uncertainty, tests, multiple comparisons, and diagnostics

Ch 8: Bootstrap and normal approximation

Main idea

Estimates vary across samples. We use sampling distributions to quantify uncertainty.

Core methods

- Bootstrap resampling
- standard error,
- confidence interval,
- CLT,
- normal approximation.

Keep in mind

A CI quantifies sampling uncertainty, not confounding, selection bias, or distribution shift.

Ch 8: Bootstrap and normal approximation

- What is bootstrap?
- Why bootstrap?
- What does bootstrap give us?
- When bootstrap fails?
- What is Law of Large Number (LLN)?
- What is Central Limit Theorem (CLT)?

Ch 8: Bootstrap and normal approximation

Computational tools

- `np.random.choice(data, size=n, replace=True)` — one bootstrap resample.
- `np.percentile(bootstrap_dist, [2.5, 97.5])` — 95% bootstrap CI via the percentile method.
- `data.var(ddof=1)` — sample variance (divides by $n-1$, the convention for an estimator from a sample); `np.sqrt(...)` for the SE formula.
- `np.median(...)` for the median; `boot_dist.mean()` and `boot_dist.std()` summarize a bootstrap distribution (the SD estimates the SE).
- `from scipy.stats import norm` for `norm.pdf(x, mu, sigma)` — a normal density evaluated at `x`, used for overlays.
- List comprehension `[f(resample) for _ in range(B)]` — Python shorthand for "do this B times and collect the results into a list."

Ch 9: Permutation tests

Main idea

A permutation test asks: how unusual is this statistic if labels were exchangeable?

Core methods

Choose a statistic, shuffle labels under H0, compare observed statistic to null distribution.

Keep in mind

Validity depends on exchangeability. Bootstrap estimates uncertainty; permutation tests a null.

Ch 9: Permutation tests

- What is a permutation test?

Definition: Permutation test

A hypothesis test that builds the **null distribution** by shuffling group labels and recomputing the test statistic. The null distribution shows what the test statistic would look like if the null hypothesis were true.

- What is a p-value?

Definition: p-value

The probability of observing a result at least as extreme as what you got, assuming the null hypothesis is true.

⚠ Three traps

1. **A p-value is NOT the probability that H_0 is true.** That is the skeptic's flipped-conditional error above.
2. **A p-value is NOT the probability the result will replicate.** Replication depends on power, sampling variability, and whether the effect actually exists — none of which a single p-value pins down.
3. **p = 0.049 and p = 0.051 are not meaningfully different.** The 0.05 threshold is a convention, not a phase transition. Report the number; resist binary verdicts.

Ch 9: Permutation tests

- Bootstrap v.s. Permutation test?

Bootstrap vs permutation: when to use which

These two simulation-based tools answer different questions:

	Bootstrap	Permutation test
Question	How precise is my estimate?	Is the effect real?
Produces	Confidence interval	p-value
Null hypothesis	Not needed	Required
Key assumption	i.i.d. sample	Exchangeability under null
Best for	Any statistic	Comparing groups
Resampling method	With replacement, **within each group** separately	Without replacement, shuffling labels **across groups**

Bootstrap = precision. Permutation = significance. Use both.

Practice 9

Question 10. Bootstrap or permutation? For each scenario, pick the right tool and answer the follow-up.

a. An economist wants a 95% CI for the **median** household income in a sample of 800 households.

Tool: _____

Why this tool? (1 sentence)

b. A health blog observes that people who drink 3+ cups of coffee daily have a 12% lower stroke rate. They run a permutation test on coffee-vs-no-coffee labels and get $p = 0.001$.

Tool used: permutation test. Does $p = 0.001$ license the conclusion *"coffee causes lower stroke rates"*? **Yes / No** — and why?

Practice 9

Solution

(a) Bootstrap. No clean closed-form SE exists for the median (the asymptotic formula involves the unknown population density at the median), so the bootstrap percentile CI is the practical tool. The CI quantifies the *precision* of the median estimate — sampling uncertainty over which 800 households happened to land in the sample.

(b) No. The permutation test is the right tool for testing whether the two groups' stroke rates differ, and $p = 0.001$ confirms they do — but **coffee drinking was not randomly assigned**, so rejecting the null tells us only that the groups *differ*, not that coffee *causes* the lower stroke rate. Coffee drinkers may also exercise more, smoke less, see doctors more often — any of which could produce the gap. Random assignment is what licenses causal claims; without it, a small p-value is association, not causation.

Ch 10: Hypothesis testing framework

Main idea

Hypothesis tests formalize evidence against a null model.

Core methods

- H0/H1,
- p-values,
- α,
- Type I/II error,
- power,
- CI/test duality.

Keep in mind

Fail to reject does not mean the null is true. Statistical significance is not practical importance.

Ch 10: Hypothesis testing framework

A hypothesis test asks:

Is the evidence in the data strong enough to reject the "nothing special is happening" explanation?

Object	Meaning
H_0	Null hypothesis: the default claim, usually "no effect" or "no difference."
H_1	Alternative hypothesis: the claim that something real is happening.
Test statistic	A number summarizing how much the data disagree with H_0.
p**-value**	If H_0 were true, how surprising would our observed result be?
α	Significance level, usually 0.05. It is our false-positive tolerance for one test.

