Global Models: Linear and Logistic Regression

How features become numerical predictions, probabilities, and evidence

Fatih Kansoy

Oxford
fatih.ai/machinelearning

2026

Day 2 outline

1. One global score, two prediction tasks
2. Linear regression: features become a number
3. Feature design determines model shape
4. Regularisation controls the fitted rule
5. Logistic regression: score becomes probability
6. What global models can legitimately claim

Part 1

One global score, two prediction tasks

We now open the fitted rule and follow one encoded record through its score, output, and evidence.

The first question: what price for a London listing?

You own a two-bedroom flat in Hackney, in east London, and you list it on Airbnb for the first time. The site asks for one number: the price for one night.

- Ask too much, and the calendar stays empty.
- Ask too little, and you give money away on every booked night.

What would you actually do?

Today we turn it into a model: a rule that reads a listing's features and returns a price.

> The same question at a larger scale: the platform suggests prices to thousands of new hosts, and a lettings analyst values a whole portfolio. Nobody can browse listings by hand at that volume. They need a rule.

A recorded case must become a numerical representation

Recorded London listing	Entries supplied to the model
capacity: 4 guests	$z_1 = 4$
bedrooms: 2	$z_2 = 2$
room type: private room	$z_3 = 1$, with entire home as the reference
neighbourhood: Hackney	one Hackney indicator among the represented areas
minimum stay: 2 nights	$z_4 = 2$

$$\underbrace{\mathbf{x}}_{\text{recorded case}} \xrightarrow{\phi} \underbrace{\mathbf{z} = \phi(\mathbf{x})}_{\text{model representation}} .$$

ϕ = feature map from permitted record $\mathbf{x}$ to model inputs $\mathbf{z}$; any data-dependent parts are fitted on development data and frozen before validation or audit.

The feature map can encode categories, transformations, interactions, and missingness rules. It is part of the fitted model.

Linear and logistic models begin with a global additive score

$$\eta(\mathbf{x}) = \beta_0 + \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\beta} = \beta_0 + \sum_{j=1}^{p} \beta_j \phi_j(\mathbf{x})$$

p = number of represented feature columns; j indexes one such column.

Intercept β_0 — the reference score before the represented feature contributions are added.

Feature value $\phi_j(\mathbf{x})$ — what this case contributes in column (j) after encoding.

Coefficient β_j — the fitted weight attached to that represented column.

"Global" means the same fitted coefficient rule is applied to every represented case.

A1 · Matrix expression →

The output rule determines what the score means

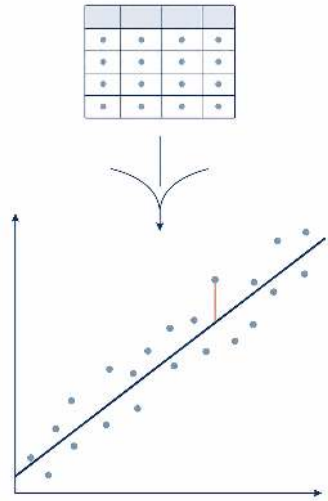
	Linear regression	Logistic regression
Target	numerical Y	binary $Y \in \{0, 1\}$
Shared score	$\eta(\mathbf{x}) = \beta_0 + \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\beta}$	
Output	$\widehat{y} = \eta(\mathbf{x})$	$\widehat{p} = \sigma(\eta(\mathbf{x})) = 1/(1 + e^{-\eta(\mathbf{x})})$
Fitting loss	squared error	binary log loss

The representation and additive score survive; the output map and fitting loss change.

Part 2

Linear regression: features become a number

A baseline becomes a line, the line becomes a multi-feature score, and unseen listings decide whether its comparisons travel.



The London task predicts a quoted nightly listing price

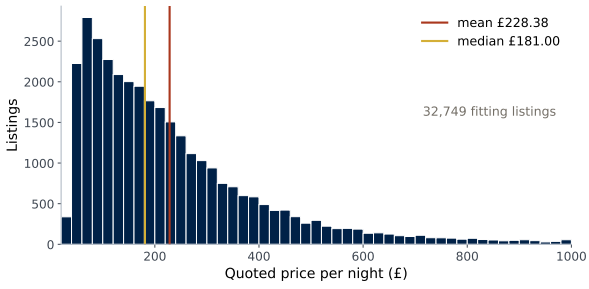
Unit	one eligible London short-stay listing
Target	quoted price per night, in pounds, for a recorded 1–7-night quote
Information	capacity, bedrooms, beds, minimum stay, room type and represented neighbourhood
Development role	32,749 listings from 16,381 hosts fit candidate workflows
Validation role	11,063 listings from different hosts compare candidates
Final audit	10,824 listings remain sealed through the Day 3 family comparison

Host-disjoint = no host appears in more than one role; the final audit remains unused until the Day 3 comparison.

Source: Inside Airbnb London snapshot, 19 June 2026; frozen eligibility and host-disjoint split reproduced from the course data pipeline.

A28 · London sample and split provenance →

A no-feature prediction establishes the benchmark



Squared-error comparator

$$\widehat{y}_i = \bar{y}_{\text{dev}} = £228.38.$$

dev = development listings only; $\bar{y}_{\text{dev}}$ = their average quoted nightly price.

Every validation listing receives the same frozen development mean.

Absolute-error comparator

$$\widehat{y}_i = \text{median}(y_{\text{dev}}) = £181.$$

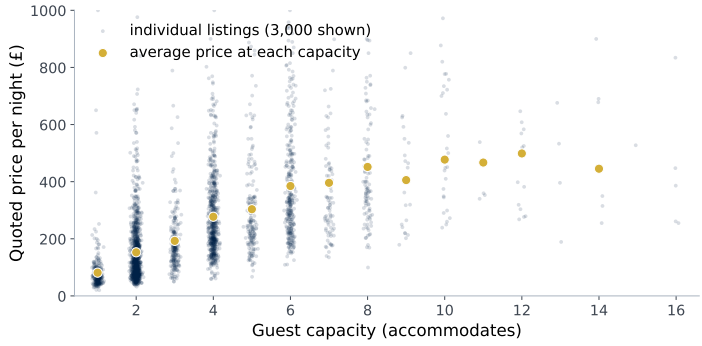
A baseline must match the loss or metric being discussed.

Inputs earn their place only by improving relevant new-listing evidence over a credible no-feature rule.

A4 · Why squared loss targets the mean →

Using what you know: guest capacity

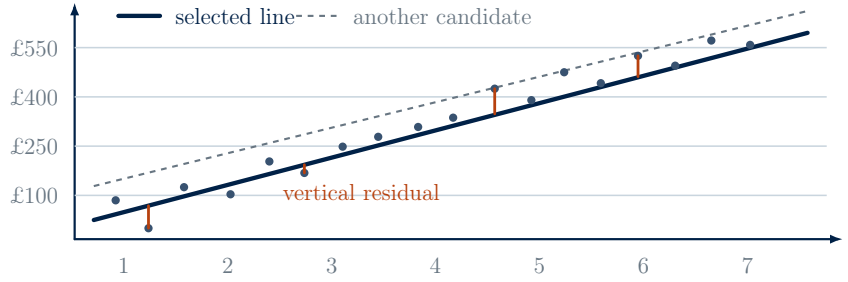
The one-number model ignores everything we know about a listing. The most obvious input to add: how many guests it sleeps.



$$\widehat{y} = \beta_0 + \beta_1 c.$$

- Average price rises with capacity: about £81 at one guest, £277 at four, £452 at eight.
- Above eight guests the averages flatten and listings become scarce.

Least squares selects coefficients, not a visual impression



$$(\widehat{\beta}_0, \widehat{\beta}_1) = \arg\min_{\beta_0,\beta_1} \frac{1}{n} \sum_{i=1}^{n} \big(y_i - \beta_0 - \beta_1 c_i\big)^2, \qquad c_i = \text{capacity}_i - 4.$$

arg min = value that minimises the displayed objective; n = number of development listings.

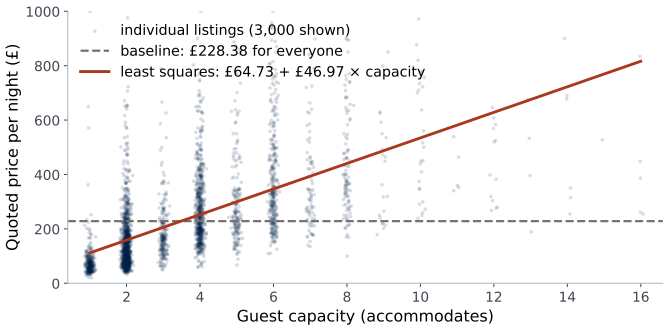
The objective adds squared vertical residuals across development listings. The selected coefficients minimise that total; the fitted line is therefore a rule chosen by a declared loss, not by visual judgement.

A2 · Least-squares conditions → **A3 · Closed-form line →**

The fitted line for London

On the 32,749 fitting listings, least squares returns:

$$\widehat{\text{price}} = £64.73 + £46.97 \times \text{capacity}.$$



- Among all straight lines in capacity, this one has the smallest total squared residual. From here on, call it **the capacity line**.

Interpreting the slope and the intercept

> Among the listings in this snapshot, listings that sleep one more guest are quoted, on average, £46.97 more per night.

