Quantitative Research Methods

Conformal Prediction

Advanced module A3 — optional self-study

Prof. Dr. Christoph Weisser

HSBI

Summer Semester 2026

Where this module fits

The four **advanced modules** are optional self-study decks that extend the twelve ISLP lectures. This one turns any predictive model from the course into a model with *honest uncertainty*.

You already know (chapter)	**What module A3 adds**
Ch. 3 — prediction intervals for linear regression	Intervals with *guaranteed* coverage, without Gaussian errors or a correct model
Ch. 5 — train / validation splits	A third role for held-out data: the *calibration set*
Ch. 4 — classifiers and their probabilities	Prediction *sets*: "{flu, covid}, with 90% guarantee"
Ch. 8 — gradient boosting	Boosted *quantile* regression as the engine of adaptive intervals (CQR)

Prerequisites: Chapters 3 and 5 are essential, Chapters 4 and 8 helpful. Short exercises every $\sim$20 min; extended exercises every $\sim$45 min; the module has a companion lab notebook.

Contents

Optional and advanced material is collected in the **appendix** at the end of the deck.

Notation in this module

Symbol	Meaning
(x_i, y_i)	feature–response pair of observation i
$\mathcal{D}_{\text{tr}}$, $\mathcal{D}_{\text{cal}}$	proper training set and calibration set (disjoint)
n	number of calibration points, $n = \lvert\mathcal{D}_{\text{cal}}\rvert$
$\hat{f}$	prediction rule fitted on $\mathcal{D}_{\text{tr}}$ only
s_i	conformity score of calibration point i (e.g. $\lvert y_i - \hat{f}(x_i)\rvert$)
$s_{(k)}$	k-th smallest calibration score (order statistic)
α	target miscoverage level (we use $\alpha = 0.1$: 90% coverage)
$\hat{q}$	conformal quantile: $s_{(k)}$ with $k = \lceil (n+1)(1-\alpha) \rceil$
$C(x)$	prediction interval (regression) or prediction set (classification)
$\rho_\tau(u)$	pinball loss of quantile regression at level τ
$\hat{q}_{\text{lo}}$, $\hat{q}_{\text{hi}}$	fitted conditional quantile curves (levels $\alpha/2$, $1-\alpha/2$)
$\hat{p}_k(x)$	estimated probability of class k at x (softmax output)
K	number of classes

Sources and references

Primary textbook

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning, with Applications in Python.* Springer. (Chapters 3 and 5 supply the prerequisites.)

Further reading on conformal prediction

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic Learning in a Random World.* Springer. — Lei, J., et al. (2018). Distribution-free predictive inference for regression. *JASA* 113(523). — Romano, Y., Patterson, E., & Candès, E. (2019). Conformalized quantile regression. *NeurIPS*. — Angelopoulos, A., & Bates, S. (2023). Conformal prediction: a gentle introduction. *Foundations and Trends in Machine Learning* 16(4).

Learning objectives

By the end of this module you will be able to:

- **Explain** why classical prediction intervals need assumptions that conformal prediction drops.
- **State** exchangeability and **judge** when it is plausible.
- **Apply** split conformal prediction: compute $\hat{q}$ and the interval by hand and in Python.
- **Distinguish** marginal from conditional coverage — and say which one is guaranteed.
- **Build** adaptive intervals with conformalized quantile regression (CQR).
- **Construct** prediction sets for classifiers and read their sizes.
- **Avoid** the pitfalls: leakage into the calibration set, time series, tiny calibration sets.

Roadmap of this module

Uncertainty you can guarantee

1. Why a point prediction is a guess — and what Ch. 3's intervals assume.
2. Exchangeability: the one assumption conformal prediction needs.
3. Split conformal for regression: the whole recipe, and its proof sketch.
4. Marginal vs conditional coverage — the honest reading of the guarantee.
5. Adaptive intervals: locally weighted scores and CQR.
6. Prediction sets for classification; conformal vs the classical interval; pitfalls and production practice.

Where conformal prediction is used in industry

Sector	Concrete application	Which decision it drives
Energy	Intervals around day-ahead load and wind forecasts	How much reserve capacity to buy
E-commerce, logistics	Delivery-time windows behind the checkout promise	Which arrival dates to display and guarantee
Healthcare	Prediction sets over diagnoses in triage support	Which cases are routed to a human specialist
Banking	Intervals on loss and prepayment forecasts	Capital buffers; which cases go to manual review
Pharma R&D	Conformal sets in virtual screening of compounds	Which molecules advance to the wet lab
Manufacturing	Set-valued defect classification on the line	Which parts get pulled for manual inspection

In industry — The common thread

In every row the model's *point* prediction was already available — the money is in the **width of the interval** or the **size of the set**, because that is what sizes the buffer, the promise, or the manual-review queue.

Industry case in depth: the delivery promise at an online retailer

In industry — The problem: a point ETA is a promise you will break

The checkout page must commit to an arrival date. A gradient-boosted model predicts delivery time from warehouse, carrier, destination and weekday — but showing its point estimate ("arrives Wednesday") is wrong for a large share of parcels, and every miss costs a support ticket or a refund. What the business needs is a **window** ("Tuesday–Thursday") that holds, say, 90% of the time.

How conformal prediction is wired in

Last month's delivered parcels are split: most **retrain** the model, a held-out slice becomes the **calibration set**. Absolute errors on that slice give $\hat{q}$; the site displays $\hat{f}(x) \pm \hat{q}$ rounded to days. Recalibration runs *weekly*, so carrier disruptions feed into $\hat{q}$ within days; a dashboard tracks realised coverage and median window width.

Two honest caveats

Delivery times *drift* (peak season, strikes) — exchangeability between calibration and next week is an approximation, hence the weekly recalibration. And the guarantee is *marginal*: remote postcodes can be undercovered while city centres are overcovered, unless calibration is done per group.

Outline

A point prediction without uncertainty is a guess

Take the `Wage` data ($n = 3\,000$ workers): predict **wage** from **age**.

How much do 45-year-olds actually earn?

Among the 268 workers aged 44–46, wages average \$120 400 — but they range from \$36 700 to \$318 300, with a standard deviation of \$43 400. A model that answers "\$120k" communicates *none* of that spread: the single number is a guess dressed up as knowledge.

- Every model from this course — linear regression, boosting, a neural net — produces $\hat{f}(x)$. The question this module answers: *how far can the truth be from $\hat{f}(x)$, with a guarantee?*
- We want a recipe that works for **any** $\hat{f}$, on **finite** samples, **without** trusting the model.

In industry — Pricing and planning: the interval is the product

A buyer sizing stock, a hospital planning beds, a grid operator buying reserve — none of them can act on "the forecast is 120". They act on "between 80 and 160, 9 times out of 10".

What Chapter 3 gave us — and what it silently assumed

The classical 90% prediction interval (Ch. 3)

Under the linear model $Y = X\beta + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ i.i.d.: $\quad \hat{y}_0 \;\pm\; t_{n-p-1,\,0.95}\,\hat{\sigma}\sqrt{1 + x_0^\top (X^\top X)^{-1} x_0}.$

The fine print — three assumptions doing all the work

1. **Correct mean model**: $E[Y \mid X = x]$ really is linear in x.
2. **Gaussian errors**: the t/normal quantile calibrates the width.
3. **Homoskedasticity**: one σ for every x.

Break any of the three and the "90%" printed by `statsmodels` is a claim, not a fact.

Wage breaks the fine print

Wages are strongly right-skewed (skewness 1.68; 79 workers, 2.6%, earn above \$250k). The Gaussian recipe $\hat{f}(x) \pm 1.645\,\hat{\sigma}$ gives half-width 66.1 — and covers 95.2% of test wages while *claiming* 90%: miscalibrated and needlessly wide.

The conformal promise

What conformal prediction guarantees

From any fitted model $\hat{f}$ and a calibration set of n held-out points, build $C(x)$ such that for a new exchangeable point (X_{n+1}, Y_{n+1}):

$$\boxed{\Pr\left(Y_{n+1} \in C(X_{n+1})\right) \geq 1 - \alpha}$$

Why this is remarkable — read the guarantee word by word

- **Distribution-free**: no Gaussian errors, no model for $P(Y \mid X)$ — wages can be as skewed as they like.
- **Finite-sample**: holds exactly at $n = 500$, not just as $n \to \infty$.
- **Any model**: $\hat{f}$ may be misspecified, overfit, or a black box.
- **Marginal**: averaged over the new point *and* the calibration draw — more on this honest word soon.