Decision rule:

$$\text{Reject } H_0 \quad \text{if} \quad p \leq \alpha.$$

A small p-value means the data would be unusual if H_0 were true. It does **not** mean that H_0 has probability p of being true.

Ch 10: Hypothesis testing framework

Two possible errors

Reality	Decision	Error Type
H_0 true	Reject H_0	Type I error: false positive
H_0 false	Do not reject H_0	Type II error: false negative

Note:

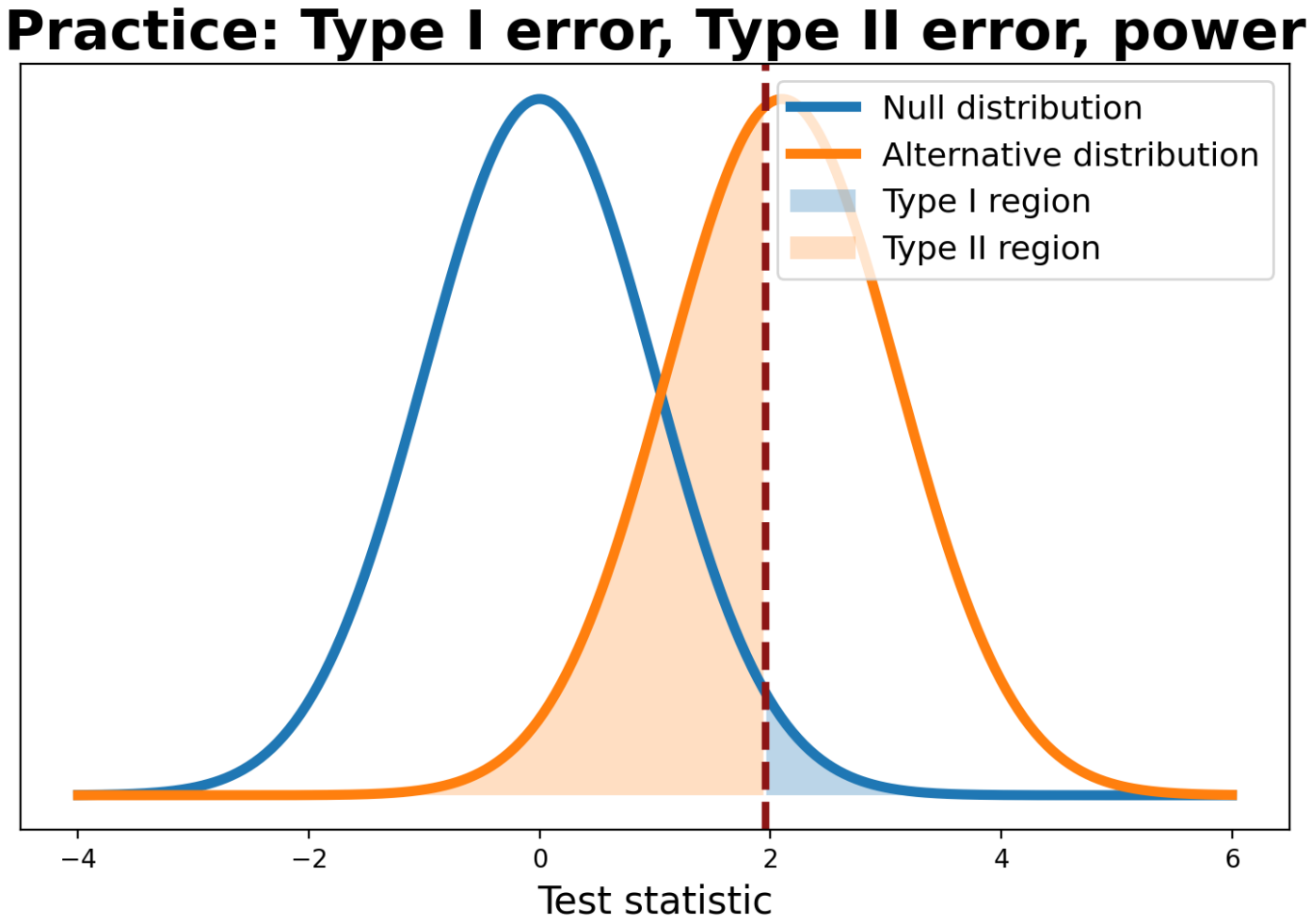
$$p > \alpha \quad \Longrightarrow \quad \text{fail to reject } H_0,$$

not "accept H_0."

Failing to find evidence against H_0 is not the same as proving H_0 is true.

Power: The probability to reject the null when the null is false.

Practice 10: power and errors



Question

Which shaded region is Type I error?
Which is Type II error?
How would increasing sample size affect power?

Practice 10 answer

Answer

Right tail under the null = Type I error.

Left side under the alternative = Type II error.

Increasing sample size usually separates the distributions more, reducing Type II error and increasing power.

Ch 11: Multiple testing (see notes on Canvas)

Main idea

Many tests create false discoveries even when every null is true.

Core methods

- FWER,
- Bonferroni,
- FDR,
- Benjamini–Hochberg,
- p-value histograms.

Keep in mind

Correction is not optional when the search process creates many chances to be fooled.

Ch 12: Regression inference + diagnostics

Main idea

Regression tables are conditional claims; diagnostics ask whether the model assumptions are plausible.

Core methods

t-tests for coefficients, confidence intervals, residual plots, Q-Q plots, LINE assumptions.