Three cautions before using that sentence:

- **It compares different listings.** It does not say your price rises by £46.97 if you add a sofa bed. Capacity also stands in for size, room type, and location; Part III separates those.
- **It is an average along the line.** Individual listings sit far above and below it, as the scatter showed.
- **The intercept is not a market price.** £64.73 is the line's value at zero guests; no such listing exists. It anchors the line, nothing more.

Measuring the misses: MAE and RMSE

Collect the misses $e_i = y_i - \widehat{y}_i$ and summarise them in the two standard ways:

$$\text{MAE} = \frac{1}{n}\sum_i |e_i|, \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_i e_i^2}.$$

MAE, mean absolute error · RMSE, root mean squared error · both are in pounds

Three misses: $+£40$, $-£100$, $+£10$. MAE $= 150/3 = £50$; RMSE $= \sqrt{11{,}700/3} \approx £62$, larger because the £100 miss weighs more once squared.

Measure	Reads as	The right one when
MAE	the average size of a miss	every pound of miss hurts equally
RMSE	a typical miss, large misses weighted more	big misses hurt disproportionately; matches the loss we fit by

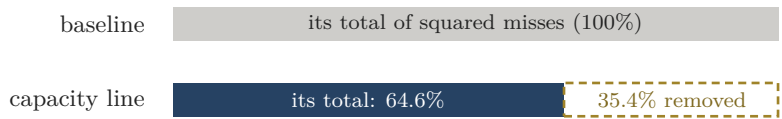
SSE fits the line; MAE and RMSE tell us, in pounds, how wrong its predictions are.

R^2: the share of the baseline error removed

Is a typical miss of £137 good? Alone, hard to say. The natural yardstick is Part II's baseline: £228.38 for everyone. On the 11,063 checking listings:

- add up all squared misses, once for the baseline (a large total) and once for the capacity line (smaller, if it helps);
- report the share of the baseline's total that disappeared.

$$R^2 = 1 - \frac{\text{the line's total of squared misses}}{\text{the baseline's total of squared misses}}.$$



Here $R^2 = 0.354$: 35.4% of the baseline's squared error removed.

A11 · Training fit and predictive R^2 →

How to read predictive R^2

Value	Reading
1	perfect prediction: the model's SSE is zero
0.354	the model removes 35.4% of the baseline's squared error
0	no better than quoting the mean for everyone
below 0	worse than the baseline; possible on unseen data

- Higher is better only when models are scored on the same outcome, the same listings and the same baseline.
- A negative value is possible: the median baseline has $R^2 = -0.050$ because it produces more squared error than the mean.
- R^2 has no units and is not a grade. Report it with MAE or RMSE to show the remaining error in pounds.

RMSE tells us how large the misses are; R^2 tells us how much of the baseline's squared error the model removed.

Does capacity help on unseen hosts?

The constants and line were estimated using 32,749 fitting listings. We now freeze them and score their predictions on 11,063 listings belonging to different hosts.

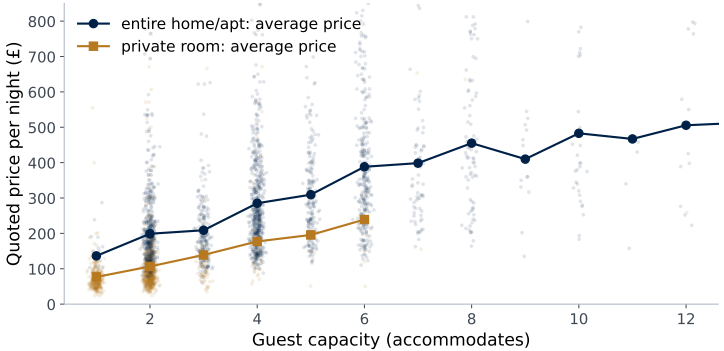
Model	MAE	RMSE	Predictive R^2
Mean baseline: £228.38	£129.00	£170.73	0.000
Median baseline: £181.00	£120.74	£174.95	−0.050
Capacity line	**£96.01**	**£137.27**	0.354

- Capacity reduces MAE from £129 to £96 and RMSE from £171 to £137 relative to the mean baseline.
- Its $R^2 = 0.354$: one input removes 35.4% of the mean baseline's squared error.

Part II: one input beats no inputs. Part III asks what we gain by adding room type, bedrooms, beds, minimum nights and borough.

Same capacity, different prices

The first candidate to add: **room type**. At the same capacity, is a whole flat quoted like a private room in someone's home?



- **No**: at four guests, entire homes average about £108 more per night than private rooms.
- The capacity line cannot see this. It quotes both the same, so part of its miss is built in.

Fitting listings; averages shown at capacities with at least 30 listings in the group.

How room type enters the model

Room type is categorical. Represent it with 0/1 indicators and leave one category, entire home/apartment, as the **reference**:

Room type of a listing	private?	shared?	hotel?
entire home/apt (the reference)	0	0	0
private room	1	0	0
shared room	0	1	0
hotel room	0	0	1

The two-input model is

$$\widehat{y} = \beta_0 + \beta_1 \text{capacity} + \gamma_P I(\text{private}) + \gamma_S I(\text{shared}) + \gamma_H I(\text{hotel}).$$

for an entire home all three indicators equal zero · each γ compares that room type with an entire home at the same capacity

The indicators let one regression give different predicted price levels to different room types.

What changes after adding room type?

Fit the model on the same 32,749 listings, then score it on the same 11,063 checking listings:

	Capacity only	Capacity + room type
Capacity coefficient	£46.97	£36.60
Private room vs entire home	—	−£98.49
Checking R^2	0.354	0.433

- At the same capacity, a private room is predicted about £98 below an entire home.
- Capacity's coefficient falls because the one-input model used capacity partly to represent differences in room type.

Adding an input can improve prediction and change how an existing coefficient should be read.

From two inputs to the compact model

Multiple regression extends the same rule:

$$\widehat{y} \;=\; \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k.$$

$x_1, \ldots, x_k$: the inputs · β_j: predicted difference for a one-unit change in input j, with the represented values of other inputs fixed

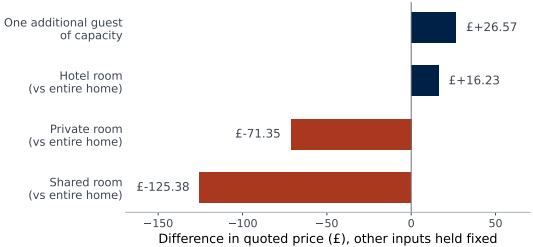
Our **compact model** uses all six pre-specified inputs:

Numerical	capacity, bedrooms, beds, minimum nights
Categorical	room type; borough (33 areas, one reference)
Model size	39 slope coefficients plus the intercept

The compact model uses more information; whether that information helps must be decided on the checking listings.

What the compact coefficients say

The figure shows selected coefficients from the model fitted on the same 32,749 listings:



- Capacity: £46.97 alone, £36.60 after room type, and £26.57 after all six inputs are represented.
- With the other represented inputs fixed, private rooms are predicted about £71 below entire homes; shared rooms about £125 below.

Compact model on the fitting listings; reference category: entire home/apt.

Each coefficient is a conditional comparison within the model, not the causal effect of changing a listing.

Do added inputs help on unseen hosts?

On the same 11,063 checking listings:

Model	MAE	RMSE	Predictive R^2
One number: the mean	£129.00	£170.73	0.000
One input: capacity	£96.01	£137.27	0.354
Two inputs: capacity, room type	£85.51	£128.60	0.433
Six inputs: the compact model	£75.48	£114.63	0.549

- Each step lowers MAE and RMSE: £129 → £96 → £86 → £75 in MAE.
- The compact model reaches $R^2 = 0.549$, removing 54.9% of the mean baseline's squared error.

More information helps, but an average miss of £75 remains. The next diagnostics ask what structure the linear model still misses.

Adjusted R^2: a guard when counting inputs

One caution before adding inputs freely: on the *fitting* data, R^2 can never fall when an input joins. Least squares will use anything, even noise, at least a little.

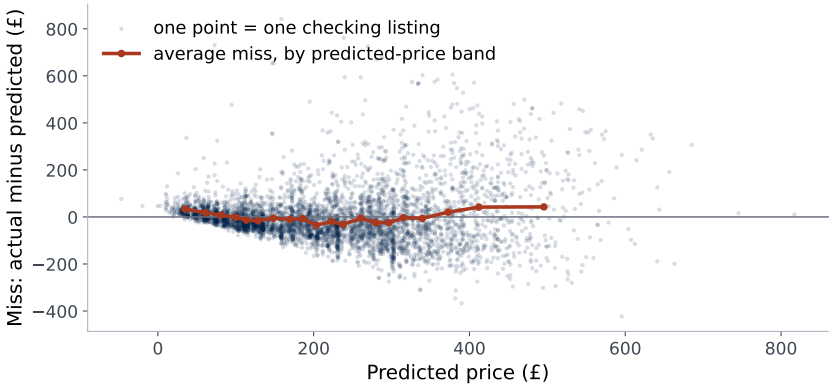
Model	Fitting R^2	Adjusted R^2
Capacity	0.350	0.350
Capacity + room type	0.405	0.405
Compact model: six inputs (39 columns)	0.528	0.527

- **Adjusted R^2** charges a small fee per input, so clutter stops looking free. With 32,749 listings the fee is barely visible; in small samples it bites hard (worked example in the appendix).
- Here the adjusted value is almost unchanged: the fitting sample is large relative to the 39 model columns.

