SAPIENZA
UNIVERSITÀ DI ROMA

Copula-Gated Pairs Trading: Unsupervised Pairs Selection and Regime-Shift Control with CUSUM

Faculty of Economics

Master's Degree in Financial Risk and Data Analysis

Candidate

Fabio Tripodi
ID number 1970235

Advisor
Prof. Galiani Stefano Simone

Co-Advisor
Dott.ssa Ranucci Maria

Academic Year 2024/2025

Thesis defended on 7 November 2025
in front of a Board of Examiners composed by:

Prof. Bianchi Sergio (chairman)
Prof. Ceci Claudia
Prof. Paiardini Paola
Prof. Galiani Stefano Simone
Prof. Vitale Domenico
Prof. Marconi Silvia
Prof. Frezza Massimiliano

Copula-Gated Pairs Trading: Unsupervised Pairs Selection and Regime-Shift Control with CUSUM
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: tripodi.1970235@studenti.uniroma1.it

Contents

Introduction

The profitability of a pairs–trading strategy depends primarily on the quality of the pair–selection step: one must identify securities whose relative valuation exhibits sufficiently stable behavior (cointegration or, more pragmatically, mean reversion of the spread). In the traditional setup, practitioners look for "similar" stocks (sector, risk exposures, fundamentals) and aim to monetize temporary deviations from the usual relationship. With broad access to data and computation, however, such opportunities are quickly arbitraged away and alpha compresses, making robust selection and filtering indispensable.

The difficulty is twofold. Restricting the search to a few industries creates a controlled environment but drastically reduces the opportunity set. Enlarging the universe increases the number of candidates but introduces considerable statistical noise and regime risk. Effective pairs trading therefore requires a balance between coverage and reliability, together with filters that suppress spurious signals and react to structural breaks.

This dissertation proposes an end-to-end pipeline that combines three elements:

1. **Unsupervised clustering for pair selection.** In a multivariate feature space (returns, volatility, co-movement, etc.), we form homogeneous groups and extract candidate pairs using two complementary criteria: an *Intra-Cluster Distance* score (`ICD`) and a *medoid*–based selection (`MEDOID`). This allows us to contrast "central" representatives with nearest-neighbor pairs inside each cluster.

2. **Copula-based trading architecture.** On a formation window we estimate the best univariate marginals for each leg and a copula $C_{\hat{\theta}}$ (Gaussian, Student–t or Archimedean) for the joint dependence of daily returns. In the test window a *rare-event gate* is applied: trades are allowed only when the observed joint outcome falls into a low-probability region under the trained copula. The spread signal itself is the z-score of the (log-price) spread with hedge ratio β estimated by OLS.

3. **Regime management via CUSUM.** We evaluate two variants of the strategy: a **CUSUM** version, where a cumulative–sum statistic can block entries and force exits under suspected regime shifts, and a **NO-CUSUM** version where the filter is disabled. This isolates the risk/return trade-off of the regime filter.

All performance is measured out of sample with non-overlapping formation and test windows. Results are reported in both gross and net terms under a conservative cost model: slippage of **2 bps per leg** and a **$0.50 fixed fee per leg** charged at both entry and exit.

Thesis structure

- **Chapter 1** reviews the foundations of pairs trading: origins and early development, the economic intuition behind market neutrality, a brief historical perspective, and applications across markets. It introduces the two–stage workflow (selection → trading) and surveys clustering families for pair discovery (connectivity/hierarchical, centroid–based, model–based, and density–based), together with evaluation criteria.
- **Chapter 2** formalizes the proposed methodologies: k-means and OPTICS for selection, cointegration testing for spread stationarity, copulas (marginals, PIT, fitting objectives) and the rare–event gate, and the CUSUM procedure for regime–shift detection.
- **Chapter 3** details the Python implementation: design choices; data, ESG variables, and preprocessing; price ingestion and annualization; balance–sheet feature selection; pair discovery (baseline k-means, final OPTICS); cointegration testing; marginals, copulas, and rare–event gating; spread construction, signal and execution rules; CUSUM gate; portfolio–level performance and risk (CUSUM vs. NO–CUSUM); and a results overview.
- **Chapter 4** concludes and outlines extensions and future research.

Chapter 1

Pairs Trading: Background and Definitions

1.1 Origins and Early Development

Pairs trading emerged at Morgan Stanley in the late 1980s within the quantitative research group led by Nunzio Tartaglia. That team—largely composed of mathematicians and physicists—seeded several of the industry's most influential quantitative franchises. Alumni went on to shape firms such as PDT Partners and D.E. Shaw, and their influence, directly and through later spinouts, extended to houses like Two Sigma. A pivotal early contributor was Gerry Bamberger, often credited with pioneering the technique, who subsequently left Morgan Stanley to work with Ed Thorp at Princeton Newport Partners.

As markets computerized and competition intensified, pure arbitrage opportunities compressed: edges became rarer and profit margins thinner. Around the early 2000s the once universally lauded pairs-trading playbook entered what practitioners dubbed an "ice age": returns declined relative to the strategy's early years and many managers curtailed risk. Roughly a decade later, a renewed wave of research and technological progress revitalized the area. A large body of empirical and methodological work accumulated, broadening the scope of the approach and improving robustness across instruments, data frequencies, and implementation details.

What is Pairs Trading?

Pairs trading belongs to the broader family of *statistical arbitrage*. While the terms are sometimes used interchangeably, pairs trading is a subset: all pairs trading is statistical arbitrage, but not all statistical arbitrage is pairs trading. In practice, statistical arbitrage encompasses (i) factor or cross-sectional styles and (ii) mean-reverting portfolios, of which pairs trading is the canonical example.