Takeaway

Conformal trades "if my model is right, coverage is exact at every x" for "*whatever* my model is, coverage is right *on average*".

Exercise A3.1 — What did we just buy? [Concept]

Exercise

1. Name the three assumptions behind the Chapter 3 prediction interval.
2. Which of the three does conformal prediction keep, and what does it demand instead?
3. A colleague says: "With conformal prediction, my 90% interval covers the true wage of *every 45-year-old* with probability 90%." Is that what the guarantee says?
4. Why does an overfit $\hat{f}$ not break the conformal guarantee (hint: where are the residuals computed)?

Solution — Exercise A3.1

Solution

Method. Compare the two guarantees assumption by assumption.

- **Part 1.** Correct (linear) mean model; Gaussian errors; homoskedastic errors (one σ for all x).
- **Part 2.** It keeps *none* of them. Instead it demands (i) **exchangeability** between calibration and test points and (ii) a **clean split**: $\hat{f}$ never sees the calibration data.
- **Part 3.** No. The guarantee is **marginal**: averaged over all new points (and calibration draws), at least 90% are covered. Coverage *among 45-year-olds specifically* may be higher or lower — we will measure exactly this on Wage (98.6% for under-30s, 82.9% for 60-plus, same interval).
- **Part 4.** The scores are residuals on *held-out* calibration points. An overfit $\hat{f}$ produces *large* held-out residuals, so $\hat{q}$ grows and the interval honestly widens — the guarantee survives; only the width suffers.

Common mistake

Reading "90%" as a per-person (conditional) probability. Marginal coverage is a statement about the *average* over the population, not about your specific x.

Outline

Exchangeability — the one assumption we need

Definition

Random variables $Z_1, \ldots, Z_m$ are **exchangeable** if for every permutation π,

$$(Z_{\pi(1)}, \ldots, Z_{\pi(m)}) \stackrel{d}{=} (Z_1, \ldots, Z_m).$$

What it means in plain words

- Every *ordering* of the data is equally likely — the labels "calibration point 17" and "test point" carry no information.
- **i.i.d.** $\Rightarrow$ **exchangeable**, but not conversely: exchangeability is *weaker* (e.g. sampling without replacement).
- Key consequence: among $n+1$ exchangeable scores with no ties, the **rank** of the last one is *uniform* on $\{1, \ldots, n+1\}$.

Takeaway

Conformal prediction is rank arithmetic on exchangeable scores. Everything follows from "the new point is nothing special".

The honest slide: what breaks exchangeability

- **Time series**: today's demand depends on yesterday's — orderings are *not* equally likely, and tomorrow may differ systematically.
- **Covariate shift**: calibrate on last year's customers, deploy on a new market — the X-distribution moved.
- **Feedback loops**: the model's own decisions (credit granted, price shown) change the data it will be judged on.

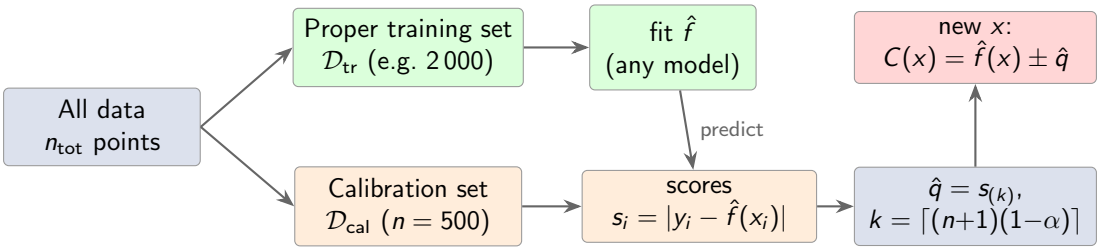
What still works

Violations degrade the guarantee *gradually*, and there are repairs: *weighted* conformal under known covariate shift (Tibshirani et al., 2019), blocked or online variants for dependence. This module sticks to the exchangeable case — the pitfalls section returns here.

In industry — Model monitoring: coverage as an alarm

Production ML teams track *realised coverage* weekly. Coverage sliding from 90% to 80% is a drift detector that needs no labels beyond the ones you already collect — often the first sign that the world has moved away from the training distribution.

Split conformal — the pipeline (native diagram)



Takeaway

One split, one model fit, one sorted list of residuals, one quantile. The model only ever sees $\mathcal{D}_{\text{tr}}$; the guarantee only ever uses $\mathcal{D}_{\text{cal}}$.

The whole recipe on one slide

Algorithm: split conformal prediction for regression

1. Split the data into $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{cal}}$ (sizes e.g. 2 000 and $n = 500$).
2. Fit $\hat{f}$ on $\mathcal{D}_{\text{tr}}$ *only*.
3. Compute calibration scores $s_i = |y_i - \hat{f}(x_i)|$ for $i \in \mathcal{D}_{\text{cal}}$.
4. Set $\boxed{\hat{q} = s_{(k)}, \qquad k = \lceil (n+1)(1-\alpha) \rceil}$ — the k-th smallest score.
5. For any new x report $C(x) = \big[\hat{f}(x) - \hat{q},\ \hat{f}(x) + \hat{q}\big]$.

Decoding the boxed formula

n = number of calibration points; α = tolerated miscoverage (0.1 for a 90% interval); $s_{(k)}$ = the k-th order statistic of the sorted scores; the ceiling $\lceil \cdot \rceil$ rounds *up*, so $\hat{q}$ is the $\frac{k}{n}$ empirical quantile — slightly *above* the naive $(1-\alpha)$-quantile. With $n = 500$, $\alpha = 0.1$: $k = \lceil 501 \times 0.9 \rceil = 451$, the 0.902 quantile.

Why the odd index? A tiny worked example

Nine calibration scores at 10% miscoverage

Scores: 1.2, 0.4, 2.8, 0.9, 1.7, 3.5, 0.6, 2.1, 1.0. Sorted: 0.4, 0.6, 0.9, 1.0, 1.2, 1.7, 2.1, 2.8, 3.5. Then $k = \lceil (9+1)(0.9) \rceil = 9 \Longrightarrow \hat{q} = s_{(9)} = 3.5$ (the maximum!). At $\alpha = 0.2$: $k = \lceil 10 \times 0.8 \rceil = 8$, so $\hat{q} = s_{(8)} = 2.8$. At $n = 19$, $\alpha = 0.1$: $k = \lceil 20 \times 0.9 \rceil = 18$ — the 18th of 19.

The intuition for the "$+1$"

The new point's score joins the n calibration scores as an $(n{+}1)$-th, equally-ranked member. To catch it with probability ≥ 0.9 we must clear at least 90% of *all* $n+1$ *ranks* — hence $(n+1)(1-\alpha)$, rounded up because ranks are integers. The naive 90% quantile of n scores would undercover, worst at small n.

Takeaway

Small calibration sets force conservative quantiles: with $n = 9$ the 90% interval must stretch to the *largest* observed residual.

The coverage guarantee, and why it is true

Theorem (split conformal coverage)

If the calibration pairs and the test pair are exchangeable, and $\hat{f}$ was fit without touching them, then

$$1-\alpha \;\le\; \Pr\big(Y_{n+1} \in C(X_{n+1})\big) \;\le\; 1-\alpha+\tfrac{1}{n+1},$$

the upper bound requiring almost-surely distinct scores.