Use adjusted R^2 to compare complexity on the fitting data; use the unseen-host results from the previous slide to judge prediction.

What the misses still show

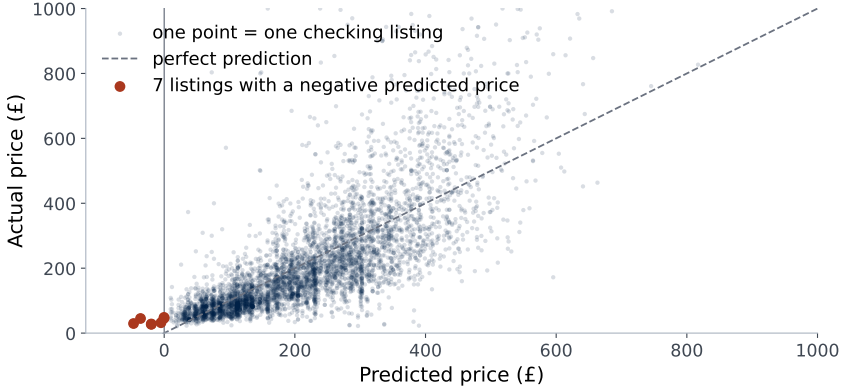
Plot each of the compact model's checking misses against its predicted price. If the model captured every pattern, the red average would sit near zero everywhere.



- Instead it bends: too low at the cheap end (+£34), slightly too high around £200 to £300 (−£20 to −£35), too low again at the top (+£43).

Impossible prices

Each point is one checking listing (5,000 shown): predicted price across, actual price up; the dashed diagonal is a perfect prediction.



Seven red points lie left of £0: linear predictions have no lower bound and can produce impossible negative prices.

Part 3

How sure are we about a coefficient in the model?

A coefficient in the model is only as certain as the data that went into it.

Would another sample give the same slope?

To make the issue visible, return to the one-input capacity line fitted on 32,749 listings:

$$\widehat{\text{price}} = £64.73 + £46.97 \times \text{capacity}, \qquad \widehat{\beta}_1 = £46.97.$$

This is the slope from one observed sample, not a fixed market constant.

A different market snapshot, or another random set of listings from the same market, would contain different capacity–price pairs:

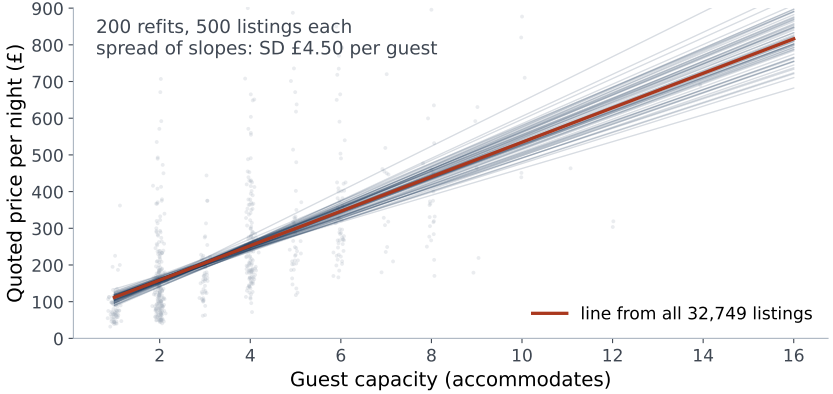
$$\text{different listings} \implies \text{different fitted line} \implies \text{different } \widehat{\beta}_1.$$

Sampling uncertainty is how much an estimate such as $\widehat{\beta}_1$ would vary across comparable samples.

Next: fit the same line to 200 random samples and watch $\widehat{\beta}_1$ move.

The experiment: refit on 500 random listings

200 repetitions: **1.** draw 500 listings; **2.** refit the line; **3.** record the slope.

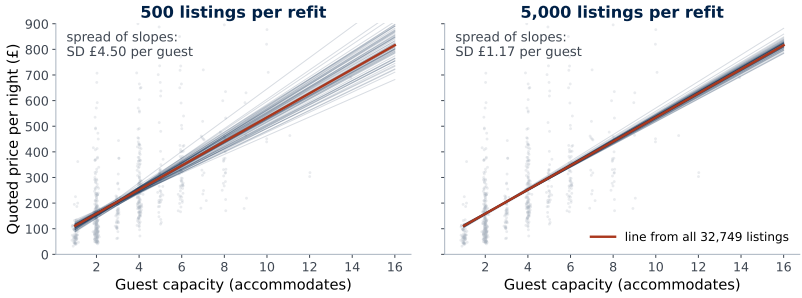


Different samples produce different lines. Across 200 refits, most slopes are about £38–£56 per additional guest; their standard deviation (SD) is £4.50.

60 of the 200 refitted lines shown; grey points show the price–capacity cloud for context.

More listings per refit, less wobble

Run the identical experiment once more, changing one thing only: each refit now uses 5,000 listings instead of 500.



- The fan tightens: the spread of slopes falls from SD £4.50 to £1.17. Ten times the data, roughly a third of the wobble.
- Our real fit uses all 32,749 listings, so its wobble must be smaller still. How small is the next question.

A19 · OLS variance and standard errors →

Same recipe, seed, and display as the previous slide; 60 of 200 refits per panel.

Standard error: estimate slope wobble from one sample

The last two slides estimated slope uncertainty from the SD of 200 refits. In practice we usually observe one sample and one slope.

> The **standard error** (SE) of $\widehat{\beta}_1$ uses one sample to estimate the SD of $\widehat{\beta}_1$ across comparable samples.

$$\widehat{\mathrm{SE}}(\widehat{\beta}_1) = \frac{\widehat{\sigma}}{\sqrt{\sum_i (x_i - \bar{x})^2}} \approx \frac{\widehat{\sigma}}{s_x \sqrt{n}}.$$

$\widehat{\sigma}$: residual price standard deviation · s_x: capacity spread (sample standard deviation) · larger n or s_x lowers the SE

London: $\widehat{\beta}_1 = £46.97$. The textbook SE is £0.35; the robust SE is £0.52. Both are small relative to the slope.

SE concerns the coefficient, not one listing's price. Next: estimate ± 1.96SE gives the large-sample 95% interval.

A19 · OLS variance and standard errors →

A 95% confidence interval for the slope

A 95% interval is the estimate ± a critical value times its SE:

$$\widehat{\beta}_1 \;\pm\; t_{0.975,\,n-2}\,\mathrm{SE}(\widehat{\beta}_1), \qquad t_{0.975,\,32747} \approx 1.96.$$

2 is a convenient verbal approximation; 1.96 is the large-sample 95% critical value

$$\begin{aligned} \text{margin of error} &= 1.96(0.3538) = 0.6934, \\ 95\%\ \text{CI} &= 46.9679 \pm 0.6934 = [£46.27,\ £47.66] \text{ per additional guest.} \end{aligned}$$

- Across repeated comparable samples, about 95% of intervals built this way contain the fixed true slope.
- It is an interval for the average capacity–price relationship, not for the price of an individual listing.

Next: measure the estimate's distance from zero in standard-error units.

The t statistic: distance from the null in SE units

In the simple model, test whether mean price changes linearly with capacity. The null hypothesis says it does not:

$$H_0 : \beta_1 = 0, \qquad t = \frac{\widehat{\beta}_1 - 0}{\mathrm{SE}(\widehat{\beta}_1)} = \frac{46.9679}{0.3538} = 132.8.$$

the sign gives the direction · $|t|$ gives the distance from the null in SE units

$|t| = 132.8$ is far beyond 1.96: the data are strongly inconsistent with $H_0 : \beta_1 = 0$. This is evidence of a nonzero association, not causality or predictive accuracy.

The p-value: how surprising under H_0?

The t statistic is a distance; the **p-value** is its tail probability under $H_0 : \beta_1 = 0$:

$$p = \Pr_{H_0}(|T| \geq |t_{\text{observed}}|) \,.$$

two-sided test: count equally extreme negative and positive t statistics

$\lvert t \rvert$	Two-sided p	Under H_0
1.96	0.050	at the conventional 5% boundary
3.00	0.0027	unusual
132.8	< 0.001	extremely inconsistent with H_0

A small p means the observed distance would be unusual under H_0, so it is evidence against that null. It is not $\Pr(H_0 \text{ is true})$, and does not measure effect size, causality, or prediction.

For capacity, $p < 0.001$. Software may display 0.000, but the probability is not literally zero.

A19 · OLS variance and standard errors →

Reading a software summary

Read the capacity row from left to right: **estimate → uncertainty → test of zero**.

	coefficient	std. error	t	p	95% interval
intercept	64.73	1.45	44.6	< 0.001	61.89 to 67.58
capacity	**46.97**	**0.35**	**132.8**	< 0.001	**46.27 to 47.66**

1. **Size and direction:** one additional guest is associated with £46.97 higher predicted price.
2. **Precision:** SE = £0.35; the textbook 95% interval is £46.27–£47.66.
3. **Test of zero:** $t = 132.8$ and $p < 0.001$; the slope is strongly inconsistent with zero.