In its simplest form, a pairs strategy trades two assets; in general it can be extended to an n-asset mean-reverting portfolio.
The core object is a spread

$$S_t \;=\; P_{1,t} - \beta\, P_{2,t},$$

where $P_{i,t}$ denotes the price (or log-price) of asset i at time t and β is a hedge ratio calibrated to capture the equilibrium co-movement between the two assets.
For illustration, one may construct S_t by going long the first asset and short, say, 60% of the second; if S_t is mean-reverting, it can be traded directly without forecasting the individual price paths.

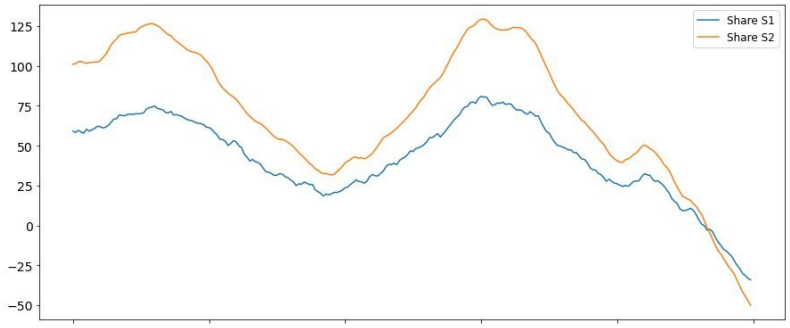


Figure 1.1. Simulated cointegrated series and the cointegration error, $\beta = -0.6$

Figure 1.2. Example Spread for CACC vs NXST

1.1.1 Economic Intuition and Market Neutrality

The intuitive foundation echoes a basic investment principle—buy the relatively cheap, sell the relatively expensive—but replaces the (hard-to-observe) notion of intrinsic value with *relative value*. If two securities share similar characteristics and risk exposures, their prices should co-move. Deviations from this relationship are summarized by the spread S_t. When the spread widens away from its long-run level, either one security has become overpriced, the other underpriced, or both; taking the opposite positions aims to profit from subsequent convergence toward equilibrium. A key advantage is near market-neutrality: by adjusting β, the spread can be engineered to have a negligible exposure (beta) to the market, thus focusing risk on the specific relative-value relationship rather than broad market moves.

Implementations typically differ along two axes: (i) the method used to detect co-moving candidates (e.g., distance-based selection, correlation screens, or cointegration tests), and (ii) the trading rules that govern entries, exits, and risk management (e.g., z-score thresholds, stop-outs, and dynamic re-hedging). The literature offers many variants, ranging from very simple heuristics to fully fledged, high-frequency systems.

1.1.2 Historical Perspective

The strategy's industrial influence is difficult to overstate. Beyond spawning independent powerhouses (PDT, D.E. Shaw, among others), pairs trading helped shape surveillance practices: anecdotal accounts describe regulators such as the SEC adopting algorithmic tools to flag unusual cross-sectional price patterns. As the know-how diffused—accelerated by the advent of affordable computing—the number of practitioners grew sharply. Performance pressures in the early 2000s

ushered in the aforementioned "ice age" (a term popularized by Andrew Pole), but subsequent advances in data, computation, and methodology reignited interest. Today, pairs trading is best viewed not as a single recipe but as a versatile framework with multiple estimation and execution choices, applicable across asset classes and horizons.

1.1.3 Pairs Trading Across Markets

Having clarified the "what" and "why" of pairs trading, the natural next question is "where?"—in which markets should we implement such strategies. Because pair selection is the first design step, choosing the trading universe is consequential.

Although pairs trading has its roots in equities, which remain the default choice for many practitioners due to the sheer number of possible pair combinations, the approach extends well beyond stocks. Commodities, foreign exchange, and even crypto-assets all host variants of statistical arbitrage backed by an expanding empirical literature—some strands dating back to the early 1990s for commodity futures, others more recent for digital assets. The unifying requirements—*price co-movement* and *short-horizon pricing inefficiencies*—are not asset-class specific. Whenever the *Law of One Price* is temporarily violated, relative-value strategies such as pairs trading can flourish. That said, each market comes with microstructural nuances that shape how we build, test, and deploy strategies.

Equities and ETFs

Equity and ETF universes are often the first port of call because they offer the richest combinatorial set of candidate pairs. Cross-country evidence (e.g., the study of Jacobs & Weber across 34 international markets) reports stronger pairs-trading profits in emerging markets—plausibly reflecting higher frictions and mispricings—or in markets where the number of viable pairs is large.

These benefits come with constraints tied to *short selling*, which is integral to market-neutral relative-value trades:

1. **Collateral and financing.** Short sales typically require margin collateral and incur borrow and financing costs.

2. **Borrow availability.** The less-followed the stock or ETF, the harder (and costlier) it can be to secure a borrow.

3. **Regulatory limits.** Jurisdictions may impose rules that restrict shorts (e.g., the historical U.S. uptick rule or episodic short-sale bans in some markets).

These frictions should be reflected in backtests via realistic borrow fees, locate failures, and execution constraints.

Futures

Futures—particularly commodity futures—are long-standing venues for pairs trading, with co-movement phenomena documented decades ago. A practical advantage is that going short is symmetric and does not require locating shares. However, implementation details matter:

- **Backtesting data.** P&L simulations typically use back-adjusted or continuous series to account for contract rolls and avoid artificial jumps.

- **Sizing and capital.** Position sizing and margin planning must use *raw* tradable prices and contract specifications (multiplier, tick size).
- **Hedge ratio.** For some specifications, contract size and implied volatility should enter the hedge-ratio estimation to align risk on both legs.