Keep in mind

Small p-values do not rescue a bad model. Diagnose before over-interpreting coefficients.

OLS Regression Results

Dep. Variable:	price_clean	R-squared:	0.358
Model:	OLS	Adj. R-squared:	0.357
Method:	Least Squares	F-statistic:	1362.
Date:	Tue, 26 May 2026	Prob (F-statistic):	0.00
Time:	23:41:59	Log-Likelihood:	-82945.
No. Observations:	14689	AIC:	1.659e+05
Df Residuals:	14682	BIC:	1.660e+05
Df Model:	6		
Covariance Type:	nonrobust		

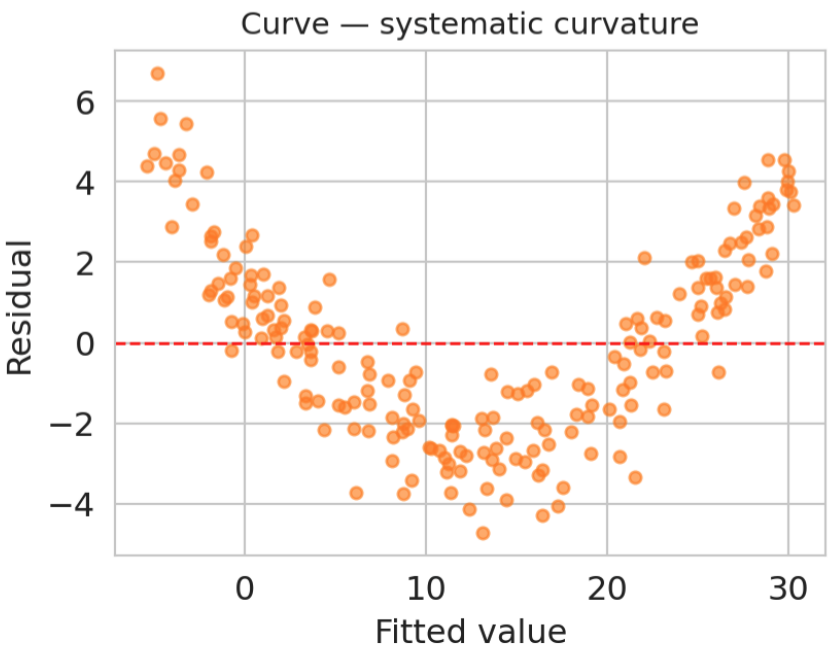
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.3836	5.679	-0.420	0.675	-13.515	8.747
C(borough)[T.Brooklyn]	37.5698	5.481	6.854	0.000	26.826	48.313
C(borough)[T.Manhattan]	95.5836	5.464	17.492	0.000	84.873	106.294
C(borough)[T.Queens]	17.9948	5.794	3.106	0.002	6.638	29.352
C(borough)[T.Staten Island]	-20.1103	9.645	-2.085	0.037	-39.015	-1.206
bathrooms	62.4363	1.861	33.557	0.000	58.789	66.083
bedrooms	34.0280	0.734	46.331	0.000	32.588	35.468

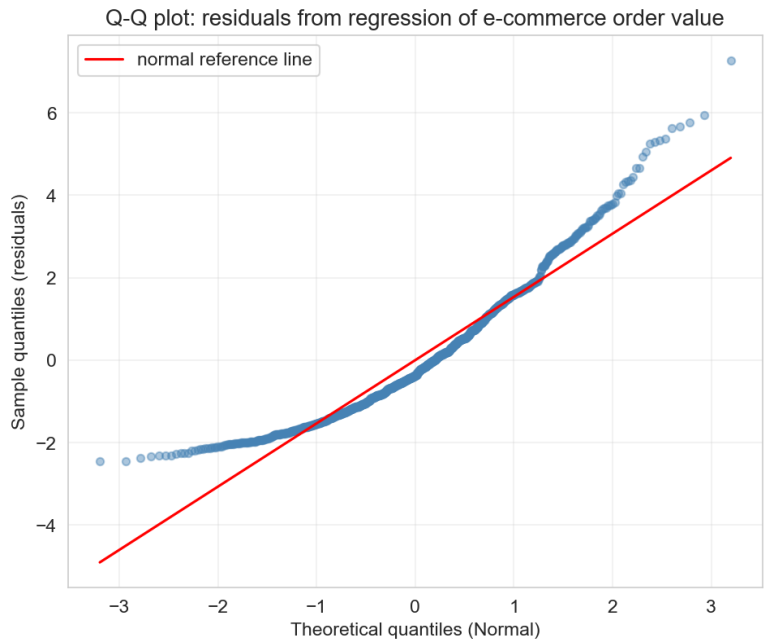
Omnibus:	2189.941	Durbin-Watson:	1.991
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4613.485
Skew:	0.900	Prob(JB):	0.00
Kurtosis:	5.074	Cond. No.	49.3