Read the coefficient and interval before the p-value: statistical evidence does not tell you the size of the relationship.

Significant and important are different

With 32,749 listings, even a small relationship can have a small standard error and therefore a tiny p-value. Ask four separate questions:

1. **Different from zero?** Use t and p: here $p < 0.001$.
2. **Large enough to matter?** Use the coefficient and its units: about £47 per additional guest.
3. **Useful for prediction?** Use checking performance: MAE £96.01 and $R^2 = 0.354$ for the capacity line.
4. **Causal?** That requires a research design; this observational regression establishes only an association.

A tiny p-value answers only the first question.

A precise slope does not imply precise prices

Coefficient uncertainty and prediction error concern different objects:

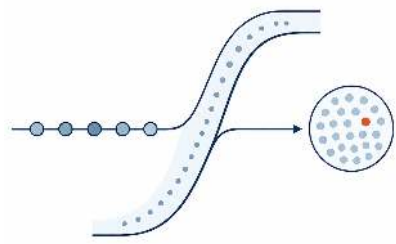
	Coefficient uncertainty	Prediction error
Object	population slope β_1	price of one new listing
Measure	SE and confidence interval	checking MAE and RMSE
London	SE £0.35; 95% CI [£46.27, £47.66]	MAE £96.01; RMSE £137.27
More data	usually narrows the interval	cannot remove variation across listings

A precise average relationship does not guarantee an accurate individual prediction. Next, Part V changes the target from a price to a probability.

Part 4

Logistic regression: score becomes probability

The additive score and regularisation survive. A bounded output map and a probability-aware loss rebuild the same learning grammar for a binary target.



The bank data, reintroduced

Back to the second question. The Portuguese bank sells term deposits by telephone; most clients say no. Three of the 41,188 recorded client contacts:

Client	Age	Job	Previous campaign	**Duration**	**Subscribed** (y)
A	31	admin	never contacted	96 sec	no
B	47	technician	success	420 sec	yes
C	55	retired	failure	185 sec	no

- Target: $y = 1$ if the client subscribed; 4,640 of 41,188 did (11.27%).

> The question, stated precisely: using only what the bank knows *before* a call, estimate the probability that this client subscribes.

So the first thing to settle is what the bank actually knows before a call.

The Bank task predicts response before the client outcome is known

Unit	one recorded client contact in the UCI Bank Marketing file
Target	whether that contact ended in a term-deposit subscription
Prediction moment	after the planned contact channel, month, weekday, and current attempt count are known, but before the call outcome is observed
Forbidden field	call duration, because it is only known after the call has happened
No-prior-contact marker	$pdays = 999$ means not previously contacted; represent it as an indicator plus elapsed days, not as 999 days

This is a response-prediction benchmark, not evidence that contacting the client causes subscription.

The random row split preserves Day 1 continuity. With no stable client identifier or exact date, it is not a deployment-matched final audit.

A29 · Bank provenance and prediction clock →

For a yes–no target, the conditional mean is a probability

Let $Y = 1$ for subscription and $Y = 0$ otherwise. Then

$$\mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}] = 1 \cdot \Pr(Y = 1 \mid \mathbf{x}) + 0 \cdot \Pr(Y = 0 \mid \mathbf{x}) = \Pr(Y = 1 \mid \mathbf{x}).$$

The no-feature probability model is therefore the fitting-sample response rate:

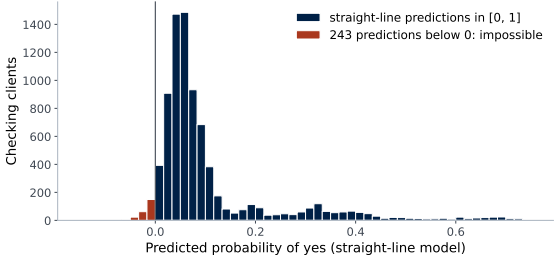
$$\widehat{p}_i = \bar{y}_{\text{fit}} = \frac{2,784}{24,712} = 0.1127.$$

The model's job is not merely to reproduce 11.27% overall. It is to assign different, defensible probabilities to different represented records and improve held-out probability loss.

A21 · Binary means are event probabilities →

A straight probability model can leave the probability scale

$$\widehat{p} = \beta_0 + \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\beta} \qquad \text{while a probability must satisfy} \qquad 0 \le p \le 1.$$



Corrected validation output	Records	Boundary fact
inside $[0, 1]$	7,996	maximum 0.890
below 0	242	minimum -0.078
above 1	0	—

- It gives 242 clients a *negative* predicted probability: mathematically possible for a line, impossible for a probability.
- It also imposes the same probability change per unit of input, whether the client starts near 0.02 or near 0.50.

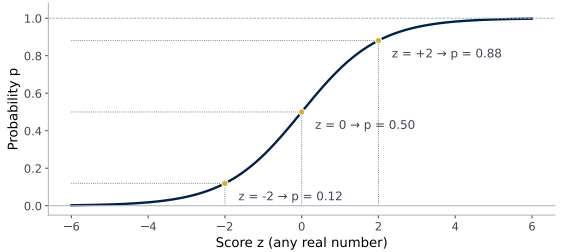
Same Bank split and corrected *pdays* sentinel representation as the logistic benchmark; 8,238 validation contacts.

From a linear index to a valid probability

Keep the familiar linear combination of the inputs and call it the **linear index** z. Then transform it with the logistic function:

$$z = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k, \qquad p = \frac{1}{1 + e^{-z}}.$$

z: any real number · p: always strictly between 0 and 1 · this is **logistic regression**



The S-curve keeps $0 < p < 1$ and preserves the ranking: higher z, higher p.

Probability changes most near 50%

The logistic transformation is steepest in the middle and flatter near the two probability limits:

linear index z	-4	-2	0	$+2$	$+4$
probability p	0.018	0.119	0.500	0.881	0.982

$$\frac{\partial p}{\partial z} = p(1-p): \qquad 0.25 \text{ at } p = 0.50, \quad \longrightarrow 0 \text{ near } p = 0 \text{ or } 1.$$

A $+2$ move from $z = 0$ raises p by 0.381; the same move from $z = 2$ raises it by only 0.101. Thus a coefficient changes z by a fixed amount, but its probability effect depends on the client's starting probability.

We now have valid client-specific probabilities. Next: how should we score them?

From a probability claim to a penalty

A claim of p for "yes" also assigns $1 - p$ to "no." Score the probability assigned to the outcome that actually happened:

$$q = \begin{cases} p, & \text{if the client said yes,} \\ 1 - p, & \text{if the client said no,} \end{cases} \qquad \text{penalty} = -\ln(q).$$

q: probability assigned to the observed outcome; ln: natural logarithm

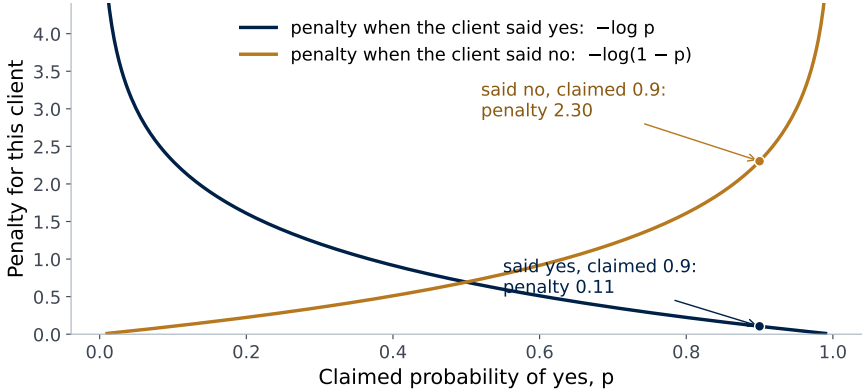
Suppose the model claims $p = 0.90$:

What happened?	Probability given to it	Calculation	Penalty
Client said yes	$q = 0.90$	$-\ln(0.90)$	$0.105 \approx 0.11$
Client said no	$q = 0.10$	$-\ln(0.10)$	$2.303 \approx 2.30$
Either, if $p = 0.50$	$q = 0.50$	$-\ln(0.50)$	$0.693 \approx 0.69$

Why $-\ln(q)$? A probability of 1 costs 0; as q approaches 0, the penalty grows without bound.

Scoring a probability claim

The figure repeats the same calculation for every possible claimed probability of yes, p:



If the client says yes, moving p toward 1 lowers the penalty. If the client says no, it raises the penalty. Confident mistakes are therefore expensive.

Log loss: average the clients' penalties

The previous figure scored one client. To score the whole model, average that penalty over all checking clients:

$$\text{log loss} = \frac{1}{n}\sum_{i=1}^{n} -\ln(q_i), \qquad \text{lower is better.}$$

- Each client contributes one penalty, based on the probability assigned to that client's observed outcome.
- The average gives one score for the model. It has no units and is compared across models on the same checking clients.

Lower log loss means the model assigned more probability to what actually happened. Next, compute the baseline score.