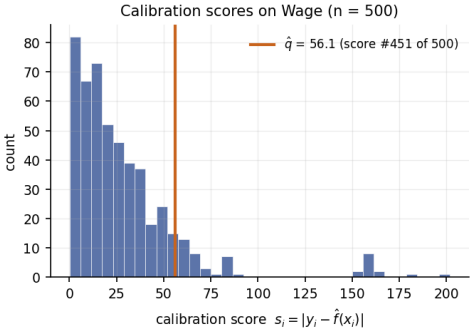
Proof sketch — three lines of rank arithmetic

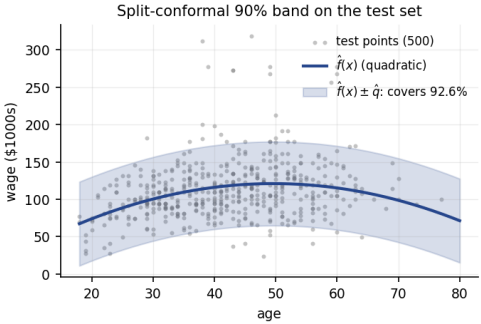
1. Let $s_{n+1} = |Y_{n+1} - \hat{f}(X_{n+1})|$ join the n calibration scores. All $n+1$ scores are exchangeable (each is the same function of its own data point, with $\hat{f}$ fixed).
2. Coverage happens $\iff$ $s_{n+1} \le s_{(k)}$ $\iff$ $\mathrm{rank}(s_{n+1}) \le k$ among the $n+1$ scores.
3. That rank is uniform on $\{1,\dots,n+1\}$, so coverage $= \frac{k}{n+1} \ge \frac{(n+1)(1-\alpha)}{n+1} = 1-\alpha$. (Full argument: appendix.)

How tight is the guarantee at n of 500?

$k = 451 \Rightarrow$ coverage is pinned between 0.9 and $0.9 + \frac{1}{501} = 0.902$ — not just a bound, essentially an *equality*.

Wage: the calibration quantile and the band it buys





Takeaway

Quadratic fit on 2 000 training points; the 451st of 500 sorted calibration residuals gives $\hat{q} = 56.1$. The band $\hat{f}(\text{age}) \pm 56.1$ covers 92.6% of the 500 test wages — and is *narrower* than the miscalibrated Gaussian band (± 66.1, covering 95.2%).

Exercise A3.2 — Computing the conformal quantile by hand [Math]

Exercise

A model is calibrated on $n = 9$ held-out points with absolute residuals

$$2.1, \quad 0.7, \quad 4.0, \quad 1.3, \quad 0.5, \quad 2.9, \quad 1.8, \quad 5.2, \quad 1.1.$$

1. Compute $\hat{q}$ for $\alpha = 0.2$ and write down $C(x)$ if $\hat{f}(x) = 100$.
2. Compute $\hat{q}$ for $\alpha = 0.1$. What do you notice?
3. For $n = 99$ and $\alpha = 0.05$: which order statistic is $\hat{q}$?
4. Exactly which coverage does the theorem guarantee in part 3 (both bounds)?

Solution — Exercise A3.2

Solution

Method. Sort, compute $k = \lceil (n+1)(1-\alpha) \rceil$, pick $s_{(k)}$. Sorted scores: 0.5, 0.7, 1.1, 1.3, 1.8, 2.1, 2.9, 4.0, 5.2.

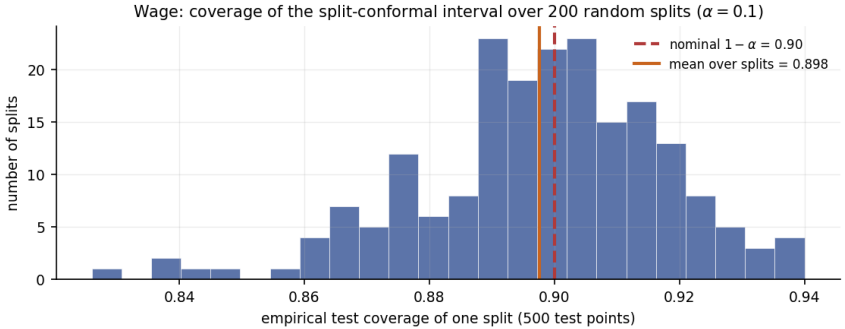
- **Part 1.** $k = \lceil 10 \times 0.8 \rceil = 8 \Rightarrow \hat{q} = s_{(8)} = 4.0$, so $C(x) = [96.0,\ 104.0]$.
- **Part 2.** $k = \lceil 10 \times 0.9 \rceil = 9 \Rightarrow \hat{q} = s_{(9)} = 5.2$ — the *maximum*. With only 9 calibration points, a 90% guarantee forces the most conservative choice available.
- **Part 3.** $k = \lceil 100 \times 0.95 \rceil = 95 \Rightarrow \hat{q}$ is the **95th smallest of 99** scores.
- **Part 4.** Coverage lies in $\left[\frac{95}{100},\ \frac{95}{100} + \frac{1}{100}\right] = [0.95,\ 0.96]$ — read off as $k/(n+1) = 95/100$ and $k/(n+1) + 1/(n+1)$.

Take-away. $\hat{q}$ is always a *single order statistic* — no interpolation, no model, just a sorted list and one index.

Common mistake

Computing the plain 90% empirical quantile ($0.9 \times 9 = 8.1$, so interpolating near $s_{(8)}$). The conformal index uses $n+1$, not n — forgetting the $+1$ is precisely what loses the finite-sample guarantee.

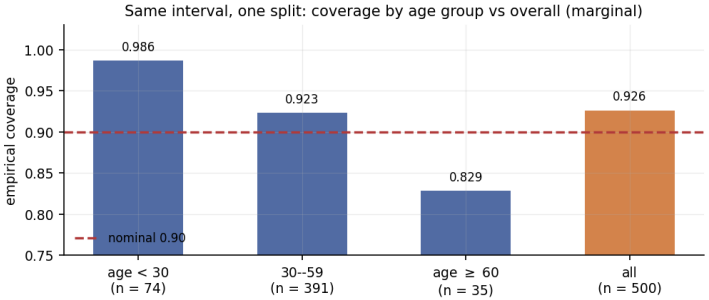
The guarantee is an average over splits — see it hold



Takeaway

200 random splits of Wage: per-split test coverage fluctuates between 0.826 and 0.940 (sd 0.020), but the *mean* is 0.898 $\approx$ 0.9 — the marginal guarantee in action. Any *single* split can land at 0.87; nothing is wrong when it does.

The honest slide: marginal is not conditional



Takeaway

Same interval, same split: coverage is 98.6% for under-30s but only 82.9% for the 60-plus group — the marginal 92.6% is an *average* that hides both. A constant-width band overcovers where wages are predictable and undercovers where they spread out.

Common mistake

Exercise A3.3 — How small can the calibration set be? [Math]

Exercise

The conformal index is $k = \lceil (n+1)(1-\alpha) \rceil$, and $\hat{q} = s_{(k)}$ only exists if $k \leq n$.

1. Show that $k \leq n$ is equivalent to $n \geq \frac{1-\alpha}{\alpha} = \frac{1}{\alpha} - 1$.
2. Evaluate the minimal n for $\alpha = 0.1$, 0.05, and 0.01.
3. A team calibrates with $n = 10$ points at $\alpha = 0.05$. What interval does the recipe return, and why is that the honest answer?
4. Which coverage levels are *exactly* achievable with $n = 9$?

Solution — Exercise A3.3

Solution

Method. Unpack the ceiling: $k \le n \iff (n+1)(1-\alpha) \le n$.

- **Part 1.** $(n+1)(1-\alpha) \le n \iff n+1-\alpha n-\alpha \le n \iff 1-\alpha \le \alpha n \iff n \ge \frac{1-\alpha}{\alpha} = \frac{1}{\alpha} - 1$. ✓
- **Part 2.** $\alpha = 0.1$: $n \ge 9$. $\alpha = 0.05$: $n \ge 19$. $\alpha = 0.01$: $n \ge 99$.
- **Part 3.** $k = \lceil 11 \times 0.95 \rceil = \lceil 10.45 \rceil = 11 > 10$: there is no 11th score, so $\hat{q} = \infty$ and $C(x) = (-\infty, \infty)$. Honest, because with 10 exchangeable scores the new point exceeds *all* of them with probability $1/11 \approx 9.1\% > 5\%$ — no finite cut-off can promise 95%.
- **Part 4.** Achievable exact levels are $\frac{k}{n+1} = \frac{k}{10}$: $10\%, 20\%, \ldots, 90\%, 100\%$ — coverage is *quantized* in steps of $\frac{1}{n+1}$.

Common mistake

Requesting $\alpha = 0.01$ from $n = 50$ calibration points and "fixing" the infinite interval by switching to the plain empirical maximum. The interval is then no longer guaranteed — collect more calibration data instead.

