

Prologue: data, models, and learning

Hicham Janati

Fall 2026

This prologue sketches the arc of the course: from a table of numbers to a model of the process that produced it, and from models of processes to machines that predict. Each idea returns later, in this course or in its follow-up on time series and uncertainty.

1 Data is multivariate

A shop tracks its loyalty-card customers. Its database holds one row per customer:

	visits / month	spend / month (Dh)	days since last visit
customer 1	11	480	3
customer 2	7	310	8
customer 3	16	690	1
customer 4	12	520	4
customer 5	5	260	12

Each row collects several quantities measured simultaneously on the same customer: an *observation* is not a number but a vector,

$$\mathbf{x}_i = (x_{i1}, \dots, x_{id})^\top \in \mathbb{R}^d, \quad i = 1, \dots, n, \quad (1)$$

and the table stacks the n observations as the rows of a matrix in $\mathbb{R}^{n \times d}$ (here $d = 3$). The notation in (1) fixes a convention used throughout: bold letters ($\mathbf{x}_i$) denote vectors, plain italic letters denote one-dimensional quantities, such as the components x_{ij} . Frequent visitors spend more: the columns move together, and the way they move together is often the information sought. Segmenting the customers into profiles, or spotting the odd ones, is done on combinations of columns, not on any column alone.

Tables of this kind are everywhere. A bank records card transactions (amount, hour, merchant category, minutes since the previous transaction), and fraud lives in the combination: a large electronics purchase at 3 a.m., seconds after the previous transaction, where each feature taken alone is unremarkable. A sales team keeps one row per day (price, promotion, units sold); an insurer one row per claim (amount, delay, category); a biology lab one row per patient, with thousands of gene-expression levels that co-vary in pathways. In every case the natural unit is the row $\mathbf{x}_i \in \mathbb{R}^d$, with d ranging from a handful to many thousands.

2 From columns to a joint model

First-year statistics equips us to model a single column: draw its histogram, fit a distribution, compute quantiles. Nothing prevents applying that recipe to the shop's table one column at a time, and figure 1 shows the result for its 400 customers: three histograms, one per column, each perfectly legible on its own. Visits center near 12 per month, spend near 530 Dh, and the days since the last visit pile up early (84% of customers came within the last week), with a long right tail of absentees. For a question that concerns one column alone (what fraction of customers visits more than 15 times a month?), these plots are all that is needed.

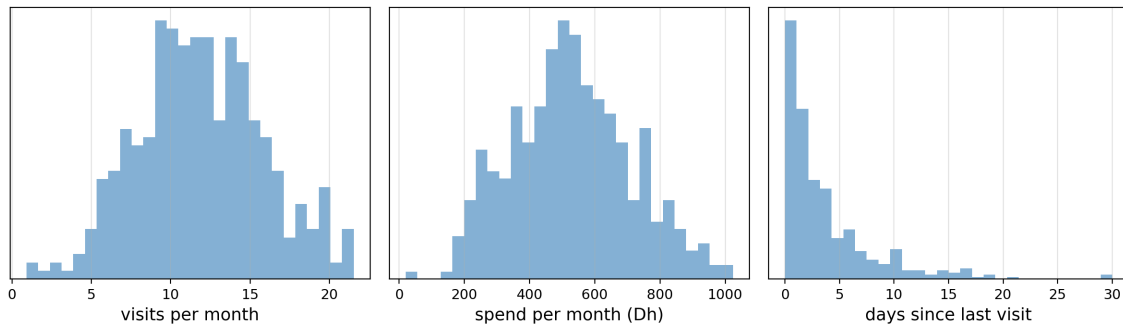


Figure 1: The shop’s table, one column at a time: a histogram per column for the 400 customers. Each marginal distribution is legible on its own; nothing in the three plots reveals how the columns move together.

The shop’s questions, however, concern the columns *jointly*, and the column-by-column view is blind to precisely that. Figure 2 plots the same customers by the first two columns together, visits per month against monthly spend. The two columns are strongly dependent, since customers who come often spend more in total. The customer marked in red makes 17 visits (more than 90% of customers, but nowhere near a record) and spends 300 Dh (more than 12% of customers, again well inside the observed range). Examined column by column, this customer triggers nothing: both red lines fall well inside their histograms, the very histograms of figure 1. Examined as a *pair*, the same customer sits far outside the cloud: someone who visits every other day yet buys almost nothing, a combination the process essentially never produces. Whether this signals fraud, a malfunctioning loyalty card, or a very patient window-shopper is a business question; that it must be *flagged* at all is visible only jointly.