The baseline sets the score to beat

Before using the 19 inputs, predict the fitting response rate for everyone: $p = 2{,}784/24{,}712 = 0.1127$. Score that flat rule on the checking clients (no client-level information):

Checking outcome	Clients	q: probability of outcome	Penalty each
Said yes	928	0.1127	$-\ln(0.1127) = 2.1834$
Said no	7,310	0.8873	$-\ln(0.8873) = 0.1195$

$$\text{baseline log loss} = \frac{928(2.1834) + 7{,}310(0.1195)}{8{,}238} = 0.3520.$$

Bank data split: 24,712 fitting contacts (2,784 yes) set $p = 0.1127$; 8,238 checking contacts (928 yes, 7,310 no) are scored with that frozen p. The two counts are not interchangeable.

The 0.352 is a benchmark, not a probability. A candidate model is useful only if its checking log loss is lower.

How we will judge a probability model

The discipline from the price model is unchanged: learn from fitting data, judge on checking data, and compare with a simple baseline. Only the score changes from MAE/RMSE to log loss.

1. Fit a candidate model on the 24,712 fitting clients.
2. Use it to predict p_i for the 8,238 checking clients.
3. Compute their average log loss and compare it with 0.352.

Candidate checking log loss	What it tells us
Below 0.352	the inputs improve the probability predictions
0.352 or above	the inputs add no predictive value beyond the baseline

The scorecard is fixed before fitting. Next: choose the S-curve's coefficients on the fitting clients.

Did the model beat the baseline?

On the 8,238 checking clients, the score to beat was 0.352:

Model	Checking log loss
Baseline: 0.113 for everyone	0.352
The bank model (19 inputs)	0.272

- The model's probability claims sit far closer to what happened than the flat 0.113: 0.352 falls to 0.272.
- Adding the forbidden `duration` would look better still, and mean nothing: Day 1's leakage lesson stands.

This is the same model whose 0.50 labels Day 1 judged. One question remains before using its probabilities: do the claims match reality?

Part 5

Appendix

Technical Appendix

Technical appendix: map

Linear fitting	A1 design matrix · A2 first-order conditions · A3 closed-form line · A4 conditional mean · A5 partial comparison · A6 reference coding
Representation	A7 feature maps · A8 transformations · A9 interactions · A10 fitted pipelines · A11 training fit and predictive R^2 · A12 support · A13 collinearity
Complexity / uncertainty	A14 ridge · A15 lasso · A16 scaling · A17 tuning · A18 stability · A19 OLS variance · A20 mean and prediction intervals
Binary probability	A21 binary mean and LPM · A22 logit/sigmoid · A23 likelihood · A24 optimisation · A25 separation · A26 probability changes · A27 calibration
Audit ledgers	A28 London · A29 Bank
Worked numbers	A30 typical miss · A31 R^2 computed · A32–A33 adjusted R^2 · A34 slope SE · A35 the $\sqrt{n}$ law · A36 t and p

Every appendix button returns to the concept that called it. These derivations support the main narrative; they are not an additional lecture sequence.

Appendix A1: feature map and design matrix

For case i, let

$$\mathbf{z}_i = \boldsymbol{\phi}(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \ldots, \phi_p(\mathbf{x}_i))^\top.$$

p = number of represented features; a tilde marks an object augmented with the intercept.

Stack an intercept and all represented cases:

$$\widetilde{\boldsymbol{\Phi}} = \begin{bmatrix} 1 & \mathbf{z}_1^\top \\ \vdots & \vdots \\ 1 & \mathbf{z}_n^\top \end{bmatrix}, \qquad \widetilde{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix}, \qquad \widehat{\mathbf{y}} = \widetilde{\boldsymbol{\Phi}}\widetilde{\boldsymbol{\beta}}.$$

The ith row describes one case; the jth column is one model feature; the jth coefficient is shared across all rows. Encoding and transformations therefore change $\widetilde{\boldsymbol{\Phi}}$, hence the fitted model.

← **Return to the represented London record**

Appendix A2: least-squares first-order conditions

With centred capacity c_i, minimise

$$Q(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 c_i)^2.$$

At an interior minimum,

$$\frac{\partial Q}{\partial \beta_0} = -2 \sum_i (y_i - \beta_0 - \beta_1 c_i) = 0,$$

$$\frac{\partial Q}{\partial \beta_1} = -2 \sum_i c_i (y_i - \beta_0 - \beta_1 c_i) = 0.$$

Writing fitted residuals as $\widehat{e}_i = y_i - \widehat{y}_i$, these imply

$$\sum_i \widehat{e}_i = 0, \qquad \sum_i c_i \widehat{e}_i = 0.$$

Thus OLS residuals are orthogonal to the intercept and each fitted design column. This is a sample fitting identity, not proof that residuals are independent, homoscedastic, or causally unconfounded.

← **Return to the least-squares objective**

Appendix A3: closed-form slope and intercept

For one numerical input,

$$\widehat{\beta}_1 = \frac{\sum_i (c_i - \bar{c})(y_i - \bar{y})}{\sum_i (c_i - \bar{c})^2} = \frac{\widehat{\mathrm{Cov}}(c, y)}{\widehat{\mathrm{Var}}(c)}, \qquad \widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{c}.$$

$\bar{c}$ and $\bar{y}$ = sample means; $\widehat{\mathrm{Cov}}(c, y)$ = sample covariance; $\widehat{\mathrm{Var}}(c)$ = sample variance.

If $c = \text{capacity} - 4$, the slope is unchanged by centring:

$$\widehat{y} = a + b\,\text{capacity} = (a + 4b) + b(\text{capacity} - 4).$$

The coefficient b is identical; the intercept moves from the prediction at capacity zero to the prediction at capacity four. Centring improves the reference meaning without changing fitted values.

← **Return to the fitted London line**

Appendix A4: why squared loss targets the conditional mean

For a fixed feature value $\mathbf{x}$, choose a constant a to minimise

$$R(a \mid \mathbf{x}) = \mathbb{E}[(Y - a)^2 \mid \mathbf{X} = \mathbf{x}].$$

Let $m(\mathbf{x}) = \mathbb{E}[Y \mid \mathbf{X} = \mathbf{x}]$. Add and subtract $m(\mathbf{x})$:

$$\begin{aligned} R(a \mid \mathbf{x}) &= \mathbb{E}[(Y - m + m - a)^2 \mid \mathbf{x}] \\ &= \mathrm{Var}(Y \mid \mathbf{x}) + \big(m(\mathbf{x}) - a\big)^2, \end{aligned}$$

because the cross term has conditional expectation zero. The first term does not depend on a; the second is minimised at

$$a^\star = m(\mathbf{x}).$$

The fitted linear model is a restricted approximation to that conditional mean, not an assumption that every outcome lies on a line.

← **Return to the loss-matched baseline**

Appendix A5: what a partial regression coefficient compares

Partition the design into one feature $\mathbf{x}_j$ and the remaining columns $\mathbf{X}_{-j}$. Remove from both y and x_j the parts linearly predicted by $\mathbf{X}_{-j}$:

$$\mathbf{r}_y = \mathbf{M}_{-j}\mathbf{y}, \qquad \mathbf{r}_j = \mathbf{M}_{-j}\mathbf{x}_j, \qquad \mathbf{M}_{-j} = \mathbf{I} - \mathbf{X}_{-j}(\mathbf{X}_{-j}^\top \mathbf{X}_{-j})^{-1}\mathbf{X}_{-j}^\top.$$

$\mathbf{I}$ = identity matrix; $\mathbf{M}_{-j}$ = residual-maker matrix that removes the part linearly predicted by $\mathbf{X}_{-j}$.

The Frisch–Waugh–Lovell result gives

$$\widehat{\beta}_j = \frac{\mathbf{r}_j^\top \mathbf{r}_y}{\mathbf{r}_j^\top \mathbf{r}_j}.$$

So $\widehat{\beta}_j$ compares the part of x_j not linearly explained by the other represented columns with the corresponding remaining part of y. It is conditional on the chosen representation, not automatically on all real-world confounders.

← **Return to the London contribution path**

Appendix A6: reference coding and matrix rank

For three categories A, B, C, the indicators satisfy

$$\mathbb{1}(A) + \mathbb{1}(B) + \mathbb{1}(C) = 1.$$

Including all three indicators together with an intercept makes one column an exact linear combination of the others, so the coefficient vector is not unique. Choose A as reference:

$$\widehat{y} = \beta_0 + \gamma_B \mathbb{1}(B) + \gamma_C \mathbb{1}(C).$$

Then

A	β_0
B	$\beta_0 + \gamma_B$
C	$\beta_0 + \gamma_C$

and γ_B, γ_C are comparisons with A, conditional on the other represented columns. Predictions can be parameterised in other equivalent ways; interpretation must follow the chosen coding.

← **Return to room-type indicators**

Appendix A7: linear in parameters

The model

$$\widehat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 \log(1 + z)$$

is linear in its parameters because each β_j enters to the first power and the score is a weighted sum. Define

$$\boldsymbol{\phi}(x, z) = (x, x^2, \log(1 + z))^\top.$$

Then the expression is simply

$$\widehat{y} = \beta_0 + \boldsymbol{\phi}(x, z)^\top \boldsymbol{\beta}.$$

By contrast, $y = \beta_0 + \exp(\beta_1 x)$ is nonlinear in β_1. Feature maps can make a global linear score quite expressive, but every new column is a modelling assumption whose transfer must be checked.