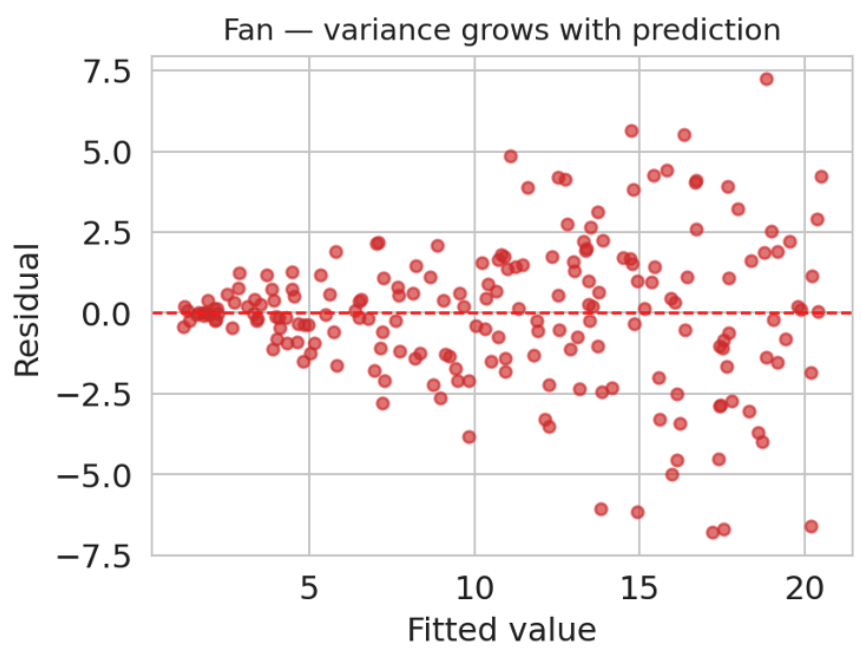
Definition: LINE Conditions

The assumptions for regression inference are remembered by the **LINE** mnemonic:

- **L**inearity: the relationship between predictors and response is linear.
- **I**ndependence: observations are independent of each other.
- **N**ormality: residuals are approximately normally distributed.
- **E**qual variance: the spread of residuals is roughly constant across fitted values (no **heteroscedasticity**).







Act 3: See Further

Flexible models, dimension reduction, unsupervised learning, deployment, and AI

Ch 13: Decision trees and random forests

Main idea

Trees learn if/then partitions. Forests average many noisy trees to reduce variance.

Core methods

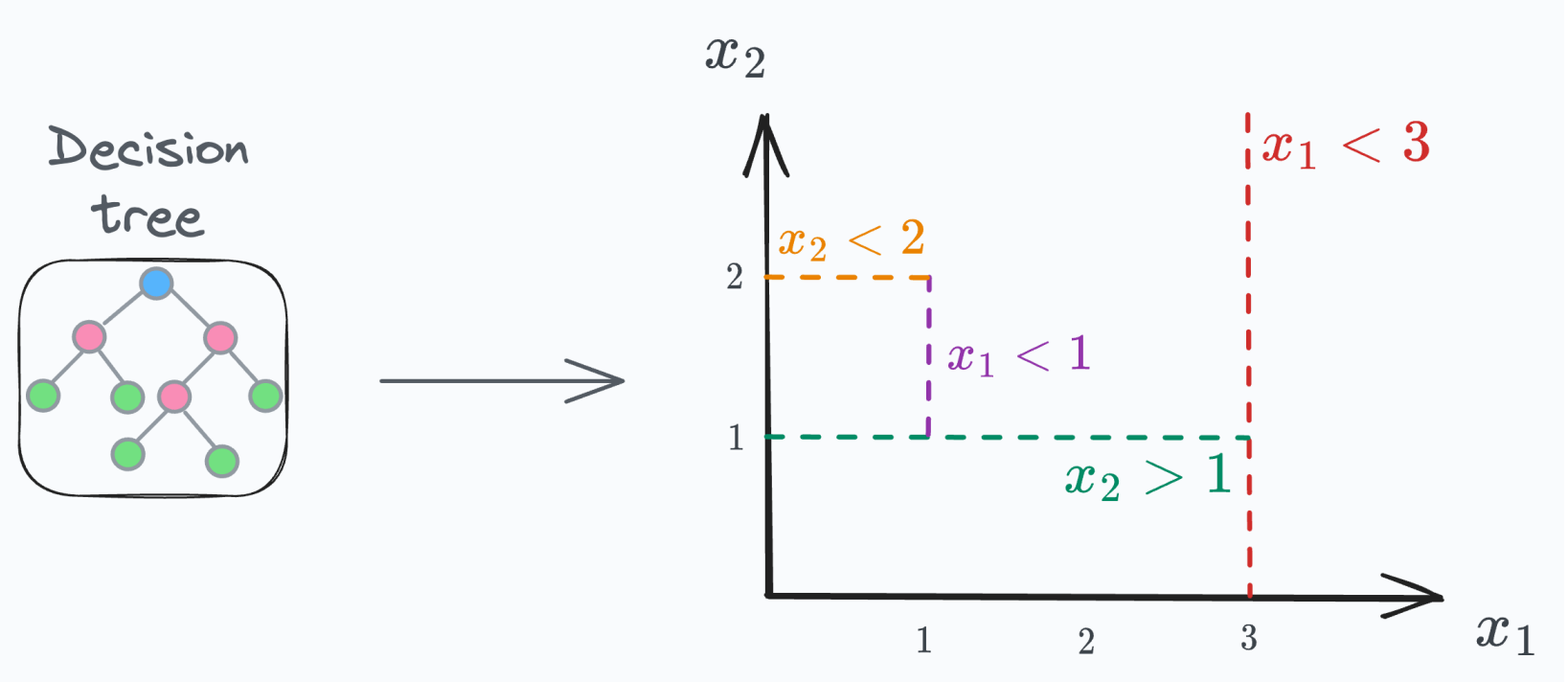
- Splits,
- leaves,
- Gini/SSE,
- max_depth,
- bagging,
- random feature subsets,
- feature importance.