Outline

One width for everyone is a blunt instrument

The absolute-residual band on Wage is ± 56.1 *everywhere*: for 20-year-olds (tightly clustered wages) and for 60-year-olds (wide spread) alike.

Two ways to make the width follow the data

- **Locally weighted scores**: fit a second model $\hat{\sigma}(x)$ for the typical error size, and score

$$s_i = \frac{|y_i - \hat{f}(x_i)|}{\hat{\sigma}(x_i)} \qquad \Longrightarrow \qquad C(x) = \hat{f}(x) \pm \hat{q}\,\hat{\sigma}(x).$$

- **Conformalized quantile regression (CQR)**: model the *quantiles* of $Y \mid X$ directly, then calibrate them.

Both keep the marginal guarantee — any fixed score function works, because the scores of exchangeable points remain exchangeable.

Locally weighted conformal on Wage

$\hat{\sigma}(x)$: quadratic fit of |residual| on age. Result: $\hat{q} = 1.90$ (in σ-units), test coverage 89.6%.

The engine: quantile regression with the pinball loss

Pinball (quantile) loss at level τ

$$\rho_\tau(u) = \begin{cases} \tau\, u & u \geq 0 \\ (\tau - 1)\, u & u < 0 \end{cases} \qquad u = y - q.$$

Minimising $E\,\rho_\tau\big(Y - q(X)\big)$ over functions q yields the conditional τ-quantile of $Y \mid X$.

Why it works — in one sentence

Undershooting costs τ per unit, overshooting costs $1 - \tau$: the optimal $q(x)$ sits where a fraction τ of the mass lies below — exactly the quantile. At $\tau = 0.5$ it is the absolute loss, and we recover the median.

```
from sklearn.ensemble import GradientBoostingRegressor # Ch. 8 engine
lo = GradientBoostingRegressor(loss="quantile", alpha=0.05, # 5% curve
       n_estimators=200, max_depth=2, learning_rate=0.05)
hi = GradientBoostingRegressor(loss="quantile", alpha=0.95, # 95% curve
       n_estimators=200, max_depth=2, learning_rate=0.05)
```

Conformalized quantile regression (CQR)

Algorithm (Romano, Patterson & Candès, 2019)

1. On $\mathcal{D}_{\text{tr}}$, fit quantile curves $\hat{q}_{\text{lo}}$ (level $\alpha/2$) and $\hat{q}_{\text{hi}}$ (level $1-\alpha/2$).
2. On $\mathcal{D}_{\text{cal}}$, score $\boxed{s_i = \max\left(\hat{q}_{\text{lo}}(x_i) - y_i,\ y_i - \hat{q}_{\text{hi}}(x_i)\right)}$
3. $\hat{q} = s_{(k)}$ with the usual $k = \lceil (n+1)(1-\alpha) \rceil$; report $C(x) = \left[\hat{q}_{\text{lo}}(x) - \hat{q},\ \hat{q}_{\text{hi}}(x) + \hat{q}\right]$.

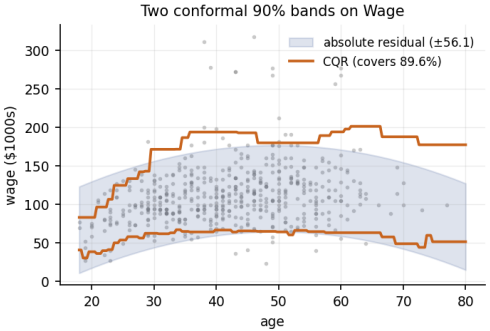
Decoding the score

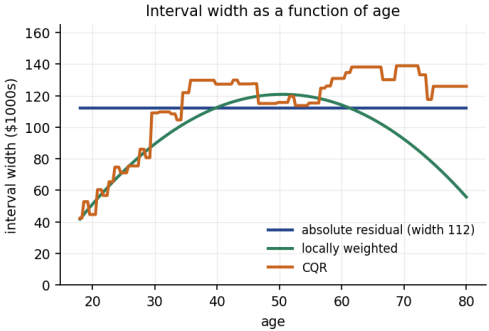
s_i is the *signed distance to the nearer violated edge*: positive if y_i falls outside $[\hat{q}_{\text{lo}}, \hat{q}_{\text{hi}}]$ (by how much), *negative* if inside (how deep). If the fitted band already overcovers, $\hat{q} < 0$ *shrinks* it; if it undercovers, $\hat{q} > 0$ pads it. Same quantile index, same guarantee — the width now lives in the quantile curves.

Takeaway

CQR = "let Chapter 8 draw the band, let conformal calibrate it". On Wage the correction is tiny: $\hat{q} = 0.66$ — the boosted 5%/95% curves were almost calibrated already.

Wage: constant width vs adaptive width





Takeaway

Both bands cover $\approx 90\%$ marginally (CQR: 89.6%), but CQR spends its width where the data spread: average width 70.5 for under-30s vs 134.9 for 60-plus (constant band: 112.3 for both). Note the locally weighted curve *falls* after age 60 — its quadratic $\hat{\sigma}$ extrapolates poorly: the allocation is only as good as the scale model, even though marginal coverage

Extended Exercise A3.1 — CQR on Wage, end to end [Python]

Extended exercise (15 min)

Reproduce the adaptive band. Using the deck's split of `Wage` (seed 2024; 2 000 / 500 / 500):

1. Fit `GradientBoostingRegressor(loss="quantile")` at levels 0.05 and 0.95 on the training set (200 trees, depth 2, learning rate 0.05, `random_state=0`).
2. Compute the CQR scores on the calibration set and $\hat{q}$ at $\alpha = 0.1$; report test coverage.
3. Compare average interval width for ages < 30, 30–59 and ≥ 60 against the constant-width band (112.3).
4. Compare the two bands' *subgroup coverage* for under-30s and 60-plus. Which failure of the constant band does CQR repair, and what does it *not* guarantee?

Run this live

The worked solution sits in `advanced_03_conformal_lab.ipynb` under *Lecture exercises — worked Python solutions*: the cell headed *Extended Exercise A3.1 — CQR on Wage, end to end [Python]*.

Solution — Extended Exercise A3.1 (1/2)

Solution — code: fit / score / calibrate

```
import numpy as np, pandas as pd # data handling
from sklearn.ensemble import GradientBoostingRegressor

wage = pd.read_csv("Wage.csv") # lab: via load()
x, y = wage["age"].to_numpy(float), wage["wage"].to_numpy()
rng = np.random.default_rng(2024) # the deck's split
i = rng.permutation(len(y)); tr, ca, te = i[:2000], i[2000:2500], i[2500:]
X = x.reshape(-1, 1)
gb = dict(n_estimators=200, max_depth=2, learning_rate=0.05, random_state=0)
lo = GradientBoostingRegressor(loss="quantile", alpha=0.05, **gb).fit(X[tr], y[tr])
hi = GradientBoostingRegressor(loss="quantile", alpha=0.95, **gb).fit(X[tr], y[tr])
s = np.maximum(lo.predict(X[ca]) - y[ca], y[ca] - hi.predict(X[ca])) # CQR score
k = int(np.ceil((len(ca) + 1) * 0.9)) # k = 451
qhat = np.sort(s)[k - 1] # 0.66: tiny correction
L, U = lo.predict(X[te]) - qhat, hi.predict(X[te]) + qhat
print(round(qhat, 2), ((y[te] >= L) & (y[te] <= U)).mean()) # 0.66 0.896
```

Solution — Extended Exercise A3.1 (2/2)

Solution — numbers and interpretation

Parts 2–3. $\hat{q} = 0.66$; test coverage $= 0.896$. Average widths:

	age < 30	age 30–59	age ≥ 60
Constant band (abs. residual)	112.3	112.3	112.3
CQR band	70.5	120.4	134.9

Part 4. Subgroup coverage on this split: constant band 0.986 (under-30) vs 0.829 (60-plus); CQR 0.797 vs 0.886. CQR repairs the *width misallocation* — it stops wasting 40 units of width on the young and reinvests them in the old, whose coverage rises from 0.829 to 0.886. **What it does not guarantee:** exact per-group coverage — the young bin ($n = 74$) lands at 0.797 here, within sampling noise of 0.9 but not pinned to it. Only the *marginal* 90% is guaranteed; per-group guarantees need per-group calibration.