← **Return to feature-map shapes**

Appendix A8: input transformations and target retransformation

Transforming an input changes the permitted shape while keeping the outcome scale:

$$\widehat{y} = \beta_0 + \beta_1 \log(1 + x).$$

Transforming the target changes the conditional quantity being fitted:

$$\mathbb{E}[\log \widehat{Y \mid X} = x] = \beta_0 + \beta_1 x.$$

Exponentiating gives

$$\exp\{\mathbb{E}[\log Y \mid X = x]\},$$

which is generally not $\mathbb{E}[Y \mid X = x]$ because

$$\mathbb{E}[e^Z] \neq e^{\mathbb{E}[Z]}.$$

$\mathbb{E}[\cdot]$ denotes expectation (a probability-weighted average); Z is a random quantity on the log scale; σ^2 is the assumed constant variance of the log-scale error.

Under a homoscedastic Gaussian log-error model, a factor $\exp(\sigma^2/2)$ corrects the mean; in general, retransformation requires a model-appropriate or smearing estimate learned without test leakage.

← **Return to the transformation distinction**

Appendix A9: interaction slopes and conditional gaps

With centred capacity c and private-room indicator P,

$$\widehat{y} = \beta_0 + \beta_1 c + \gamma P + \delta c P.$$

For an entire home ($P = 0$):

$$\widehat{y}_E = \beta_0 + \beta_1 c.$$

For a private room ($P = 1$):

$$\widehat{y}_P = (\beta_0 + \gamma) + (\beta_1 + \delta) c.$$

Therefore

$$\frac{\partial \widehat{y}_E}{\partial c} = \beta_1, \qquad \frac{\partial \widehat{y}_P}{\partial c} = \beta_1 + \delta, \qquad \widehat{y}_P - \widehat{y}_E = \gamma + \delta c.$$

The main effect γ is the private-room gap at the centred reference $c = 0$, not an average gap over every capacity.

← **Return to the interaction mechanism**

Appendix A10: fitting and applying a complete pipeline

Let T_θ denote imputation, scaling, encoding, and feature construction with learned state θ. Development fitting produces

$$\widehat{\theta} = \mathrm{FitTransformState}(\mathbf{X}_{\mathrm{dev}}), \qquad \mathbf{\Phi}_{\mathrm{dev}} = T_{\widehat{\theta}}(\mathbf{X}_{\mathrm{dev}}),$$

$\widehat{\theta}$ = fitted preparation state; $\mathbf{\Phi}_{\mathrm{dev}}$ = resulting development design matrix.

$$\widehat{\boldsymbol{\beta}} = \mathrm{FitModel}(\mathbf{\Phi}_{\mathrm{dev}}, \mathbf{y}_{\mathrm{dev}}).$$

Validation uses no refitting:

$$\mathbf{\Phi}_{\mathrm{val}} = T_{\widehat{\theta}}(\mathbf{X}_{\mathrm{val}}), \qquad \widehat{\mathbf{y}}_{\mathrm{val}} = f_{\widehat{\boldsymbol{\beta}}}(\mathbf{\Phi}_{\mathrm{val}}).$$

If validation records determine $\widehat{\theta}$, information has crossed the evidence boundary even when the coefficient estimator never saw $\mathbf{y}_{\mathrm{val}}$.

← **Return to the 39-column London pipeline**

Appendix A11: nested training fit and predictive R^2

If the smaller feature space is contained in the richer space, every smaller model is available to the richer fit by setting new coefficients to zero. Hence

$$\min_{f \in \mathcal{F}_{\text{rich}}} \sum_{\text{dev}} (y_i - f(x_i))^2 \leq \min_{f \in \mathcal{F}_{\text{small}}} \sum_{\text{dev}} (y_i - f(x_i))^2.$$

$\mathcal{F}_{\text{small}}$ and $\mathcal{F}_{\text{rich}}$ = permitted rule sets; the richer set contains the smaller one.

No corresponding ordering holds on validation data.

The London predictive R^2 uses the frozen development mean as comparator:

$$R^2_{\text{pred}} = 1 - \frac{\sum_{\text{val}} (y_i - \widehat{y}_i)^2}{\sum_{\text{val}} (y_i - \bar{y}_{\text{dev}})^2}.$$

$\bar{y}_{\text{dev}}$ = frozen development-sample mean used as the held-out baseline.

Using $\bar{y}_{\text{val}}$ would let the validation outcomes improve the comparator and would define a different quantity.

← **Return to training fit versus generalisation**

Appendix A12: support, leverage, and extrapolation

For a represented new case $\widetilde{\mathbf{x}}_0$, the fitted linear prediction is

$$\widehat{y}_0 = \widetilde{\mathbf{x}}_0^\top \widehat{\boldsymbol{\beta}}.$$

Its design leverage relative to the fitting sample is

$$h_0 = \widetilde{\mathbf{x}}_0^\top (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{x}}_0.$$

$\widetilde{\mathbf{x}}_0$ = new case after fitted representation, including the intercept; $\widetilde{\mathbf{X}}$ = matching fitting design matrix; h_0 = design leverage.

Large h_0 means the case lies in a weakly supported direction of the fitted feature space. This may occur because a raw input is outside range, a rare combination activates an interaction, or polynomial terms grow rapidly.

The equation always returns a number if its inputs are accepted. The leverage and support checks answer the different question: how much fitting evidence anchored that number?

← **Return to the impossible-price diagnostic**

Appendix A13: exact and near multicollinearity

If $\mathbf{x}_2 = 2\mathbf{x}_1$, then

$$\beta_1\mathbf{x}_1 + \beta_2\mathbf{x}_2 = (\beta_1 + 2\beta_2)\mathbf{x}_1,$$

so the individual coefficients are not identified. Under near collinearity, the solution is unique but can be highly sensitive to small sample changes.
For feature j, the variance-inflation factor is

$$\mathrm{VIF}_j = \frac{1}{1 - R_j^2},$$

where R_j^2 comes from regressing that design column on the others. It describes linear redundancy in this fitted design; there is no universal VIF threshold that automatically invalidates a predictive model. Prediction stability, coefficient stability, and causal identification are three separate questions.

← **Return to correlated-input instability**

Appendix A14: ridge normal equations and software conventions

After centring and excluding the intercept, ridge minimises

$$\frac{1}{2n}\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2.$$

$\|\cdot\|_2$ denotes Euclidean length; $\mathbf{I}$ denotes the identity matrix.

Setting the gradient to zero gives

$$\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}+n\lambda\mathbf{I}\right)\widehat{\beta}_\lambda=\boldsymbol{\Phi}^\top\mathbf{y},$$

$$\widehat{\beta}_\lambda=\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}+n\lambda\mathbf{I}\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{y}.$$

The added diagonal makes the matrix invertible for $\lambda>0$.

Convention warning. With this objective, scikit-learn Ridge uses $\alpha=n\lambda$. Scikit-learn Lasso uses the $1/(2n)$ loss convention; LogisticRegression reports inverse strength C. Values are not directly comparable without mapping the objective.

← **Return to ridge intuition**

Appendix A15: lasso subgradients and soft thresholding

For

$$\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

the optimality condition for coefficient j is

$$\frac{1}{n}\boldsymbol{\phi}_j^\top(\mathbf{y}-\boldsymbol{\Phi}\widehat{\boldsymbol{\beta}}) = \lambda s_j,$$

where

$$s_j = \begin{cases} \operatorname{sign}(\widehat{\beta}_j), & \widehat{\beta}_j \neq 0, \\ \text{any value in } [-1,1], & \widehat{\beta}_j = 0. \end{cases}$$

$\|\boldsymbol{\beta}\|_1$ = sum of absolute coefficients; s_j = subgradient of $|\beta_j|$; z = one-feature OLS estimate; $(a)_+ = \max(a,0)$.

With one standardised orthogonal feature and OLS estimate z, the solution is the soft-threshold operator

$$\widehat{\beta}^{\text{lasso}} = \operatorname{sign}(z)(|z|-\lambda)_+.$$

The flat subgradient interval at zero is why exact zeros are possible.

← **Return to ridge versus lasso**

Appendix A16: scaling changes the penalty

If $x_j^\star = a x_j$, preserving the same fitted contribution requires

$$\beta_j^\star = \frac{\beta_j}{a}.$$

Without standardisation, the ridge charge changes from $\lambda\beta_j^2/2$ to

$$\frac{\lambda}{2}\left(\frac{\beta_j}{a}\right)^2.$$

Thus measurement units alter shrinkage even when predictions are represented identically. Training-fitted standardisation makes the numerical comparison more coherent:

$$z_{ij} = \frac{x_{ij} - \bar{x}_{j,\text{fit}}}{s_{j,\text{fit}}}.$$

Caution: penalised reference-coded categorical models are not fully invariant to the omitted category because the penalty is applied only to represented coefficients. Reference choice and penalty design should be documented.

← **Return to penalty scaling**

Appendix A17: cross-validation and final refitting

For fold k and candidate λ:

$$\widehat{f}_{\lambda}^{(-k)} = \mathrm{FitPipeline}\left(\mathcal{D}_{\mathrm{dev}} \setminus \mathcal{D}_k; \lambda\right),$$

$$\mathrm{CV}(\lambda) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} L\left(y_i, \widehat{f}_{\lambda}^{(-k)}(x_i)\right).$$

$\mathcal{D}_k$ = fold k; K = number of folds; L = declared loss.