Keep in mind

Deep single trees overfit. More trees in a forest usually stabilize; depth controls overfitting more directly.

Ch 13: Decision trees and random forests

Trees are doing partitions.



Ch 13: Decision trees and random forests

- What are ensemble methods?
  - Bagging?
  - Boosting?

- What are the benefits of ensemble methods, how about the cost?

- What are the advantage of tree-based methods compare to linear method
- How about disadvantages?

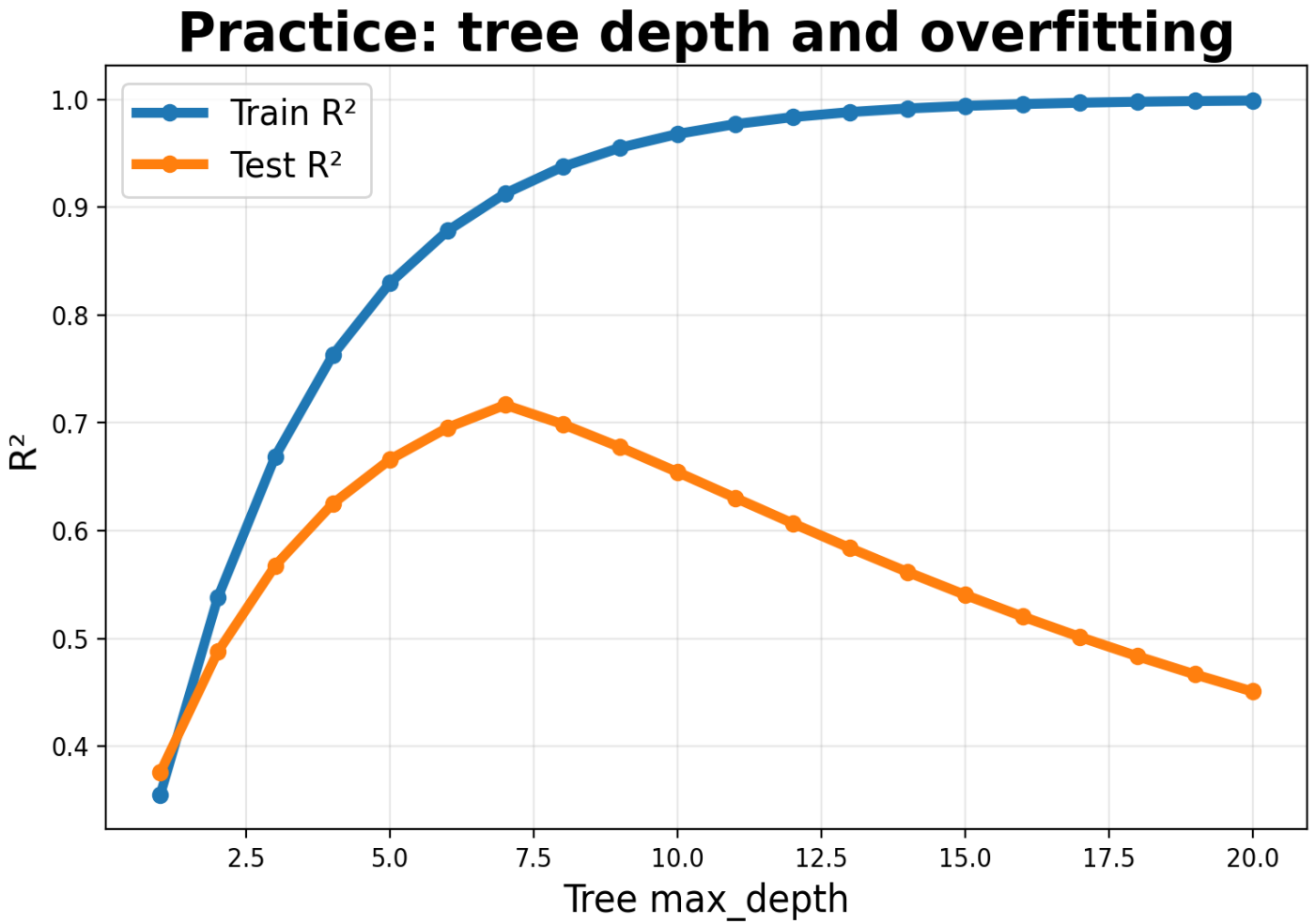
- How to choose depth?

Ch 13: Decision trees and random forests

Computational tools

- `DecisionTreeRegressor(max_depth=4).fit(X, y)` / `DecisionTreeClassifier(max_depth=5).fit(X, y)` — fits a tree with controlled depth.
- `RandomForestRegressor(n_estimators=100).fit(X, y)` / `RandomForestClassifier(...)` — fits a forest.
- `plot_tree(tree, feature_names=..., filled=True)` / `export_text(tree, ...)` — visualize/print tree rules.
- `rf.feature_importances_` — MDI scores.
- `confusion_matrix`, `roc_curve`, `roc_auc_score` — classifier evaluation (Chapter 7).
- `train_test_split(X, y, test_size=0.3)` and `cross_val_score(model, X, y, cv=5)` — for honest evaluation and depth selection.