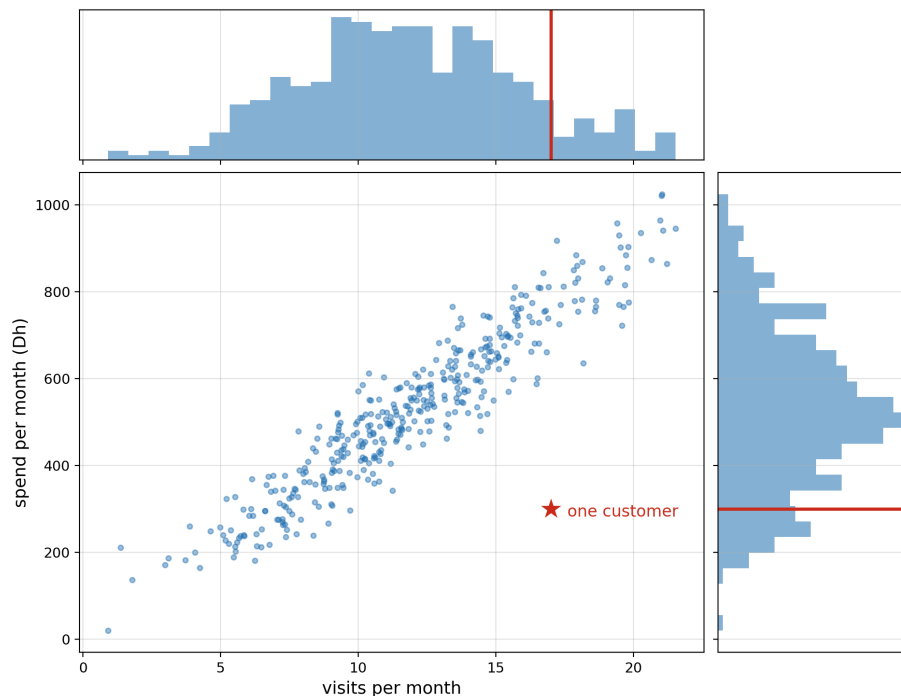


Figure 2: Four hundred customers described by two columns. The customer marked in red is unremarkable in each marginal histogram (red lines) yet far from the cloud as a pair: univariate screening, column by column, cannot flag what the joint view makes obvious.

In general, the marginal distributions of the columns do not determine the distribution of the

row: infinitely many joint distributions share the same d marginals and differ only in how the columns are tied together. The row must therefore be treated as a single object (the vector $\mathbf{x}_i$ of (1)), and the object to model is its *joint* distribution.

A table of n rows is, by itself, only a record of the past. To *model* it is to make a decision: to treat the rows as the output of a data-generating process, in the simplest and most common form,

$$\mathbf{x}_1, \dots, \mathbf{x}_n \stackrel{\text{i.i.d.}}{\sim} P, \quad (2)$$

independent draws of a *random vector* $\mathbf{X}$ with a single unknown probability distribution P on $\mathbb{R}^d$, identically distributed because the process is assumed stable across rows. The abbreviation i.i.d. (*independent and identically distributed*) will be used constantly in this course; when P admits a density on $\mathbb{R}^d$, we write it f .

The decision is consequential because a distribution can be interrogated in ways a table cannot. If P were known, we could compute the probability $\mathbb{P}(\mathbf{X} \in A)$ of any future event A (a transaction above 5 000 Dh tonight), and summarize $n \times d$ recorded numbers by the few parameters that pin down P . Five uses recur throughout the course: describe the *typical* rows; read the *dependence* between columns; flag any new row as *anomalous* under the process (the red customer of figure 2 would receive a near-zero density); fill in a *missing* entry from the observed ones, using that dependence; and *generate* synthetic rows indistinguishable in law from the real ones. A single distribution serves all five purposes.

The first part of this course follows this program: choose a family of candidate distributions on $\mathbb{R}^d$, use the table to determine which member of the family the process resembles, and then interrogate it. Whether the assumptions behind the program (rows independent, process stable, family adequate) hold is examined from chapter 1 onward.

3 Predicting one column from the others

Modeling the full joint distribution is the most complete ambition, and the most expensive one. Often the question at hand is narrower. One column has a special status (the quantity to be predicted) and the others are context: units sold (Y) given price and promotion ($\mathbf{X}$); fraudulent or not (Y) given the transaction's features ($\mathbf{X}$); a patient's outcome (Y) given the covariates ($\mathbf{X}$). The question is not “how is the table distributed?” but “what will Y be when $\mathbf{X} = \mathbf{x}$?”