Choose $\lambda^{\star}$ by the declared rule, then refit the complete pipeline on all permitted development evidence:

$$\widehat{f}^{\star} = \mathrm{FitPipeline}(\mathcal{D}_{\mathrm{dev}}; \lambda^{\star}).$$

Groups that must travel together—London hosts, repeated clients, sites, or time blocks—must remain together inside the resampling design. The final test does not select λ, features, or the model family.

← **Return to validation choice of λ**

Appendix A18: workflow stability by host-level resampling

For resample $b = 1, \ldots, B$:

$$\mathcal{H}^{(b)} \sim \text{ResampleWithReplacement}(\text{development hosts}),$$

$$\widehat{f}^{(b)} = \text{FitCompleteWorkflow}\left(\{\text{listings belonging to } \mathcal{H}^{(b)}\}\right).$$

Record for every resample:

$$\widehat{\beta}_j^{(b)}, \qquad \widehat{f}^{(b)}(x_{\text{profile}}), \qquad \text{Loss}_{\text{val}}(\widehat{f}^{(b)}).$$

B = number of host-level resamples; $\mathcal{H}^{(b)}$ = resampled development hosts; x_{profile} = one fixed represented profile scored by every refitted workflow.

The distributions answer different questions:

- coefficient wobble: is the claimed comparison stable?
- profile wobble: are case-level predictions stable?
- validation wobble: is apparent performance driven by a few hosts?

Resampling individual listings would pretend that listings from the same host are independent evidence.

← **Return to the slope-wobble figure**

Appendix A19: classical OLS sampling variance

Under the fixed-design linear model

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

$\boldsymbol{\varepsilon}$ = error vector; $\mathbb{E}[\cdot \mid \mathbf{X}]$ and $\mathrm{Var}(\cdot \mid \mathbf{X})$ = conditional mean and variance; σ^2 = common error variance; $\mathbf{I}$ = identity matrix.

OLS has

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}} \mid \mathbf{X}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}.$$

Replacing σ^2 by the residual estimate gives classical standard errors. Heteroscedasticity-robust estimators replace the middle $\sigma^2\mathbf{I}$ term with a residual-weighted matrix.

These formulas do not repair dependence between listings, post-selection inference, target leakage, distribution shift, or a misspecified causal claim. For this course, grouped workflow resampling is the main stability evidence; classical intervals remain a technical reference.

← **Return to the frozen London evidence**

Appendix A20: mean interval versus prediction interval

For a new represented case $\mathbf{x}_0$, a confidence interval for the conditional mean uses

$$\mathrm{SE}\{\widehat{m}(\mathbf{x}_0)\} = \widehat{\sigma}\sqrt{\mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0}.$$

A prediction interval for a new outcome adds irreducible outcome variation:

$$\mathrm{SE}\{Y_{\text{new}} - \widehat{m}(\mathbf{x}_0)\} = \widehat{\sigma}\sqrt{1 + \mathbf{x}_0^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_0}.$$

SE = estimated standard error; $\widehat{m}(\mathbf{x}_0)$ = fitted conditional mean; $\widehat{\sigma}$ = residual standard-deviation estimate.

The second is wider because it addresses an individual future Y, not only uncertainty about the mean. Both formulas rely on the stated linear-model and error assumptions; neither guarantees subgroup or shifted-deployment coverage.

← **Return to the frozen London evidence**

Appendix A21: binary means and the linear-probability model

For $Y \in \{0, 1\}$,

$$\mathbb{E}[Y \mid X = x] = 0 \Pr(Y = 0 \mid x) + 1 \Pr(Y = 1 \mid x) = \Pr(Y = 1 \mid x).$$

Ordinary least squares on a binary target therefore approximates this conditional probability with

$$\widehat{p}_{\mathrm{LPM}}(x) = \beta_0 + \boldsymbol{\phi}(x)^\top \boldsymbol{\beta}.$$

Its coefficient is a probability-point comparison under the represented linear form, but:

- fitted values need not remain in $[0, 1]$;
- conditional variance is $p(x)[1 - p(x)]$, so it is inherently heteroscedastic;
- constant probability slopes can be implausible near the boundaries.

← **Return to the binary conditional mean** ← **Return to the LPM diagnostic**

Appendix A22: odds, logit, and the sigmoid inverse

Start with the additive log-odds model:

$$\eta = \log\left(\frac{p}{1-p}\right).$$

Exponentiate and solve for p:

$$e^{\eta} = \frac{p}{1-p}, \qquad e^{\eta}(1-p) = p,$$

$$e^{\eta} = p(1+e^{\eta}), \qquad p = \frac{e^{\eta}}{1+e^{\eta}} = \frac{1}{1+e^{-\eta}}.$$

Thus the logit maps $p \in (0,1)$ to $\eta \in \mathbb{R}$, and the sigmoid maps every $\eta \in \mathbb{R}$ back to one valid probability. At $\eta = 0$, $p = 0.5$; the mapping is monotone, so ranking by score and probability is identical.

← **Return to the unrestricted score**

Appendix A23: Bernoulli likelihood becomes log loss

For independent Bernoulli outcomes with fitted probabilities p_i,

$$\Pr(Y_i = y_i \mid x_i) = p_i^{y_i}(1-p_i)^{1-y_i}.$$

The likelihood and log-likelihood are

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i},$$

$$\log \mathcal{L}(\beta) = \sum_i \left[y_i \log p_i + (1-y_i)\log(1-p_i)\right].$$

Maximising the log-likelihood is exactly equivalent to minimising average binary log loss:

$$-\frac{1}{n}\log \mathcal{L}(\beta) = -\frac{1}{n}\sum_i \left[y_i \log p_i + (1-y_i)\log(1-p_i)\right].$$

The likelihood supplies an estimation principle; held-out log loss supplies the generalisation evidence.

← **Return to logistic fitting**

Appendix A24: logistic gradient, Hessian, and IRLS

With design matrix $\mathbf{\Phi}$, probability vector $\mathbf{p} = \sigma(\mathbf{\Phi}\boldsymbol{\beta})$, and average log loss $\bar{\ell}$,

$$\nabla_{\boldsymbol{\beta}} \bar{\ell} = \frac{1}{n} \mathbf{\Phi}^{\top} (\mathbf{p} - \mathbf{y}).$$

Let

$$\mathbf{W} = \operatorname{diag}\{p_i(1 - p_i)\}.$$

$\nabla_{\boldsymbol{\beta}} \bar{\ell}$ is the gradient (vector of first partial derivatives); $\nabla^2_{\boldsymbol{\beta}} \bar{\ell}$ is the Hessian (matrix of second partial derivatives); $\operatorname{diag}(a_i)$ is the diagonal matrix with a_i on its diagonal.

Then

$$\nabla^2_{\boldsymbol{\beta}} \bar{\ell} = \frac{1}{n} \mathbf{\Phi}^{\top} \mathbf{W} \mathbf{\Phi},$$

which is positive semidefinite. Newton's update is

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left[\nabla^2 \bar{\ell}(\boldsymbol{\beta}^{(t)}) \right]^{-1} \nabla \bar{\ell}(\boldsymbol{\beta}^{(t)}).$$

Written as iteratively reweighted least squares, each iteration fits a local quadratic approximation. Convergence diagnostics remain part of the fitted workflow.

← **Return to the fitting objective**

Appendix A25: separation and penalised logistic fitting

Complete separation occurs when some vector $\mathbf{b}$ satisfies

$$\boldsymbol{\phi}_i^\top \mathbf{b} > 0 \quad \text{for every } y_i = 1, \qquad \boldsymbol{\phi}_i^\top \mathbf{b} < 0 \quad \text{for every } y_i = 0.$$

Multiplying $\mathbf{b}$ by a growing constant drives fitted probabilities toward one for all positives and zero for all negatives. The unpenalised likelihood approaches its supremum while finite coefficients need not exist.

Ridge logistic regression instead minimises

$$\bar{\ell}(\beta_0, \boldsymbol{\beta}) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2,$$

so diverging coefficients incur an increasing penalty and a finite optimum is obtained for $\lambda > 0$. This addresses numerical and estimation instability; validation and calibration are still required.

← **Return to regularised logistic regression**

Appendix A26: odds comparisons and probability changes

For an untransformed continuous feature with no interaction,

$$\frac{\partial \eta}{\partial x_j} = \beta_j, \qquad \frac{\partial p}{\partial x_j} = \frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial x_j} = \beta_j p(1-p).$$

Thus the probability slope depends on p and is largest near $p = 0.5$. For a discrete or categorical change, prefer the exact profile contrast

$$\Delta p = \sigma\big(\eta(x_j = 1, \mathbf{x}_{-j})\big) - \sigma\big(\eta(x_j = 0, \mathbf{x}_{-j})\big).$$

$\mathbf{x}_{-j}$ denotes all represented inputs other than x_j; they are held fixed when comparing $x_j = 1$ and $x_j = 0$.
With transformations or interactions, $\partial\eta/\partial x_j$ is no longer simply β_j; all dependent terms must change together. Exponentiated coefficients describe conditional odds ratios, not relative probabilities and not causal risk ratios.