Foreign Exchange (FX)

FX is a less common but viable setting for statistical arbitrage. To construct the desired relative-value exposure when a direct cross is illiquid or unavailable, practitioners form *synthetic pairs* using a liquid intermediary (often USD). For example, a proxy for AUD/CAD can be created by combining positions in AUD/USD and USD/CAD. Shorting in FX is operationally straightforward—selling one currency naturally implies buying another—though liquidity, spreads, and rollover (swap) costs must be accounted for.

Crypto-assets

Despite being the youngest market with comparatively fewer peer-reviewed studies, crypto has appealing properties for stat arb. Some research (e.g., Al–Yahyaee et al.) reports higher inefficiency in Bitcoin relative to equities or FX, suggesting scope for mean-reversion trades. Constraints remain material: only a subset of the most-traded coins can be shorted; margin and shorting facilities vary across venues; and effective transaction costs (commissions, funding rates, and bid–ask spreads) can exceed those in mature markets.

1.2 Two-Stage Workflow

In practice, a pairs-trading program has two distinct phases:

1. **Formation.** Identify securities that share a persistent, economically plausible long-run relation and design a spread that is *mean-reverting* yet sufficiently volatile to produce opportunities. In other words, we seek a process whose variance is large enough to generate signals, while its pull toward equilibrium is strong enough to close trades reliably.
2. **Trading.** Specify entry and exit rules so that positions are initiated when the spread diverges from equilibrium and closed when it reverts. This layer also includes re-hedging, risk limits, and transaction-cost controls.

Over the years, research has developed several families of methods for these two phases. The main streams relevant to pairs trading are outlined below.

Distance-Based Approach

Popularized by *Gatev et al.* (2006), this is the most widely cited template in the literature. Its appeal is simplicity and transparency, which makes it suitable for large-scale empirical studies. During formation, assets are screened using distance-style measures to capture co-movement—beyond simple Pearson correlation, one encounters distance correlation, angular distance, and related metrics. During the trading phase, nonparametric threshold rules (e.g., z-score bands) generate the entry/exit signals.

Cointegration-Based Approach

Another influential line, detailed by *Vidyamurthy* (2004), selects pairs whose prices exhibit an econometrically validated equilibrium relation. Formation relies on cointegration testing (e.g., Engle–Granger, Johansen) to retain pairs with a stationary spread. Trading rules are typically simple and often mirror the threshold logic used in the distance tradition, as in *Gatev et al.* (2006).

Time-Series Model-Based Approach

To refine trading decisions, one can model the spread explicitly as a mean-reverting process (distinct from, though compatible with, cointegration). The goal in formation is to construct a portfolio that follows a well-behaved mean-reversion dynamic; the trading layer then exploits that model to set data-driven thresholds. A common specification is the Ornstein–Uhlenbeck (OU) process,

$$\mathrm{d}S_t = \kappa\,(\mu - S_t)\,\mathrm{d}t + \sigma\,\mathrm{d}W_t,$$

whose parameters inform half-life, expected reversion, and risk, and thus the sizing of entry/exit bands.

Stochastic-Control Approach

Here, stochastic-process techniques deliver *optimal* policies for mean-reversion trades without forecasting the next-period spread explicitly and, in some settings, without a separate formation window. The spread dynamics (often OU-type) are combined with dynamic programming/HJB arguments to obtain closed-form or numerically computed trading rules and inventory policies. This is an advanced framework that directly optimizes the objective under costs and constraints.

Other Advanced Approaches

Although grouped under "other," the methods below are among the most sophisticated and extend pairs trading beyond two assets.

Copula Models

Copulas allow one to study the dependence structure between the legs (or multiple legs) of a trade beyond linear correlation. The typical workflow has two steps: (i) map each marginal series to its quantile scale, and (ii) fit a copula (Gaussian, t, Archimedean, etc.) to those transformed data. Trading rules are then based on conditional probabilities implied by the fitted copula. A key advantage is the ability to handle *multivariate* spreads rather than a single pair.

Principal Components (PCA)

Following *Avellaneda & Lee* (2010), PCA can be used to synthesize a spread that appears mean-reverting, which is then modeled via an OU process to generate signals. The methodology hinges on the choice of PCA variant—standard, asymptotic, or links to multivariate cointegration—and on how the factors are interpreted and constrained. At the time of writing, many practitioners view this factor-based construction as a cutting edge in mean-reversion trading.

Machine Learning

Machine learning augments several steps of the pipeline. Rather than forecasting the spread outright (a difficult task), many implementations use ML for *pair selection*: ranking candidate pairs by the likelihood of stable mean reversion or by expected risk-adjusted return, using features drawn from the approaches above (distance measures, cointegration statistics, liquidity and microstructure variables, fundamentals, etc.).

Proposed Pair-Selection Methodology

The profitability of any pairs-trading program is driven first and foremost by the quality of the pairs we choose. Yet the modern data landscape and low-latency access mean that many practitioners can quickly detect and exploit price dislocations; by the time a signal is widely observed, the edge is often competed away. Superior outcomes therefore hinge on uncovering pairs that are *not* already under the constant surveillance of the crowd—opportunities where relative mispricings can persist long enough to be monetized. Identifying such candidates, however, is decidedly nontrivial.

Most prior work has concentrated on classic statistical techniques and well-established diagnostics, seeking incremental improvements to the critical steps of the strategy—how pairs are discovered, how spreads are constructed, and how trading rules are triggered.

1.2.1 Cluster Analysis for Pair Selection

Cluster analysis (or *clustering*) groups a collection of objects into subsets (*clusters*) so that items in the same group are, according to a chosen notion of similarity, more alike than items in different groups. It is a core tool for exploratory data analysis and appears widely across pattern recognition, image and information retrieval, bioinformatics, data compression, computer graphics, and machine learning. In our context, clustering is a natural way to organize the trading universe into coherent buckets of co-moving assets that can later yield candidate pairs.