The joint distribution of the pair splits into two factors: the distribution of $\mathbf{X}$ itself, and the conditional distribution of Y given $\mathbf{X}$,

$$\underbrace{f(\mathbf{x}, y)}_{\text{joint}} = \underbrace{f_{\mathbf{X}}(\mathbf{x})}_{\text{context}} \times \underbrace{f_{Y|\mathbf{X}}(y | \mathbf{x})}_{\text{what we predict}}. \quad (3)$$

Prediction needs only the second factor. How prices came to be set is irrelevant to predicting demand on a day whose price is observed; the first factor can be left unmodeled entirely. (What demand would be if we *set* the price ourselves is a different question, a causal one, which this factorization does not answer.) This is a substantial economy: instead of a distribution on $\mathbb{R}^d$, one conditional distribution on $\mathbb{R}$, indexed by the context.

Modeling $Y | \mathbf{X}$ under explicit assumptions is *regression*, and its simplest instance is already familiar from first year: the linear model, where Y given $\mathbf{X} = \mathbf{x}$ is Gaussian with mean $\beta_0 + \beta^\top \mathbf{x}$ and constant variance σ^2 . Chapter 4 re-examines that familiar object as a statistical model, with likelihoods, tests, and honest uncertainty; the second part of the course then deliberately loosens its assumptions.

4 From distributions to functions

Sometimes even the conditional distribution resists credible specification. The dependence of Y on twenty interacting covariates is nonlinear in ways no one can write down in advance; the Gaussian

noise assumption is questionable; every added assumption is one more thing that can be false. The stance of machine learning is to retreat further: stop specifying distributions altogether, and learn a bare function $f : \mathbf{x} \mapsto \hat{y}$ directly from the table, judged by a single criterion (how well it predicts rows it has never seen). Where the statistician asks “is the model right?”, the learner asks only “does it generalize?”.

The gain is flexibility: freed from distributional form, the function f can be a decision tree, an ensemble of hundreds of trees, a boosted committee (the machinery of the last chapter of this course), or a deep network, beyond it.

5 The price: uncertainty lost, and bought back

A conditional distribution answers “what will Y be?” with a full range of possibilities and their probabilities. A function answers with one number. Figure 3 shows the retreat on a single dataset: the monthly spend of a store’s customers against their visit frequency, one point per customer. Panel (a) models the pair completely; panel (b) keeps only the conditional distributions of spend given visits, whose spread visibly grows with the visit frequency; panel (c) is the pure learner’s output, a curve: accurate on average, and silent about everything else. This matters whenever the decision depends on more than the average: what spend a new ten-visit customer will bring nine times out of ten, whether next month’s figure could plausibly be double the prediction, how much trust a single number deserves.

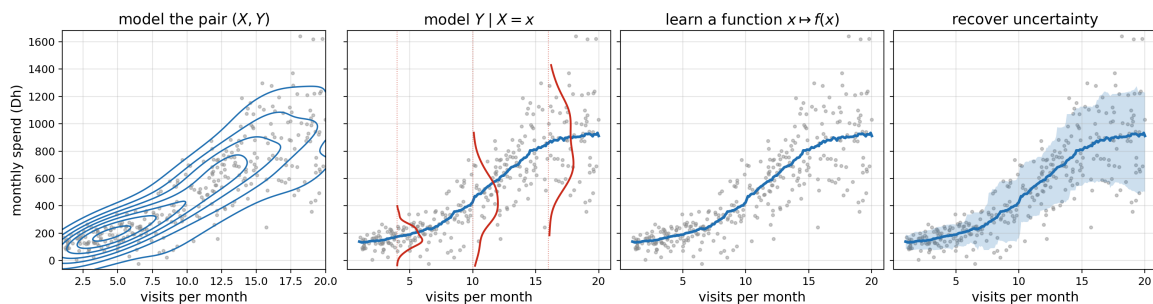


Figure 3: One dataset, four stances. (a) Model the joint distribution of the pair. (b) Model only the conditional distribution of spend given visits (red: three of its slices; note the growing spread). (c) Learn a bare function: one number per visit frequency. (d) Recover uncertainty around the function: a band containing 80% of the outcomes at each visit frequency.

Panel (d) shows what modern practice adds back: a band around the prediction calibrated so that, at each visit frequency, it contains a stated fraction of the outcomes. Two routes lead toward such bands without reinstating a full probabilistic model. *Quantile regression* learns, instead of the average, the conditional quantiles of Y directly: the edges of the band as functions of $\mathbf{x}$. *Conformal prediction* wraps any learned predictor, however opaque, in intervals whose coverage *on average over new rows* is guaranteed by the i.i.d. assumption alone; coverage at each value of $\mathbf{x}$ separately is not guaranteed. *Bayesian modeling* goes the other way: it keeps distributions everywhere, placing them on the parameters themselves so that predictions inherit a spread. Prediction uncertainty is taken up in earnest in the follow-up course, *Advanced Time Series* (ATS), through quantile regression and conformal prediction.