← **Return to nonlinear probability changes** ← **Return to odds interpretation**

Appendix A27: calibration, Brier score, and bin uncertainty

Calibration requires

$$\Pr(Y = 1 \mid \widehat{p} = p) \approx p.$$

For a validation bin B,

$$\bar{p}_B = \frac{1}{n_B} \sum_{i \in B} \widehat{p}_i, \qquad \bar{y}_B = \frac{1}{n_B} \sum_{i \in B} y_i.$$

Plot $(\bar{p}_B, \bar{y}_B)$, report n_B, and show uncertainty. A Wilson interval for an observed proportion $\widehat{q} = k/n$ is centred at

$$\frac{\widehat{q} + z^2/(2n)}{1 + z^2/n}$$

with half-width

$$\frac{z}{1 + z^2/n} \sqrt{\frac{\widehat{q}(1 - \widehat{q})}{n} + \frac{z^2}{4n^2}}.$$

z is the selected standard-normal critical value (for example, 1.96 for a two-sided 95% interval).

$$\text{Brier score} = \frac{1}{n} \sum_i (y_i - \widehat{p}_i)^2$$

is a proper probability score, but neither it nor log loss replaces the local calibration diagnostic. Bins are descriptive and bin-dependent.

← **Return to probability as a frequency claim**

Appendix A28: London data and validation ledger

Source	Inside Airbnb London snapshot, 19 June 2026; file hash recorded in the reproducible build
Downloaded / eligible	92,638 rows / 54,636 listings after the declared short-stay and field-eligibility rules
Hosts	27,278 distinct hosts; hosts do not cross the three partitions
Development	32,749 listings from 16,381 hosts
Validation	11,063 listings from held-out hosts
Final audit	10,824 listings from held-out hosts; sealed through the Day 3 family comparison
Target	advertised nightly quote for the represented 1–7-night query, not a completed transaction, market value, revenue, or optimal price
Transfer claim	in-snapshot transfer to unseen hosts; not future forecasting
Compact design	four numerical columns, three room-type indicators, and 32 borough indicators; 39 coefficient columns plus intercept

No raw host or listing record is reproduced in the slide deck.

← **Return to the London prediction contract**

Appendix A29: Bank data, feature clock, and corrected build

Source	UCI `bank-additional-full.csv`; 41,188 contacts, 4,640 yes outcomes; file hash recorded in the reproducible build
Split	stratified 24,712 development / 8,238 validation / 8,238 final using seeds 42 and 43
Prediction clock	current contact plan is known; call outcome and `duration` are not; a pre-campaign targeting task would need fewer fields
pdays correction	replace raw 999 sentinel by $\mathbb{1}(pdays = 999)$ and $pdays_{\text{days}} = 0$ for sentinel records
Preparation	10 numeric fields scaled on development only; 10 categorical fields reference encoded; 53 coefficient columns plus intercept
Estimator	unpenalised logistic; L-BFGS (a limited-memory quasi-Newton optimiser), tight tolerance; corrected validation log loss 0.271754
Calibration region	for $0.20 \leq \widehat{p} < 0.30$: $n = 416$, mean claim 0.24347, observed $141/416 = 0.33894$, Wilson 95% $[0.29511, 0.38572]$
Benchmark limitations	12 duplicate pairs, four crossing random partitions; no stable client ID or exact date, so the split cannot establish client-disjoint or future-deployment performance

← **Return to the Bank prediction contract**

Appendix A30: the typical-miss numbers of the capacity line

The typical miss of a line is its RMSE over the $n = 32{,}749$ fitting listings: square every miss, average, take the square root.

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

Line	Prediction rule	Typical miss (RMSE)
illustrative candidate	$100 + 40 \times \text{capacity}$	£140.36
least squares	$64.73 + 46.97 \times \text{capacity}$	£139.11

- Least squares is defined as the minimiser of that score, so no other straight line in capacity can go below £139.11 on these listings.
- These are fitting-sample figures. The checking-sample numbers quoted in the main narrative are larger, as unseen hosts always are.

← **Return to the fitted line for London**

Appendix A31: R^2 for the capacity line, computed

Both totals live on the 11,063 checking listings:

Quantity	How it is computed	Value
baseline's total	add all 11,063 squared misses of the £228.38 quote	322.5 million £^{2}
the line's total	the same, for the capacity line's quotes	208.4 million £^{2}

$$R^2 \;=\; 1 - \frac{208.4}{322.5} \;=\; 1 - 0.646 \;=\; 0.354.$$

- Each total is $n \times \text{RMSE}^2$: for the baseline $11{,}063 \times 170.73^2$, for the line $11{,}063 \times 137.27^2$.
- Totals of squared pounds are unreadable as money. That is why the slides report the share removed, not the totals themselves.

← **Return to R^2**

Appendix A32: before adjusting, ordinary R^2

Start with an invented fitting sample. The same numbers carry into adjusted R^2 in A33.

Quantity	Value	Meaning
Cases, n	25	observations used to fit the model
Input columns, k	10	coefficients, excluding the intercept
SST	1,000	baseline's total squared misses
SSE	600	model's total squared misses

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}} = 1 - \frac{600}{1{,}000} = 0.40.$$

The model removes 40% of the baseline squared error. Notice that this calculation never uses the number of input columns.

← **Return to adjusted R^2**

Appendix A33: adjusting R^2 for the number of inputs

Adjusted R^2 compares squared error after allowing for the number of coefficients estimated:

$$\bar{R}^2 = 1 - \frac{\mathrm{SSE}/(n-k-1)}{\mathrm{SST}/(n-1)} = 1 - \left(1 - R^2\right)\frac{n-1}{n-k-1}.$$

n = number of observations; k = number of input columns, excluding the intercept. Only $n-k-1$ degrees of freedom remain for the misses.

For the invented sample of A32:

$$\bar{R}^2 = 1 - \frac{600/(25-10-1)}{1{,}000/(25-1)} = 1 - \frac{42.86}{41.67} = -0.03.$$

- **Small study:** ten inputs among only 25 cases turn 0.40 into -0.03; the apparent gain does not justify the complexity.
- **London:** with $n = 32{,}749$, $k = 39$, and $R^2 = 0.528$, the adjusted value is 0.527; the penalty is tiny.

← **Return to adjusted R^2**

Appendix A34: the standard error formula for the slope

For the one-input line, the classical formula is one line:

$$\mathrm{SE}(\widehat{\beta}_1) = \sqrt{\frac{\widehat{\sigma}^2}{\sum_i (x_i - \bar{x})^2}}, \qquad \widehat{\sigma}^2 = \frac{\sum_i e_i^2}{n-2}.$$

$\widehat{\sigma}^2$ = spread of the misses around the fitted line; $\sum_i (x_i - \bar{x})^2$ = how much capacity varies, summed over the sample.

- Noisier prices (bigger $\widehat{\sigma}$) widen the SE; more listings and more spread in capacity shrink it.
- For London, SE $= 0.354$, an accidental twin of the R^2 value 0.354; the two numbers are unrelated.
- This version assumes the misses spread equally at every capacity. London's do not, since expensive listings miss bigger, so the robust version grows to 0.520. Both are tiny beside a slope of 46.97.

← **Return to the standard error**

Appendix A35: why more observations reduce slope wobble

Across $R = 200$ refits, the empirical SD of the slopes estimates $\mathrm{SE}(\widehat{\beta}_1)$:

$$s_{\widehat{\beta}_1} = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}\left(\widehat{\beta}_1^{(r)} - \overline{\widehat{\beta}}_1\right)^2}, \qquad s_{500} = £4.50, \quad s_{5000} = £1.17.$$

For the one-input regression, under the classical assumptions,

$$\mathrm{Var}(\widehat{\beta}_1 \mid X) = \frac{\sigma^2}{S_{xx}}, \qquad \mathrm{SE}(\widehat{\beta}_1 \mid X) = \frac{\sigma}{\sqrt{S_{xx}}} = \frac{\sigma}{s_x\sqrt{n-1}}, \quad S_{xx} = (n-1)s_x^2.$$

Holding the residual SD σ and the capacity spread s_x fixed,

$$\frac{\mathrm{SE}_{5000}}{\mathrm{SE}_{500}} \approx \sqrt{\frac{500}{5000}} = \frac{1}{\sqrt{10}} \approx 0.32.$$

Ten times the observations gives roughly one-third as much slope variation, not one-tenth.

← **Return to 500 versus 5,000 listings**

Appendix A36: the t statistic and the p-value, computed

The t statistic divides the estimate by its standard error, both taken unrounded:

$$t = \frac{\widehat{\beta}_1}{\mathrm{SE}(\widehat{\beta}_1)} = \frac{46.968}{0.354} = 132.8.$$

The p-value is the area in both tails of the sampling curve beyond $|t|$: the share of luck-only samples at least this far from zero. With $n = 32{,}749$ the curve is the normal curve.

> At $|t| = 1.96$: each tail has area 0.025, so the two-sided p-value is 0.050.

- Software does the area lookup; at $t = 132.8$ the area is far below 0.001 and prints as 0.000.
- In small samples the lookup uses Student's t curve, with heavier tails; at this sample size the difference vanishes.

← **Return to the p-value**