Clustering is not a single algorithm but a family of tasks and methods. Different procedures embody different ideas of what a "cluster" should be and how to find it efficiently. Depending on the perspective, clusters may be defined as collections of points with small within-group distances, dense regions of the feature space, contiguous intervals, or samples drawn from a particular statistical distribution. Because multiple objectives compete (compactness, separation, robustness, computational efficiency), clustering is often best viewed as a multi-objective optimization problem. The most suitable algorithm and its hyperparameters (e.g., distance or similarity function, density thresholds, expected number of clusters) depend on the data and on the intended use of the results. In practice, clustering is an iterative, interactive workflow: data preprocessing and model settings are refined until the output exhibits the desired properties.

What constitutes a cluster?

There is no universally precise definition of a cluster, which explains the variety of models and algorithms. Understanding these models helps explain why methods may disagree on the same data:

- **Connectivity models.** Hierarchical methods build clusters by linking nearby points based on distance connectivity.

- **Centroid models.** Algorithms such as k-means represent each cluster by a prototype (typically the mean) and assign points to the nearest centroid.
- **Distribution models.** Mixture-model approaches treat clusters as samples from parametric distributions (e.g., multivariate normal), typically estimated via expectation–maximization.
- **Density models.** Methods like DBSCAN or OPTICS define clusters as contiguous high-density regions separated by low-density gaps.
- **Subspace models.** Biclustering/co-clustering identifies groups together with the relevant features (attributes) for each group, acknowledging that different clusters may live in different subspaces.
- **Group/partition models.** Some procedures simply produce a grouping without an explicit generative model.
- **Graph-based models.** On graphs, cliques (or relaxed versions such as quasi-cliques) serve as prototypical clusters when edges encode similarity.
- **Signed-graph models.** With positive/negative edges, balance-theoretic conditions on cycles lead to partitions with predominantly positive ties within clusters.
- **Neural models.** Unsupervised neural maps (e.g., self-organizing maps) and related architectures often approximate one or more of the models above; when they implement PCA/ICA-like transformations, they connect to subspace models.

A *clustering* is a collection of clusters, usually covering all objects and sometimes endowed with relationships among clusters (e.g., a nested hierarchy).

Hard vs. soft assignments

Two broad assignment schemes are common:

- **Hard clustering:** each object belongs to exactly one cluster.
- **Soft (fuzzy) clustering:** each object belongs to clusters with degrees of membership. Let $\mu_{ik} \in [0, 1]$ denote the membership of object i in cluster k, typically with $\sum_k \mu_{ik} = 1$ for each i.

Algorithms and practical choices

Algorithms are often categorized by the underlying model summarized above. There is no universally "correct" procedure: a celebrated axiomatic result shows that no clustering method can simultaneously satisfy three desirable properties—scale invariance (insensitivity to uniform rescaling of distances), richness (ability to realize any partition), and consistency (partitions align with changes in distances that strengthen within-cluster ties and weaken between-cluster ties). Consequently, method selection should be guided by the geometry and noise structure of the data and by the downstream task.

In practice:

- **Preprocessing matters.** Standardization/whitening and feature engineering shape distance computations $d(x_i, x_j)$ and hence the geometry seen by the algorithm.
- **Hyperparameters are consequential.** Examples include the number of clusters k (for centroid models), the neighborhood radius and minimum points (for density models), linkage type (for hierarchical methods), and initialization strategies.
- **Model–data fit.** Algorithms designed for roughly convex or elliptical groups (e.g., k-means) will fail on non-convex clusters; density or graph-based methods are often preferable in those cases.

Relevance to pair selection.

For pairs trading, we will construct feature vectors that summarize co-movement and tradability (e.g., correlations at multiple horizons, cointegration diagnostics, liquidity and microstructure measures), cluster securities to identify homogeneous groups, and then restrict pair selection to within-cluster candidates. This reduces spurious matches across unrelated names and focuses the search where stable relative-value relationships are most plausible.

1.2.2 Connectivity-Based (Hierarchical) Clustering

Connectivity-based methods—commonly called *hierarchical clustering*—start from the premise that nearby objects are more closely related than distant ones. The algorithm connects items according to a distance function $d(\cdot, \cdot)$, building groups whose internal linkages are short relative to cross-group links. Rather than returning a single partition, these methods produce a *hierarchy* of merges that can be visualized with a dendrogram: the y-axis records the dissimilarity (linkage height) at which clusters merge, while the x-axis orders the objects so that branches do not cross.

Two design choices define a hierarchical procedure:

$$D_{\text{single}}(A, B) = \min_{i \in A,\, j \in B} d(i, j),$$

$$D_{\text{complete}}(A, B) = \max_{i \in A,\, j \in B} d(i, j),$$

$$D_{\text{average}}(A, B) = \frac{1}{|A||B|} \sum_{i \in A} \sum_{j \in B} d(i, j).$$

corresponding, respectively, to single linkage, complete linkage, and average linkage (UPGMA/WPGMA).
The algorithm can be *agglomerative* (bottom-up: start from singletons and merge) or *divisive* (top-down: start from the full set and split).