Practice 13: tree depth



Question

What happens as max_depth increases?

Why can train R^2 keep rising while test R^2 falls?

Practice 13 answer

Answer

Deeper trees memorize idiosyncrasies in the training data.

Train R^2 rises because the model fits noise; test R^2 falls because the learned splits do not generalize.

Ch 14: PCA

Main idea

PCA finds low-dimensional linear summaries that capture maximum variance.

Core methods

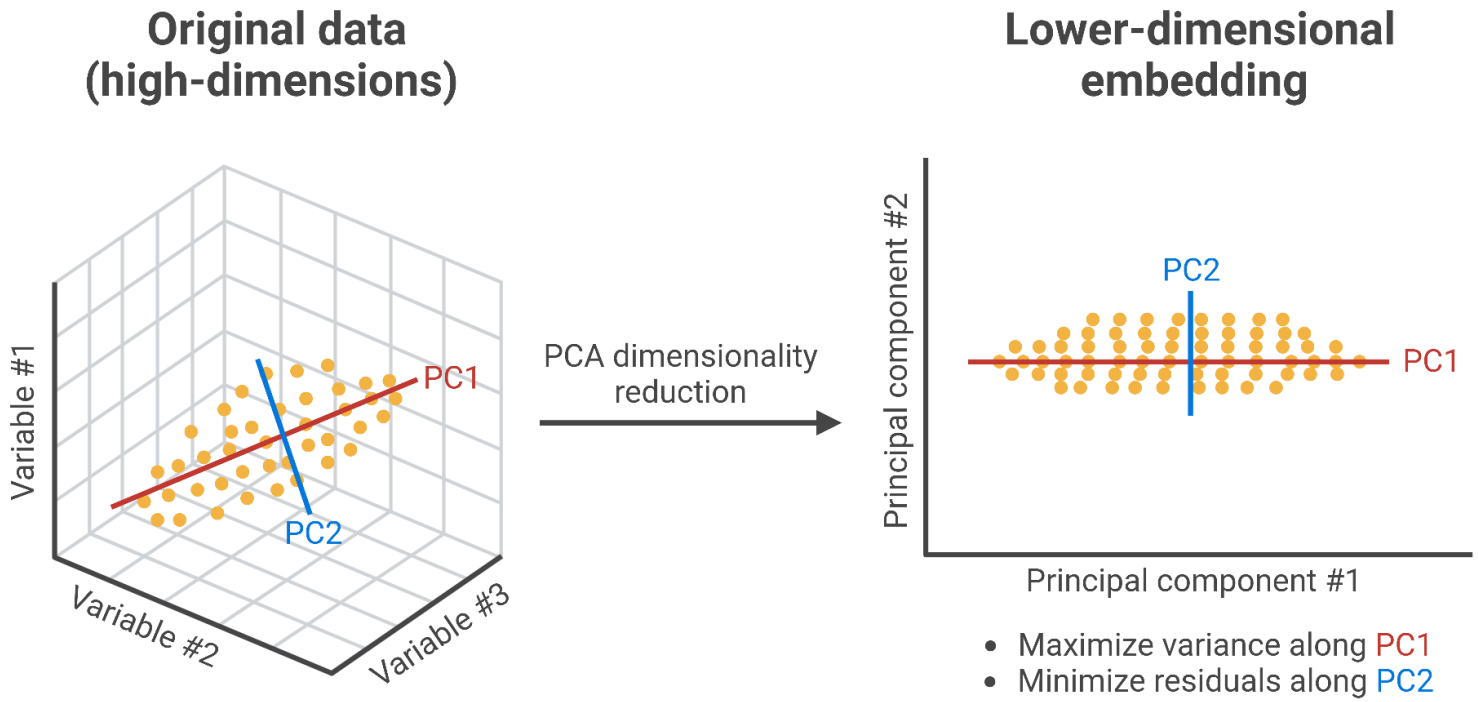
- Standardization,
- PC scores,
- loadings,
- explained variance,
- scree plots,
-
- SVD.

Keep in mind

Interpret loadings. PCA creates new combined features; it does not select original variables like lasso.