Common mistake

Fitting $\hat{q}_{\text{lo}}, \hat{q}_{\text{hi}}$ on training *plus* calibration data "to use all the data". The scores are then not exchangeable with the test score, and the guarantee is void — the split is load-bearing.

Outline

From intervals to prediction sets

For a classifier there is no interval — the honest object is a **set of classes** $C(x) \subseteq \{1, \dots, K\}$.

Split conformal for classification

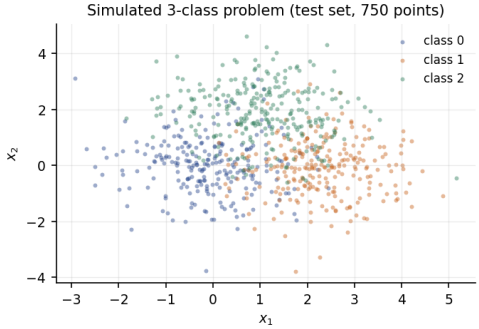
1. Fit a probabilistic classifier $\hat{p}$ on $\mathcal{D}_{\text{tr}}$ (e.g. softmax / logistic regression, Ch. 4).
2. Calibration scores: $\boxed{s_i = 1 - \hat{p}_{y_i}(x_i)}$ — one minus the probability assigned to the *true* class.
3. $\hat{q} = s_{(k)}$, $k = \lceil (n+1)(1-\alpha) \rceil$ as always; report

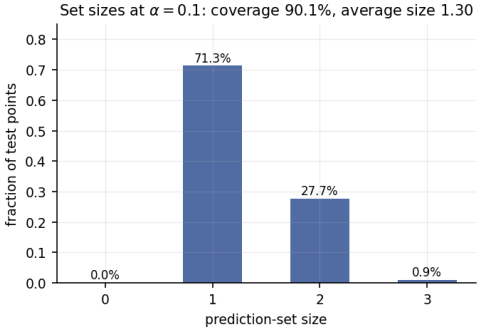
$$C(x) = \{\, k : \ \hat{p}_k(x) \geq 1 - \hat{q} \,\}.$$

Decoding the set rule

A class enters the set when its softmax probability clears the threshold $1 - \hat{q}$. Confident points get **singletons**; ambiguous points near a decision boundary get $\{1, 2\}$ or bigger. The guarantee: the *true* class is in $C(X)$ with probability $\geq 1 - \alpha$ — same rank argument, unchanged.

A 3-class simulation: sets earn their keep at the boundaries

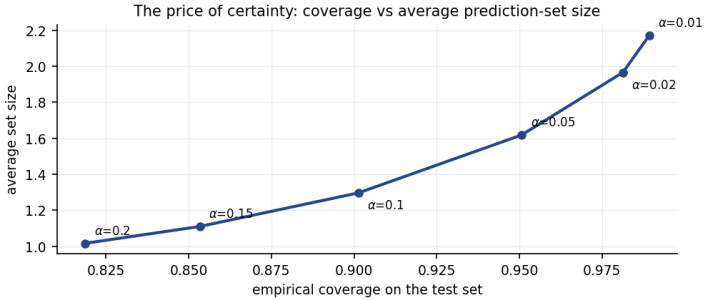




Takeaway

Three Gaussian classes (centres $(0,0)$, $(2.2,0)$, $(1.1,1.9)$; logistic accuracy 80.9%). At $\alpha = 0.1$: $\hat{q} = 0.7685$ (score 676 of 750), coverage 90.1%, average set size 1.30 — 71.3% singletons, 27.7% pairs, 0.9% full sets. The set *size* is a per-point difficulty meter the accuracy number hides.

The price of certainty: coverage vs set size



Takeaway

Demanding 99% coverage inflates the average set from 1.30 to 2.17 classes — two-thirds of the label space. Certainty is bought with set size; pick α by what a large set *costs downstream* (a manual review, a second lab test).

One slide on APS (adaptive prediction sets)

Exercise A3.4 — Reading prediction sets [Concept]

Exercise

A conformal classifier ($K = 3$, $\alpha = 0.1$) was calibrated with $\hat{q} = 0.7685$, so classes enter the set when $\hat{p}_k(x) \geq 0.2315$. Three test customers get softmax probabilities:

A: (0.95, 0.03, 0.02) B: (0.50, 0.45, 0.05) C: (0.40, 0.35, 0.25)

1. Write down the prediction set of each customer.
2. The plain classifier predicts class 0 for all three. What extra information do the sets carry?
3. Can the guarantee be read as "90% of *singleton* sets are correct"? Why not?
4. Under which circumstances can the rule return an *empty* set, and what should a production system do with one?

Solution — Exercise A3.4

Solution

Method. Threshold each probability at $1 - \hat{q} = 0.2315$.

- **Part 1.** A: $\{0\}$ (0.95 clears, others do not). B: $\{0, 1\}$ (0.50 and 0.45 clear). C: $\{0, 1, 2\}$ (all three ≥ 0.2315).
- **Part 2.** Identical point predictions, very different epistemic states: A is settled, B is a genuine two-way toss-up, C is anyone's guess. The set size is the honest "how sure are we" channel a point prediction lacks.
- **Part 3.** No. The guarantee is *marginal over all points*: $\geq 90\%$ of *all* sets contain the truth. Singletons are the *easy* points, so their hit rate is typically above 90%, while the large sets carry more of the mistakes — but neither is individually guaranteed.
- **Part 4.** Empty sets occur when even the largest probability misses the threshold: possible once $\hat{q} < 1 - \max_k \hat{p}_k(x)$, i.e. with loose α and diffident models (our simulation: 0.9% empty at $\alpha = 0.2$, none at $\alpha = 0.1$). Production systems treat empty (like oversized) sets as **abstentions**: route to a human.

Common mistake

Outline

Same data, two intervals: when the model is right, they agree

Homoskedastic simulation — the friendly case

$y = 1 + 2x + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, 2^2)$, $x \sim U(0, 10)$; 2 000 training, 1 000 calibration, 1 000 test points (seed 2024). Theory: 90% half-width $= 1.645 \times 2 = 3.29$.

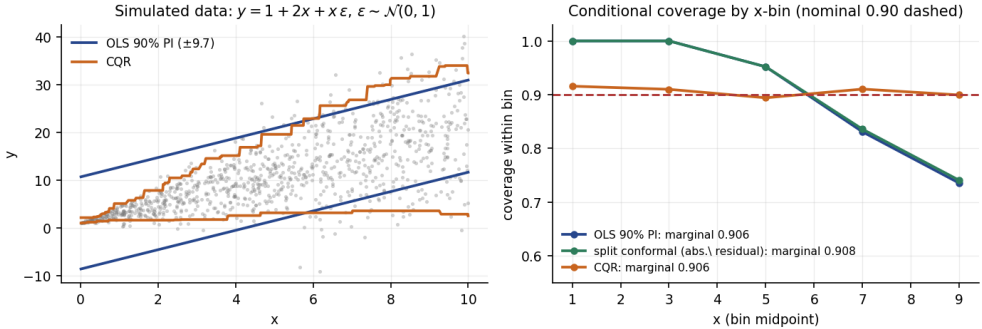
	half-width	test coverage
OLS prediction interval (Ch. 3)	3.31	0.915
Split conformal (abs. residual)	3.24	0.909

Takeaway

When the Gaussian linear model is *true*, conformal costs essentially nothing: both intervals reproduce the theoretical ± 3.29. Conformal is not a better interval — it is the same interval with a *weaker warranty requirement*.

In industry — Model risk management: two numbers as one sanity check

Heteroskedastic truth: where the 90% quietly stops being 90%



Takeaway

$y = 1 + 2x + x\,\varepsilon$: noise grows with x. All three methods hold $\approx 90\%$ *marginally* (OLS 0.906, conformal 0.908, CQR 0.906) — but conditionally the constant-width pair collapses to 73.5% (74.1%) for $x > 8$ while overcovering at 100% for $x < 2$. Only CQR stays near 0.90 in *every* bin (0.916 to 0.899). And only conformal *guarantees* its marginal number.