Hierarchical clustering yields a family of plausible partitions; the analyst selects a cut level to obtain a concrete clustering. Classic caveats include sensitivity to outliers and the *chaining* phenomenon under single linkage (long, thin clusters created by bridging points). Computationally, generic agglomerative implementations run in $\mathcal{O}(n^3)$ time and $\mathcal{O}(2^{n-1})$ memory for n objects. Divisive strategies are often at least as demanding—frequently $\mathcal{O}(n^3)$ or worse depending on the splitting heuristic—and

can be impractical at large n. For special cases, optimal $\mathcal{O}(n^2)$ time algorithms exist (e.g., SLINK for single linkage and CLINK for complete linkage).

1.2.3 Centroid-Based Clustering

In centroid-based methods, each cluster is represented by a prototype vector (a *centroid*) that need not coincide with any observed point. Fixing the number of clusters to k, k-means formulates clustering as

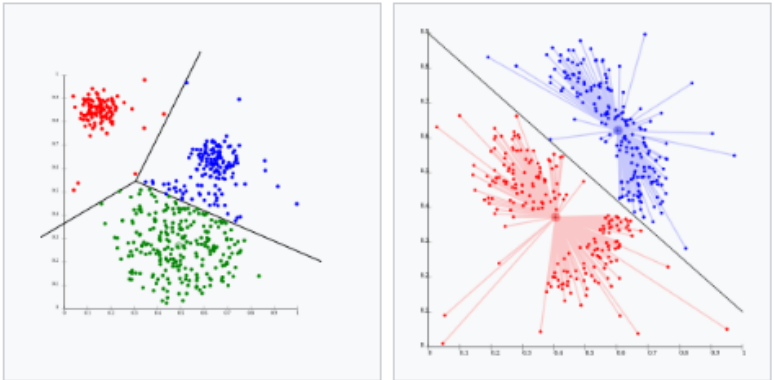
$$\min_{\{C_1,\dots,C_k\},\,\{c_1,\dots,c_k\}} \sum_{r=1}^{k} \sum_{x_i \in C_r} \|x_i - c_r\|^2,$$

i.e., choose centroids $\{c_r\}$ and assignments $\{C_r\}$ to minimize the within-cluster sum of squares. This problem is NP-hard in general; in practice one uses approximate heuristics, most famously Lloyd's algorithm (often colloquially called "the k-means algorithm"), which alternates between assigning each point to its nearest centroid and recomputing centroids as means. Because it converges only to a *local* optimum, multiple random restarts are standard.

Common variants include k-medoids (centroids restricted to data points), k-medians (using ℓ_1 distances), k-means++ (smarter seeding), and fuzzy c-means (soft memberships). Typical limitations are the need to pre-specify k and a tendency to favor clusters of comparable size and roughly convex/elliptical shape, since assignments are to the nearest centroid and boundaries are induced by Voronoi cells.

A canonical k-means workflow

1 **Initialization:** Choose k initial centroids (often with k-means++ seeding).
2 **Assignment.** For observations $x_1, \dots, x_n$, assign each x_i to the nearest centroid in squared Euclidean distance.
3 **Update:** Recompute each centroid as the mean of its assigned points.
4 **Repeat:** Iterate assignment/update until convergence (e.g., no change in assignments or negligible objective improvement).



k-means separates data into Voronoi cells, which assumes equal-sized clusters (not adequate here).

k-means cannot represent density-based clusters.

Figure 1.3. k-means clustering examples

Facility-Location View of Centroid Methods

Centroid-based clustering methods such as k-means and k-medoids can be interpreted as special cases of the uncapacitated metric *facility location* problem, a classic topic in operations research and computational geometry. In the basic formulation, one chooses locations for "facilities" (warehouses) to serve a set of "clients" at minimal cost. By analogy, centroids act as facilities and data points as clients; assigning points to centroids corresponds to routing clients to their nearest warehouse. This perspective allows one to leverage algorithmic tools, approximation guarantees, and heuristics developed for facility location when designing or analyzing centroid-based clustering procedures.

1.2.4 Model-Based Clustering

Model-based clustering adopts a statistical generative view: the data arise from a mixture of probability distributions. A common specification uses Gaussian components, leading to the Gaussian mixture model (GMM) with density

$$f(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k), \qquad \pi_k \geq 0, \ \sum_k \pi_k = 1.$$

Parameters are typically estimated via the expectation–maximization (EM) algorithm; EM converges to a local optimum, so multiple initializations are customary. For a *hard* partition, each observation is assigned to the component with the largest posterior responsibility, while *soft* clusterings retain the full responsibility vector.

The framework offers principled answers to questions such as the number of clusters K, model choice, and outlier handling, but it also risks overfitting unless model complexity is constrained. Parsimonious covariance parameterizations—e.g., spherical, diagonal, tied, or eigenvalue-decomposed structures—provide a practical balance between flexibility and robustness. A key limitation is *model misspecification*: many real data sets do not adhere to a concise distributional form (e.g., assuming Gaussianity can be restrictive), in which case fit and interpretability may degrade.

1.2.5 Density-Based Clustering

Density-based methods define clusters as regions where observations concentrate more densely than in the remainder of the space; points in sparse areas that separate dense regions are treated as noise or border points.

DBSCAN

DBSCAN uses two main parameters: a neighborhood radius ε and a minimum number of neighbors minPts. Points with at least minPts neighbors within distance ε are *core* points; clusters are maximal sets of points that are density-reachable from one another via chains of core points. This yields clusters of arbitrary shape and an algorithm with relatively low complexity (a linear number of range queries under suitable indexing). Results are stable across runs for core and noise points; only border-point assignments may vary, so repeated runs are usually unnecessary.

OPTICS

OPTICS generalizes DBSCAN by avoiding a single global choice of ε and instead producing an ordering and a *reachability* profile from which cluster structure at multiple density levels can be extracted. The output is often viewed as a hierarchical analogue to linkage clustering, tuned for density notions.