Ch 14: PCA

Principal Component Analysis (PCA) Transformation



- Maximize variance along PC1
- Minimize residuals along PC2

Ch 14: PCA

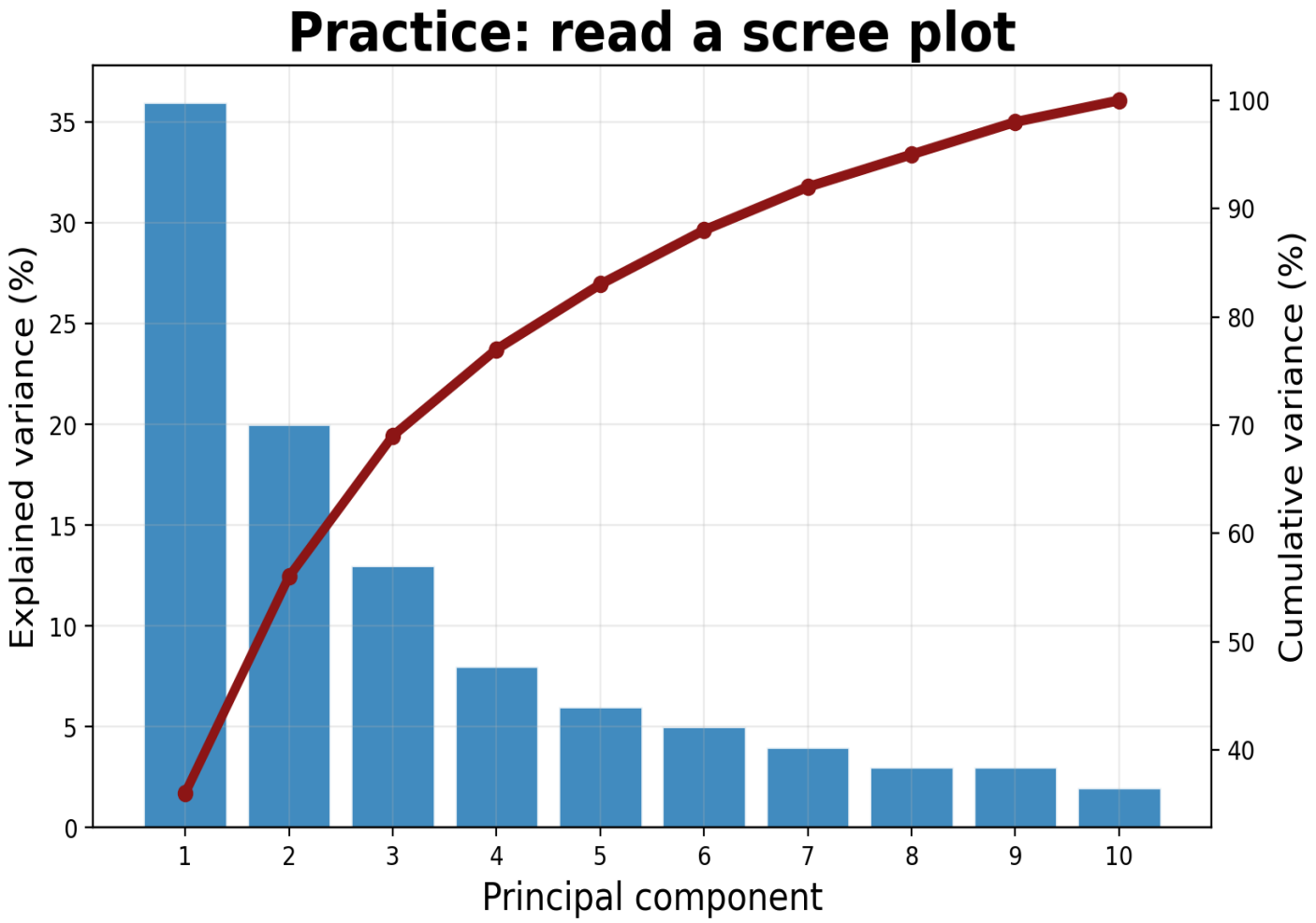
- What is PCA doing?
- How PCA differs from feature selection?
  - Name some cases PCA might have better interpretation than feature selections?
- How to interpret the PCs?
- How to choose how many PCs we need?
- Why we need to do standardization before doing PCA?

Ch 14: PCA

Computational tools

- `StandardScaler()` — standardizes features to mean 0 and variance 1
- `PCA()` — creates a PCA model; use `PCA(n_components=k)` to keep only k components
- `.fit_transform(X)` — fit the model and transform the data in one step
- `.explained_variance_ratio_` — array of variance fractions for each PC
- `.components_` — matrix of loadings (each row is a PC, each column is a feature)

Practice 14: scree plot



Question

How many PCs capture about 70% of variance?

Where is the elbow?

Why standardize first?

Practice 14 answer

Answer

About 4 PCs capture roughly 70% here.

The elbow is around PC3 or PC4.

Standardize first so PCA is not dominated by variables with large measurement units.

Ch 15: Clustering

Main idea

Clustering finds groups without labels, but the algorithm will always return groups.

Core methods

- K-means,
- centroids,
- SSE/inertia,
- elbow plot,
- initialization,
- ARI.

Keep in mind

Feature choice defines similarity. Validate clusters using external evidence or downstream use.

Ch 15: Clustering

- How to measure similarity between different units?
- What is supervised learning and unsupervised learning?
- What is K-means algorithm?