The failure that actually happens: plug-in intervals from a flexible model

Same heteroskedastic data — boosted regression

Fit a 500-tree gradient boosting model, then do what practitioners actually do: $\hat{f}(x) \pm 1.645 \times \hat{\sigma}_{\text{in}}$ with $\hat{\sigma}_{\text{in}}$ the sd of the *training* residuals.

	$\hat{\sigma}$ used	claimed	actual test coverage
Plug-in (training residuals)	3.92	90%	77.6%
Split conformal (same model)	$\hat{q} = 10.81$	$\geq 90\%$	90.7%

The truth: held-out residual sd is 6.12 — the model memorised a third of the noise, and the plug-in interval inherited the optimism.

Common mistake

Calibrating interval width on data the model has seen. Training residuals of a flexible model are systematically too small; every per-cent of memorised noise is a per-cent of silent undercoverage. The conformal split exists precisely to make this mistake impossible.

Chapter 3's interval vs split conformal — the comparison slide

	OLS prediction interval	**Split conformal**
Assumes	linear mean, Gaussian, homoskedastic errors	exchangeability + a clean split
Guarantee	exact at every x, *if* the model is true	marginal $\geq 1-\alpha$, finite-sample, *whatever* the model
Works with	the linear model it was derived for	any $\hat{f}$: boosting, nets, black boxes
Width	$\approx$ constant ($\hat{\sigma}$-driven; slightly wider at high leverage)	constant (abs. residual) or adaptive (CQR)
Cost	none	holds out n calibration points
Under misspecification	silently wrong coverage	coverage intact; only width reacts

Takeaway

Use the OLS interval when you *believe* the Gaussian linear model and want conditional statements. Use conformal when the model is flexible, the errors are ugly, or the coverage number will be *audited*.

Extended Exercise A3.2 — Stress-testing the two intervals [Integrative]

Extended exercise (15 min)

Simulate $y = 1 + 2x + x\,\varepsilon$, $\varepsilon \sim \mathcal{N}(0,1)$, $x \sim U(0,10)$, $n = 4\,000$ (split 2 000 / 1 000 / 1 000, seed 2024).

1. Fit OLS on the training set; build the 90% prediction interval $\hat{y} \pm 1.645\,\hat{\sigma}$ and measure test coverage overall, for $x < 2$, and for $x > 8$.
2. Build the split-conformal interval (absolute residuals, $\alpha = 0.1$) and measure the same three coverages. What does conformal fix here — and what does it not fix?
3. Fit a 500-tree `GradientBoostingRegressor`; report coverage of $\hat{f} \pm 1.645 \times \text{sd(training residuals)}$, then conformalise the same model. Explain the mechanism behind the gap.
4. Conclude: which of the three failure modes (skewed errors, heteroskedastic errors, optimistic $\hat{\sigma}$) does split conformal repair marginally, and which needs CQR?

Run this live

Worked solution in `advanced_03_conformal_lab.ipynb` under *Lecture exercises — worked Python solutions*, cell *Extended Exercise A3.2*.

Solution — Extended Exercise A3.2 (1/2)

Solution — code: the three intervals

```
import numpy as np, statsmodels.api as sm
from sklearn.ensemble import GradientBoostingRegressor
rng = np.random.default_rng(2024)
N = 4000; x = rng.uniform(0, 10, N); y = 1 + 2*x + x*rng.normal(0, 1, N)
tr, ca, te = np.arange(2000), np.arange(2000, 3000), np.arange(3000, 4000)
ols = sm.OLS(y[tr], sm.add_constant(x[tr])).fit() # 1. classical PI
half = 1.645 * np.sqrt(ols.scale) # pooled sigma: 5.87
pred = ols.params[0] + ols.params[1] * x[te]
cov = lambda hit: [hit.mean(), hit[x[te] < 2].mean(), hit[x[te] > 8].mean()]
print(cov(np.abs(y[te] - pred) <= half)) # [0.906, 1.000, 0.735]
k = int(np.ceil(1001 * 0.9)) # 2. conformal: k = 901
q = np.sort(np.abs(y[ca] - (ols.params[0] + ols.params[1]*x[ca])))[k-1]
print(q, cov(np.abs(y[te] - pred) <= q)) # 9.8 [0.908, 1.0, 0.741]
f = GradientBoostingRegressor(n_estimators=500, random_state=0) # 3. plug-in
f.fit(x[tr, None], y[tr]); sig = np.std(y[tr] - f.predict(x[tr, None])) # 3.92
print(cov(np.abs(y[te] - f.predict(x[te, None])) <= 1.645 * sig)) # [0.776, ...]
qg = np.sort(np.abs(y[ca] - f.predict(x[ca, None])))[k-1] # 10.81
print(cov(np.abs(y[te] - f.predict(x[te, None])) <= qg)) # [0.907, ...]
```

Solution — Extended Exercise A3.2 (2/2)

Solution — interpretation

Part 1. OLS: marginal 0.906, but 100% for $x < 2$ and 73.5% for $x > 8$ — the pooled $\hat{\sigma} = 5.87$ is one compromise width for noise that ranges from ≈ 0 to 10. **Part 2.** Conformal ($\hat{q} = 9.79$): marginal 0.908, subgroups 1.00 / 0.741 — conformal *guarantees the marginal number* (over repeated data: 0.900 vs OLS's drifting 0.894) but a constant-width score cannot repair the *conditional* imbalance. That is CQR's job (it reaches 0.899 at $x > 8$). **Part 3.** Plug-in GBR: claimed 90%, actual 77.6% — training-residual sd (3.92) understates held-out error sd (6.12) because the 500-tree model partly memorised the training noise. Conformalising the *same* model restores 90.7% by pricing the width on unseen data ($\hat{q} = 10.81$). **Part 4.** Marginal repairs come free (skew, heteroskedasticity, optimism all handled — the guarantee never moved); *conditional* honesty across x is the one thing that needs an adaptive score (CQR / locally weighted).

Common mistake

Concluding from part 2 that conformal "failed". It delivered exactly what it promises — marginal coverage; expecting per-region coverage from a constant-width score is a specification error, not a method error.

Outline

Pitfall 1: exchangeability violations in the wild

- **Time series**: calibration residuals from 2019–2024, deployment in 2026 — trends, seasonality and regime shifts make the test point *special*, which is exactly what exchangeability forbids.
- **Covariate shift**: a marketing model calibrated on organic traffic, deployed during a paid campaign — the X-mix changed overnight.
- **Selection feedback**: the model approves loans, and only approved loans generate future calibration labels.

What to do instead

Time series: calibrate on the *most recent* block, recalibrate on a schedule, or use online/adaptive conformal that tunes α from realised coverage. Known covariate shift: *weighted* conformal with likelihood-ratio weights $w(x) = p_{\text{new}}(x)/p_{\text{cal}}(x)$. Always: monitor realised coverage — the guarantee you can check is the one you keep.

In industry — Energy trading: recalibrate with the seasons

A wind-power desk recalibrates its forecast intervals *daily* on the trailing 30 days. A fixed calibration set

Pitfall 2: leakage between fitting and calibration

The guarantee has exactly one hard requirement inside your pipeline: *nothing about $\hat{f}$ may depend on the calibration set.*

The four common leak channels

1. **Preprocessing**: scaler / imputer / encoder fit on *all* data before the split.
2. **Tuning**: hyperparameters or early stopping selected on calibration performance.
3. **Feature selection**: screening variables on the full sample (the Ch. 5 classic).
4. **Duplicates**: the same customer's rows on both sides of the split.

Each one shrinks calibration scores below test scores $\Rightarrow$ $\hat{q}$ too small $\Rightarrow$ silent undercoverage — the plug-in GBR's 77.6% was this mechanism in its purest form.

Common mistake

"Refitting on train + calibration after computing $\hat{q}$, since the interval is already fixed." The deployed $\hat{f}$ is then not the one that generated the scores — the pair $(\hat{f}, \hat{q})$ is calibrated *together* or not at all.