6 When the rows have an order: beyond i.i.d.

Everything so far treats the rows as exchangeable: shuffle the table and nothing of interest changes. For many tables this is right: customers in a database have no natural order. But rows are often *dated*, and then the order may carry the very structure we care about. Figure 4 shows

a year of the store’s daily sales twice: in calendar order, and shuffled. An i.i.d. model sees the two panels identically (it retains the histogram of values and nothing else), yet the panels differ in the way that matters most: on the left, tomorrow resembles today, the high season swells and recedes, and forecasting is possible; on the right, knowing today says nothing about tomorrow.

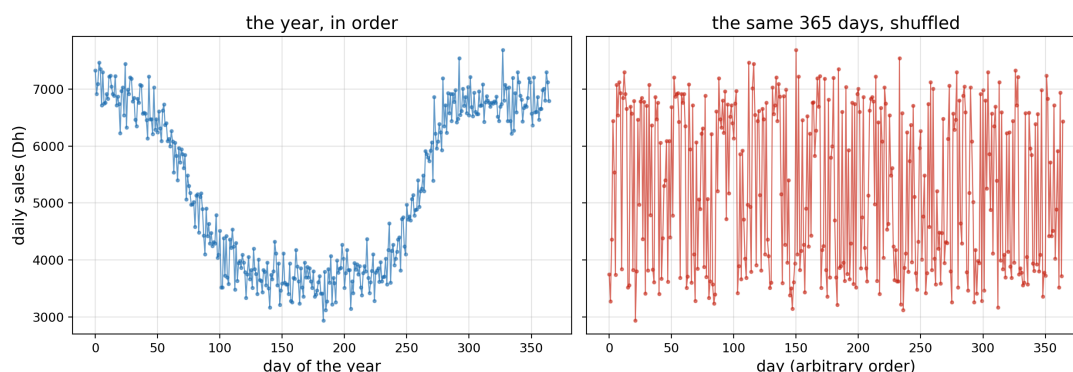


Figure 4: A year of the store’s daily sales, in calendar order (left) and shuffled (right). Both tables have exactly the same 365 values, so an i.i.d. model cannot distinguish them; the forecastable structure lives entirely in the order.

Modeling dependence *across* rows (trends, seasonality, autocorrelation) is time-series modeling, and first year devoted an entire course to it: SARIMA models, built and read in the econometrician’s spirit, where the order of the rows is the very subject of the model. This course does not rerun that material: ordered rows are the subject of the follow-up course, *Advanced Time Series* (ATS), which builds on both stances of this prologue. It recaps the classical models (exponential smoothing, SARIMA with exogenous covariates, where the forecast of a series is helped by other columns measured alongside it), then takes the learner’s side: an autoregression rewritten as a supervised problem, so that the machine-learning models of this course (regularized regression, trees, gradient boosting) become forecasters fed with the order as context (yesterday’s value, last week’s, the calendar). Evaluation and pipelines are rebuilt for dated data: baselines to beat, rolling-origin backtesting instead of random splits, and features built only from the information available when the forecast is made.

This course, then, works under the i.i.d. assumption, and for each dataset met along the way, whether exchangeability is a reasonable approximation is part of the modeling judgment, not a formality.

7 A map of the course

The course has two parts. **Part 1** carries out the programs of sections 2 and 3: chapter 1 builds the workhorse family of distributions on $\mathbb{R}^d$, the multivariate Gaussian, and confronts the cost of estimating it; chapter 2 stacks several Gaussians into mixtures to describe populations with hidden subgroups, and faces the question every enlarged family forces: how to choose among models of different complexity; chapter 3 rebuilds hypothesis testing, the art of turning probabilities computed under a model into evidence about a claim; and chapter 4, following section 3, re-reads regression as a statistical model, where assumptions yield tests and confidence statements.

Part 2 follows section 4: from models to predictors (chapter 5), where generalization replaces inference and we measure what is lost in doing so; the machine-learning pipeline (chapter 6), where predictors meet raw tables and honest evaluation; and trees and their ensembles (chapter 7), where flexibility peaks. Recovering uncertainty (section 5) and ordered rows (section 6) are left to the ATS course.

The course aims to give the judgment needed to choose, in front of a real table, between the two approaches: assumptions chosen with care, which turn scarce data into strong statements,

or abundant data, which replaces assumptions.