DeLi-Clu

Density-Link-Clustering combines ideas from single-linkage and OPTICS. It removes the need to fix ε explicitly and can deliver performance improvements by exploiting spatial indexes such as R-trees.

Limitations

DBSCAN/OPTICS expect a discernible drop in density to delineate boundaries. With overlapping continuous distributions (e.g., mixtures of Gaussians) where density decays smoothly, borders can become arbitrary; model-based approaches like EM for GMMs typically handle such cases more naturally.

Mean-Shift

Mean-shift performs mode seeking via kernel density estimation (KDE): each point iteratively moves toward the local maximum of the estimated density, and converged modes act as cluster representatives. Like DBSCAN, mean-shift can discover clusters of arbitrary shape, but it is usually slower due to repeated KDE evaluations and iterative updates. Choice of bandwidth is crucial; in higher dimensions, the KDE can become rough and lead to over-fragmentation, particularly in low-density tails.

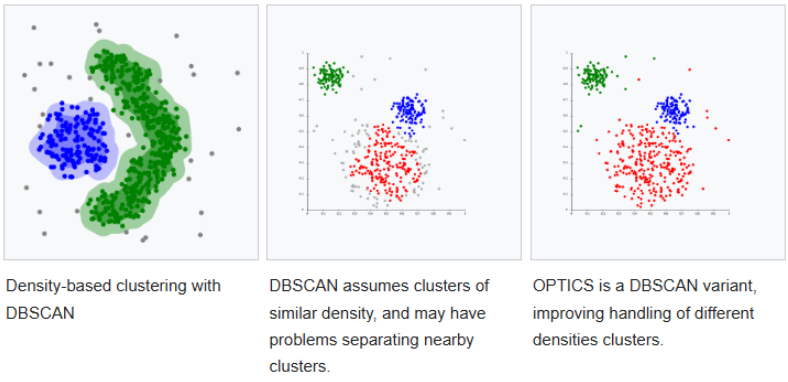


Figure 1.4. Density-based clustering examples

Grid-Based Clustering

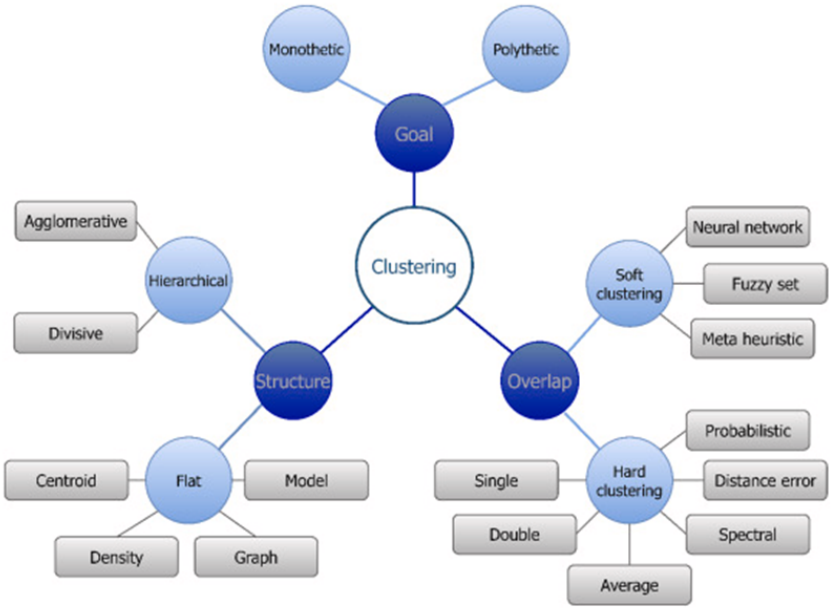
Grid-based methods partition a d-dimensional space into a finite lattice of axis-aligned *cells* (a grid) and operate on cells rather than individual observations. Because computations scale with the number of occupied cells, these techniques are typically fast and have low computational burden. Representative algorithms include *STING* and *CLIQUE*.

Basic procedure

Let $\rho(c)$ denote the density of a cell c (e.g., the count of points in c)
and let τ be a user-specified density threshold.
Partition the feature space into a finite number of cells.
while there exists an *unvisited* cell c
    Compute the density $\rho(c)$.
    if $\rho(c) \geq \tau$
        Start a new cluster with c.
        for each neighbor c' of c (according to the chosen grid adjacency)
            Compute $\rho(c')$;
            if $\rho(c') \geq \tau$
                Add c' to the cluster and continue expanding from c'.

        Mark c as visited and continue.

    Repeat until all cells have been traversed.
    Stop.



1.2.6 Evaluation and Assessment

Validating clustering results is nearly as challenging as producing them. Common strategies fall into four families: (i) *internal* validation, which maps a clustering to a single quality score; (ii) *external* validation, which compares a clustering to a known reference partition; (iii) *manual* inspection by a domain expert; and (iv) *indirect* or task-based validation, which measures usefulness in the downstream application.

Internal validation

Internal indices summarize separation between clusters and compactness within clusters using only the data and the assignments. A well-known example is the Silhouette coefficient for point i,

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}},$$

where a_i is the average dissimilarity of i to its own cluster and b_i is the smallest average dissimilarity of i to any other cluster. While such indices are convenient, they effectively define their *own* optimization problems; evaluating by an internal index often measures agreement with that index rather than practical usefulness.

External validation

External measures compare the output to a "ground-truth" labeling when one is available, using agreement scores or information-theoretic criteria. In many realistic settings no such labels exist; even when they do, they represent just one plausible partition and may not coincide with the structure relevant for the task at hand.

Manual evaluation

Human assessment incorporates domain knowledge but is inherently subjective. Expert review remains essential when automated criteria contradict practical considerations.