From ten lines to production

Practical sizing rules

- **Feasibility**: α needs $n \geq \frac{1}{\alpha} - 1$ (19 points for 95%, 99 for 99%) — below that the honest interval is infinite (Exercise A3.3).
- **Stability**: with $n = 500$ the coverage of one split has sd $\approx 1.3\%$ (appendix); n between 500 and 1 000 is the usual sweet spot — beyond that, spend the data on training.
- **Report the nature of the guarantee**: marginal, at the stated α, under exchangeability — three words that pre-empt three misreadings.

In industry — Tooling: MAPIE and friends

In production, teams reach for **MAPIE** (*Model Agnostic Prediction Interval Estimator*, scikit-learn compatible): split/cross conformal, CQR and classification sets behind one API, plus jackknife+ variants that recycle training folds. It is not installed here — deliberately: everything in this module was ten lines from scratch, and those ten lines are the audit trail you will be asked for.

Outline

Python lab — run it live in the notebook

Companion notebook: `advanced_03_conformal_lab.ipynb`

The code in this section is meant to be **demonstrated live**. Open the companion Jupyter notebook and run it cell by cell:

- §1, *Split conformal on Wage* — the ten-line recipe: $\hat{q} = 56.14$, coverage 0.926;
- §2, *Coverage over repeated splits* — 200 splits, mean 0.898;
- §3, *Adaptive intervals* — locally weighted scores and CQR, width profiles by age;
- §4, *Conformal classification* — 3-class simulation, prediction sets, size distribution;
- §5, *OLS prediction interval vs conformal* — the heteroskedastic stress test.

Data loads via the `ISLP` package or the bundled CSV files; §6, *Exercises* ends the notebook with hands-on tasks.

Split conformal in ten lines

```
import numpy as np, pandas as pd
from sklearn.linear_model import LinearRegression

wage = pd.read_csv("Wage.csv") # lab: via load()
x, y = wage["age"].to_numpy(float), wage["wage"].to_numpy()
rng = np.random.default_rng(2024) # reproducible split
i = rng.permutation(len(y)); tr, ca, te = i[:2000], i[2000:2500], i[2500:]
P = lambda a: np.column_stack([a, a**2]) # quadratic in age
f = LinearRegression().fit(P(x[tr]), y[tr]) # 1. fit on TRAIN only
s = np.abs(y[ca] - f.predict(P(x[ca]))) # 2. calibration scores
k = int(np.ceil((len(ca) + 1) * (1 - 0.1))) # 3. k = ceil((n+1)(1-a))
qhat = np.sort(s)[k - 1] # k-th smallest score
cov = (np.abs(y[te] - f.predict(P(x[te]))) <= qhat).mean() # 4. check
print(k, qhat.round(2), cov) # 451 56.14 0.926
```

Exercise A3.5 — Conformal classification from scratch [Python]

Exercise

Build prediction sets for the deck's 3-class simulation (three Gaussian classes, 1 000 points each, centres $(0,0)$, $(2.2,0)$, $(1.1,1.9)$, unit variance, seed 2024; split 1 500 / 750 / 750).

1. Fit `LogisticRegression` on the training set.
2. Compute the calibration scores $s_i = 1 - \hat{p}_{y_i}(x_i)$ and $\hat{q}$ at $\alpha = 0.1$ (which k?).
3. Build the test-set prediction sets and report coverage, average set size, and the share of singleton sets.
4. Re-run with $\alpha = 0.01$: what happens to the average set size?

Run this live

Worked solution in `advanced_03_conformal_lab.ipynb` under *Lecture exercises — worked Python solutions*, cell *Exercise A3.5*.

Solution — Exercise A3.5

Solution — code and expected numbers

```
import numpy as np
from sklearn.linear_model import LogisticRegression
rng = np.random.default_rng(2024)
mu = np.array([[0, 0], [2.2, 0], [1.1, 1.9]]) # class centres
X = np.vstack([rng.normal(m, 1.0, (1000, 2)) for m in mu])
y = np.repeat([0, 1, 2], 1000)
p = rng.permutation(3000); X, y = X[p], y[p] # shuffle once
clf = LogisticRegression(max_iter=1000).fit(X[:1500], y[:1500])
pc = clf.predict_proba(X[1500:2250]) # calibration probs
s = 1 - pc[np.arange(750), y[1500:2250]] # 1 - p_true
k = int(np.ceil(751 * 0.9)) # k = 676
qhat = np.sort(s)[k - 1] # 0.7685
sets = clf.predict_proba(X[2250:]) >= 1 - qhat # threshold 0.2315
cov = sets[np.arange(750), y[2250:]].mean() # 0.9013
print(qhat.round(4), cov, sets.sum(1).mean().round(3)) # 0.7685 0.9013 1.296
```

Parts 3–4. Coverage 0.9013, average size 1.30, singletons 71.3%. At $\alpha = 0.01$: $k = \lceil 751 \times 0.99 \rceil = 744$, $\hat{q} = 0.9638$, coverage 98.9% — and the average set grows to 2.17 of 3 classes.

Outline

Module A3 in one slide

What we accomplished

Turned any point predictor from this course into a set-valued predictor with a finite-sample coverage guarantee.

- Exchangeability — the single assumption, and what breaks it.
- Split conformal: scores, $k = \lceil (n+1)(1-\alpha) \rceil$, interval $\hat{f}(x) \pm \hat{q}$; proof by rank arithmetic.
- The guarantee is *marginal* — averaged over points and splits, not per subgroup.
- Adaptive widths: locally weighted scores and conformalized quantile regression.
- Classification: prediction sets, set size as a difficulty meter, APS.
- Versus Chapter 3: agreement under the Gaussian model, guarantees under misspecification, and the plug-in trap.

Key formulas at a glance

Conformal index $k = \lceil (n+1)(1-\alpha) \rceil$; $\hat{q} = s_{(k)}$; finite iff $n \geq \frac{1}{\alpha} - 1$.

Regression interval $s_i = |y_i - \hat{f}(x_i)|$; $C(x) = [\hat{f}(x) - \hat{q},\ \hat{f}(x) + \hat{q}]$.

Coverage guarantee $1 - \alpha \ \leq\ \Pr(Y_{n+1} \in C(X_{n+1})) \ \leq\ 1 - \alpha + \frac{1}{n+1}$ (distinct scores).

Locally weighted $s_i = |y_i - \hat{f}(x_i)| / \hat{\sigma}(x_i)$; $C(x) = \hat{f}(x) \pm \hat{q}\,\hat{\sigma}(x)$.

CQR $s_i = \max(\hat{q}_{\text{lo}}(x_i) - y_i,\ y_i - \hat{q}_{\text{hi}}(x_i))$; $C(x) = [\hat{q}_{\text{lo}}(x) - \hat{q},\ \hat{q}_{\text{hi}}(x) + \hat{q}]$.

Classification set $s_i = 1 - \hat{p}_{y_i}(x_i)$; $C(x) = \{k : \hat{p}_k(x) \geq 1 - \hat{q}\}$.

Pinball loss $\rho_\tau(u) = \tau u\,[u \geq 0] + (\tau - 1) u\,[u < 0]$ — minimised by the τ-quantile.

Vocabulary checklist

Term	Meaning
Exchangeability	every ordering of the data equally likely (weaker than i.i.d.)
Calibration set	held-out data that prices the interval width
Conformity score	how badly a point disagrees with the model's prediction
$\hat{q}$	the $\lceil (n+1)(1-\alpha) \rceil$-th smallest calibration score
Marginal coverage	$\geq 1-\alpha$ on average over points and splits
Conditional coverage	coverage at a specific x / subgroup — *not* guaranteed
Locally weighted CP	residuals scaled by a fitted error-size model
CQR	conformalized quantile regression — calibrated quantile band
Prediction set	set of classes guaranteed to contain the truth
APS	adaptive prediction sets via cumulative probability mass
Weighted / Mondrian CP	repairs for covariate shift / per-group guarantees

Decision rules of thumb

- Any black-box model, honest uncertainty needed $\Rightarrow$ **split conformal** — ten lines, one held-out slice.
- Spread visibly varies with $x \Rightarrow$ **CQR** (or locally weighted scores); constant width wastes coverage on the easy region.
- Classifier feeding a triage / review queue $\Rightarrow$ **prediction sets**; route big or empty sets to a human.
- Believe the Gaussian linear model and need per-x statements $\Rightarrow$ Chapter 3's interval is fine — and conformal is a free cross-check.
- Time series or known shift $\Rightarrow$ recent-block calibration, weighted or online conformal; *never* vanilla split conformal untouched.
- Sizing: $n_{\text{cal}} \approx 500$–$1\,000$; feasibility floor $n \geq 1/\alpha - 1$.
- A subgroup matters on its own $\Rightarrow$ calibrate within the subgroup.