k-means clustering

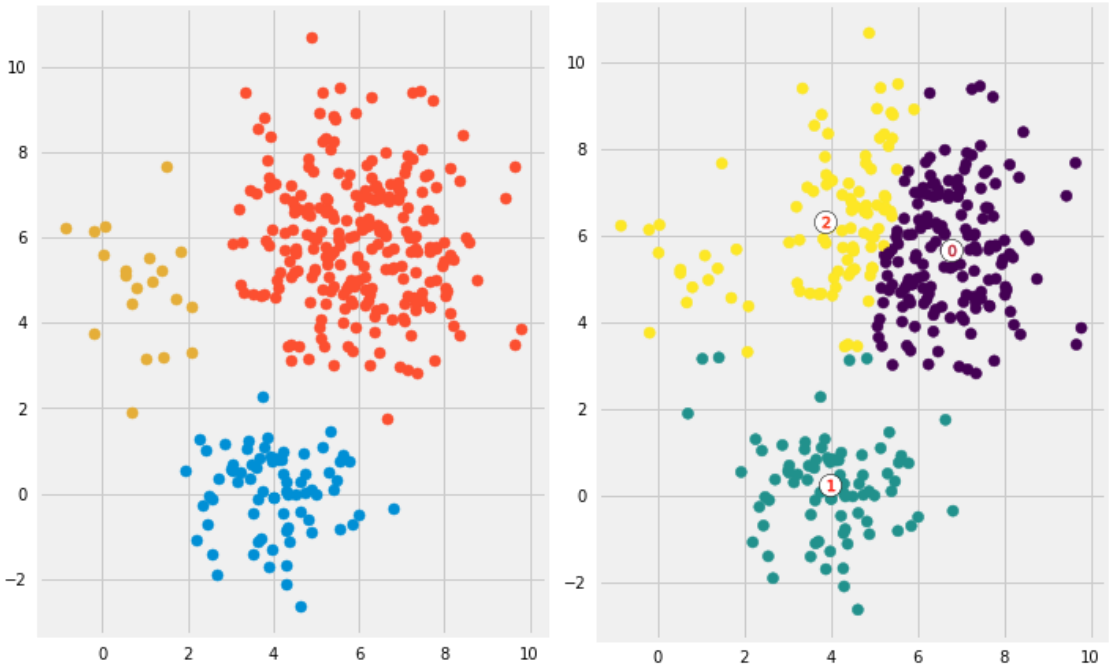
partition n points into k groups by repeating:

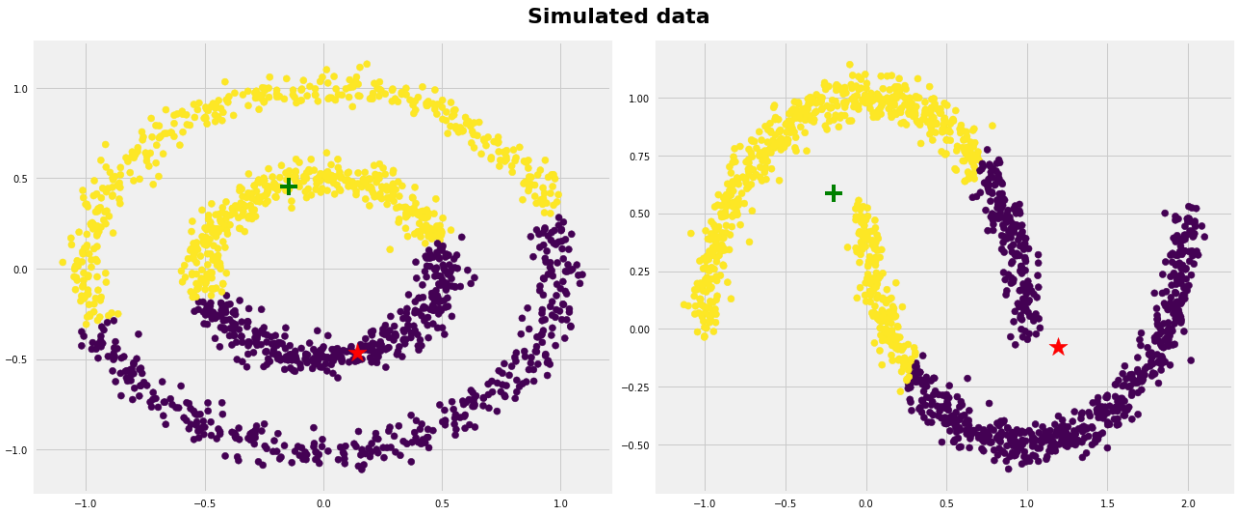
1. **initialize** k centroids (randomly)
2. **assign** each point to the nearest centroid
3. **recompute** each centroid as the mean of its assigned points

loop on 2-3 until assignments stop changing

Ch 15: Clustering

- K-means can be sensitive to initialization.
- Different choice of K can return very different clustering results.
- Your feature decide the clustering.
- When K-means fail:





Ch 15: Clustering

Computational tools

- `KMeans(n_clusters=k, random_state=42, n_init=10)` — fit k-means
- `km.fit_predict(X)` — fit and return cluster labels
- `km.inertia_` — sklearn's attribute name for the SSE
- `StandardScaler().fit_transform(X)` — standardize to zero mean, unit variance
- `silhouette_score(X, labels)` — average silhouette across all points
- `silhouette_samples(X, labels)` — per-point silhouette values for the per-point plot
- `adjusted_rand_score(labels1, labels2)` — compare two clusterings
- `scipy.cluster.hierarchy.linkage` and `dendrogram` — build and plot a hierarchical clustering

Ch 16: Validation beyond the random split

Main idea

Deployment changes the validation question: will this model work in the world where it is used?

Core methods

Temporal split, grouped split, walk-forward validation, leakage audit, distribution shift, feedback loops, Goodhart.

Keep in mind

Split along the axis of generalization. Historical accuracy may fail after intervention or policy change.

Ch 16: Validation beyond the random split

1. **Temporal leakage.** Data has time structure. A random split puts neighbors of the test point in the training set. *Fix: train on the past, test on the future.*
2. **Distribution shift.** Deployment data comes from a regime the training set never saw. *Fix: test on out-of-distribution data, stratify by subgroup, widen intervals at the tails.*
3. **Feedback loops.** Predictions change the outcome. *No passive validation catches this; you need randomized holdouts or A/B tests.*
4. **Goodhart's law.** Once a metric drives consequences, the measured agents respond. Those best at responding move the metric without advancing the goal. *Structural, not a split problem.*