Indirect (task-based) evaluation

Here the clustering is judged by its contribution to the ultimate goal—for example, whether within-cluster pair selection improves a pairs-trading strategy's risk-adjusted performance. This approach aligns evaluation with end-use, though it entangles validation with modeling and execution choices.

In short, no single criterion can definitively certify clustering quality. Quantitative indices help flag poor solutions, but expert judgment and task-level performance are crucial to determine whether a clustering is genuinely useful.

Metric	Equation	Description
Sum of the squared error	$\mathrm{SSE} = \sum_{c=1}^{C} \sum_{i \in c} \sum_{y} \sum_{m=1}^{M} \| x_{i,y,m} - \bar{x}_{c,y,m} \|^2$	C is the total number of clusters, and subscripts m and y represent the technology metric and year.
Davies–Bouldin index	$\mathrm{DB} = \frac{1}{C} \sum_{c=1}^{C} \max_{c' \neq c} \left\{ \frac{S_c + S_{c'}}{\|\bar{x}_c - \bar{x}_{c'}\|^2} \right\}$	S_c is the average distance between the data points of cluster c and its centroid.
Calinski–Harabasz index	$\mathrm{CH} = \frac{\sum_{c=1}^{C} \frac{n_c \, \|\bar{x}_c - \bar{x}_g\|^2}{(C-1)}}{\sum_{c=1}^{C} \sum_{i \in c} \frac{\|x_i - \bar{x}_c\|^2}{(N-C)}}$	$\bar{x}_c$ is the centroid of cluster c, $\bar{x}_g$ is the global centroid, and n_c is the number of objects in cluster c.
Dunn index	$\mathrm{DN} = \frac{\min_{i \in c,\, j \in c'} \|x_i - x_j\|}{\max_{i,j \in c} \|x_i - x_j\|}$	c' is a cluster other than cluster c.
Silhouette index	$\mathrm{SL} = \frac{1}{N} \sum_{c=1}^{C} \sum_{i \in c} \frac{b_i - a_i}{\max\{b_i, a_i\}}$	$\max\{b_i, a_i\}$ is the greater one of a data point's cohesion and separation values.

Chapter 2

Proposed Pairs Trading Methodologies

2.1 k-Means

k-means is a vector-quantization method (originating in signal processing) that partitions n observations in $\mathbb{R}^d$ into k clusters so that each point is assigned to the nearest cluster *mean* (centroid). The induced partition of the space corresponds to Voronoi cells around the centroids. Unlike the Weber (geometric-median) problem that minimizes *Euclidean* distances, k-means minimizes *squared* Euclidean deviations—means optimize squared error, whereas geometric medians optimize Euclidean distance. For distance objectives that are not squared, alternatives such as k-medians or k-medoids are preferable.

History

The term *k-means* was introduced by James MacQueen in 1967, with antecedents tracing back to work by Hugo Steinhaus in the 1950s. The standard iterative procedure is commonly attributed to Stuart Lloyd (Bell Labs, 1957) in the context of pulse–code modulation; it circulated internally for years and appeared in print later. Edward W. Forgy published an essentially equivalent scheme in the 1960s, hence the frequent label *Lloyd–Forgy algorithm.*

Objective and equivalent forms

Let $\mathcal{S} = \{S_1, \ldots, S_k\}$ be a partition of the data, and let
$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$
denote the mean (centroid) of cluster S_i. The k-means objective minimizes the within-cluster sum of squares (WCSS):
$\arg\min_{\mathcal{S}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg\min_{\mathcal{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var}(S_i)$.
Using the identity
$|S_i| \sum_{x \in S_i} \|x - \mu_i\|^2 = \frac{1}{2} \sum_{x,y \in S_i} \|x - y\|^2$,
the problem is equivalently written as minimizing the average pairwise squared distances within clusters:
$\arg\min_{\mathcal{S}} \sum_{i=1}^{k} \frac{1}{|S_i|} \sum_{x,y \in S_i} \|x - y\|^2$.

Because the total variance of the data is constant, minimizing WCSS is the same as maximizing the between-cluster sum of squares (BCSS), a deterministic consequence of the law of total variance.

Computation and behavior

Finding the global optimum is NP-hard, but fast heuristics converge to a local optimum. The standard procedure—often called Lloyd's algorithm—alternates between (i) assigning each point to its nearest centroid and (ii) updating centroids as the mean of assigned points. Multiple random restarts (or k-means++ seeding) are commonly used to improve solutions. Both k-means and Gaussian mixture modeling proceed by iterative refinement and maintain explicit cluster centers; however, k-means favors clusters of comparable spatial extent (spherical, equal-variance), whereas Gaussian mixtures allow clusters with different shapes via distinct covariance matrices.

Nearest-centroid classification

Although k-means is unsupervised, the centroids it returns can be used for classification: assigning a new point to the closest centroid (1-nearest neighbor on the set of centroids) yields the *nearest-centroid* or Rocchio classifier, which maps new data into the existing k-means clusters.

Standard Algorithm (naïve k-means)

The method proceeds by alternating assignment and update steps, starting from an initial set of k centroids
$\mathrm{m}_1^{(1)}, \ldots, m_k^{(1)} \in \mathbb{R}^d$.
At iteration $t = 1, 2, \ldots$:
1) Assignment step
Each observation x_p is assigned to the cluster with the nearest centroid in *squared Euclidean* distance, inducing Voronoi cells:
$\mathrm{S}_i^{(t)} = \{\, x_p : \|x_p - m_i^{(t)}\|^2 \le \|x_p - m_j^{(t)}\|^2 \ \ \forall j,\ 1 \le j \le k \,\}$,
so that every x_p belongs to exactly one set $S_i^{(t)}$ (ties broken arbitrarily).
2) Update step
Centroids are recomputed as means of the currently assigned points:
$\mathrm{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j, \qquad i = 1, \ldots, k.$
Objective and convergence
k-means minimizes the within–cluster sum of squares (WCSS)
$\Phi(\mathcal{S}, m) = \sum_{i=1}^{k} \sum_{x \in S_i} \|x - m_i\|^2$.