Common pitfalls

Don't

1. Tune hyperparameters, early stopping, scalers or features on the calibration set — that is leakage, and it undercovers silently.
2. Read "90% marginal" as "90% for every customer segment".
3. Use training residuals as calibration scores (the 77.6% trap).
4. Apply vanilla split conformal to time series and call it guaranteed.
5. Demand $\alpha = 0.01$ from 50 calibration points — infeasible by arithmetic.
6. Keep last year's $\hat{q}$ after retraining the model or after the world drifts — the pair $(\hat{f}, \hat{q})$ is calibrated together.

Self-check questions

Can you answer these without looking back?

1. Why $\lceil (n+1)(1-\alpha) \rceil$ and not the plain $(1-\alpha)$-quantile of the n scores?
2. State the two-sided coverage bound. Where does each side come from?
3. Your single split shows coverage 0.87 at $\alpha = 0.1$. Is the method broken?
4. What exactly does CQR's $\max(\hat{q}_{\text{lo}} - y,\ y - \hat{q}_{\text{hi}})$ score measure, and when is $\hat{q}$ negative?
5. Why can a prediction set be empty under the $1 - \hat{p}$ score, and which score family avoids that?
6. Name the four leakage channels between fitting and calibration.

Where to look

1–2: Split Conformal section (and the appendix proof). 3: the coverage histogram. 4: Adaptive Intervals. 5: the APS slide. 6: Pitfall 2.

Appendix — what is in here, and why

Nothing in the appendix is needed to follow the module: each slide is a formal derivation, a longer worked exercise, or a side topic. Read it when you want the full story, or when a later chapter sends you back here.

Contents of the appendix

- **The coverage theorem, proved in full** — the rank argument with both bounds and the tie-breaking detail; the main deck only sketched it.
- **How coverage fluctuates: the Beta law** — the exact distribution of one split's coverage, moved out because it needs the Beta distribution.
- **Full conformal and jackknife+** — what you gain and pay when you refuse to split; a side topic beyond the split recipe.
- **Mondrian (group-conditional) conformal** — the fix for the subgroup problem; deferred until the marginal story was complete.
- **A drill exercise on the Beta law** — connects the histogram of the main deck to the formula here.

The coverage theorem, proved in full

Setup. Scores $s_1, \ldots, s_n$ (calibration) and s_{n+1} (test) are exchangeable; $k = \lceil (n+1)(1-\alpha) \rceil$; coverage event $A = \{s_{n+1} \le s_{(k)}\}$ with $s_{(k)}$ the k-th smallest of $s_1, \ldots, s_n$.

Step 1 — the event is a rank statement

Assume first no ties. Let r be the rank of s_{n+1} among all $n+1$ scores. Exactly $r-1$ calibration scores lie below s_{n+1}, hence $s_{n+1} \le s_{(k)} \iff r - 1 \le k - 1 \iff r \le k$.

Step 2 — the rank is uniform

Exchangeability makes every assignment of the $n+1$ scores to the $n+1$ ranks equally likely, so $\Pr(r = j) = \frac{1}{n+1}$ for each j and $\Pr(A) = \frac{k}{n+1}$.

Step 3 — both bounds

Lower: $k \ge (n+1)(1-\alpha) \Rightarrow \Pr(A) \ge 1 - \alpha$. Upper: $k < (n+1)(1-\alpha) + 1 \Rightarrow \Pr(A) < 1 - \alpha + \frac{1}{n+1}$. **Ties:** with ties, break them by adding i.i.d. uniform jitter to every score — exchangeability is preserved, the jittered interval is (weakly) inside the original one, so the *lower* bound survives; only the upper bound needs the distinct-scores assumption.

How coverage fluctuates from split to split: the Beta law

Conditional coverage of one calibration set

Given the calibration set, the interval's true coverage is $F(\hat{q})$, where F is the score distribution. For continuous F, $F(s_{(k)}) \sim \text{Beta}(k,\ n+1-k)$, hence

$$E\big[F(\hat{q})\big] = \frac{k}{n+1}, \qquad \text{sd} = \sqrt{\frac{k\,(n+1-k)}{(n+1)^2(n+2)}}.$$

Predicting the Wage histogram before running it

$n = 500$, $k = 451$: mean $= \frac{451}{501} = 0.9002$, sd $= 0.0134$. One split's *measured* coverage adds test-set noise ($\sqrt{0.9 \times 0.1/500} = 0.0134$); combined: $\sqrt{0.0134^2 + 0.0134^2} = 0.019$. Observed over 200 splits: sd $= 0.0198$. Theory and histogram agree to the third decimal.

Takeaway

The guarantee fixes the *mean* of the coverage distribution; the Beta law fixes its *spread*.

Refusing to split: full conformal and jackknife+

Method	Idea	Model fits	Guarantee
Split conformal	one split, one fit	1	$\geq 1-\alpha$
Full conformal	for every candidate y, refit including (x, y) and test its score's rank	one per candidate y (a grid)	$\geq 1-\alpha$
Jackknife+	n leave-one-out fits; intervals from LOO residuals	n	$\geq 1-2\alpha$
CV+	K-fold version of jackknife+	K	$\geq 1-2\alpha$

Why this module taught the split version

Full conformal uses every point for both fitting and calibration — statistically the most efficient, computationally absurd for boosting or neural nets. Jackknife+/CV+ sit in between, with a weaker worst-case constant ($1-2\alpha$; in practice close to $1-\alpha$). Split conformal costs one held-out slice and *one* model fit — which is why it is the production default and the MAPIE default.

Mondrian conformal: buying back the subgroups

Group-conditional (Mondrian) conformal

Partition the space into groups $g(x)$ fixed *in advance* (age bands, product lines, hospitals). Calibrate a separate $\hat{q}_g$ from the calibration scores of each group. Then, for every group, $\Pr\left(Y \in C(X) \mid g(X) = g\right) \geq 1 - \alpha$.

The price

Each group needs its own feasible calibration count ($n_g \geq \frac{1}{\alpha} - 1$, and comfortably more for stable widths — the Beta law now applies *per group*). On Wage, the 60-plus group had only 35 test points and ~ 50 calibration points per split at $\alpha = 0.1$: feasible, but with visibly noisy widths. Fully *conditional* coverage at every single x is provably impossible distribution-free — groups are the honest middle ground.

Takeaway

Marginal coverage is free; group coverage costs calibration data per group; pointwise cover-

Appendix Exercise A3.A — The Beta law in your hands [Math]

Exercise

A team calibrates with $n = 99$ points at $\alpha = 0.1$.

1. Compute k and the exact expected coverage $k/(n+1)$.
2. Using the Beta law, compute the sd of the true coverage of one calibration draw. (Hint: $\text{sd} = \sqrt{k(n+1-k)}/\big((n+1)\sqrt{n+2}\big)$.)
3. The team wants the sd below 0.01. Roughly which n do they need (use $\text{sd} \approx \sqrt{\alpha(1-\alpha)/n}$)?

Solution — Appendix Exercise A3.A

Solution

Method. Plug into the Beta$(k,\ n+1-k)$ moments.

- **Part 1.** $k = \lceil 100 \times 0.9 \rceil = 90$; expected coverage $= \frac{90}{100} = 0.900$ exactly.
- **Part 2.** sd $= \sqrt{90 \times 10}\,/\,\big(100\sqrt{101}\big) = 30/1004.99 \approx 0.030$: a single calibration draw's true coverage typically sits anywhere in ≈ 0.87–0.93.
- **Part 3.** $\sqrt{0.09/n} \le 0.01 \iff n \ge 900$. Nine hundred calibration points buy percent-level stability — consistent with the deck's 500–1 000 rule of thumb ($n = 500$ gave sd 0.0134).

Common mistake: treating the guarantee's *mean* statement as if it removed this spread. The theorem constrains the average over calibration draws; the Beta law says how far your *one* draw can sit from it.