Each assignment step (with fixed centroids) and each update step (with fixed assignments) does not increase Φ; hence the sequence of objective values is nonincreasing and bounded below by 0, so the algorithm converges to a fixed point (a local optimum or saddle). Convergence is typically fast but not guaranteed to reach the global optimum; the final solution depends on initialization.

Remarks

The monotonicity argument relies on *squared Euclidean* geometry: replacing it with a different dissimilarity can break convergence. Variants tailored to other distances include spherical k-means (cosine similarity) and k-medoids (usable with general metrics). In practice, multiple random restarts or k-means++ seeding are used to improve the attained local optimum.

Initialization Methods

Two classic initialization schemes are widely used:

- **Forgy initialization.** Choose k observations uniformly at random from the data and use them as the initial centroids. This tends to place seeds spread out across the space.
- **Random partition.** Assign each observation to one of k clusters uniformly at random, then compute the initial centroid of each cluster as the mean of its assigned points. This often yields initial centroids near the global center of mass.

For variants such as k-harmonic means and fuzzy k-means, random-partition seeding is often advantageous, whereas standard k-means and expectation–maximization (EM) for Gaussian mixtures typically benefit from Forgy-style seeding (or improved schemes like k-means++).

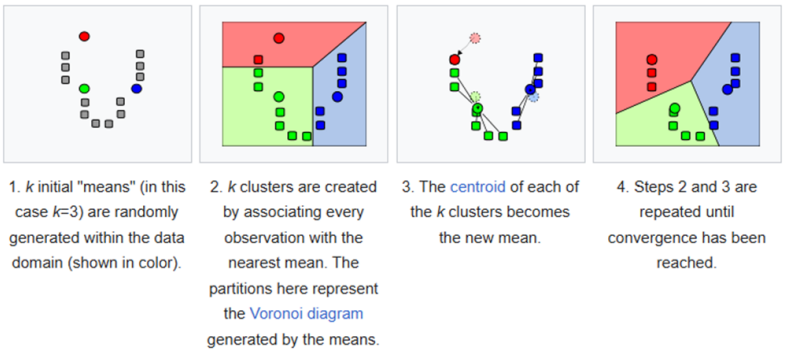


Figure 2.1. Demonstration of the standard algorithm

Convergence and complexity

k-means is not guaranteed to find the global minimum of the WCSS objective; the final solution can depend strongly on the initial centroids. Because the routine is fast in practice, it is common to perform multiple restarts with different seeds and keep the best outcome. Worst-case inputs exist (even in two dimensions) for which the number of iterations grows exponentially—on the order of $2^{\Omega(n)}$—but such pathological cases are rarely encountered; smoothed analyses show polynomial-time behavior on perturbed data.

EM viewpoint

The assignment step plays the role of an *E-step* (compute hard responsibilities by nearest-centroid assignment), and the centroid update is a *M-step* (maximize the objective with respect to centroids). In this sense, k-means is a special case of generalized EM with hard assignments.

Choosing the Number of Clusters

1 **Elbow method**
2 This heuristic inspects the within-cluster sum of squares (also called *inertia*) as a function of k and selects the point where marginal improvements flatten:
3 Inertia(k) $= \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$,
4 where μ_i is the centroid of cluster C_i. The "elbow" is the value of k after which additional clusters yield diminishing returns.

Choosing the Number of Clusters

1 **Silhouette score**
2 The silhouette quantifies, for each observation x, how well it matches its assigned cluster relative to others:
3 s(x) = b(x)-a(x)$\frac{}{\max\{a(x), b(x)\},}$
4 where $a(x)$ is the average distance from x to points in its own cluster and $b(x)$ is the smallest average distance from x to points in any other cluster. Scores range from -1 to 1, with higher values indicating better-defined clusters.

Challenges with k-Means

- **Selecting k.** Determining an appropriate number of clusters is inherently problem-dependent.
- **Sensitivity to initialization.** Different random seeds can yield different local optima.
- **Non-spherical structure.** Performance degrades when clusters vary in shape, size, or density.
- **Outliers.** Extreme points can pull centroids and distort assignments.

2.2 OPTICS

OPTICS is a density-based clustering method for spatial data. Its core idea is close to DBSCAN, but it remedies a crucial limitation: detecting meaningful clusters when densities vary across regions of the data. The algorithm produces a *linear ordering* of the points so that spatially close points appear next to one another in the order, and it records for each point a special distance that reflects the minimum

density level at which two neighboring points would belong to the same cluster. This information is visualized by a dendrogram-like *reachability plot*.

Parameters and neighborhoods

1 OPTICS uses two inputs: a radius $\varepsilon > 0$ and a minimum-points threshold MinPts $\in \mathbb{N}$.
2 For a point p, the ε-neighborhood is
3 $\mathrm{N}_\varepsilon(p) = \{\, o : \ \mathrm{dist}(p, o) \le \varepsilon \,\}$.
4 A point p is a *core* point if $|N_\varepsilon(p)| \ge \mathrm{MinPts}$ (including p itself).

Core distance

1 The core-distance of p is the distance to its MinPts-th nearest neighbor within $N_\varepsilon(p)$; if p is not a core point, it is undefined:

2 $$\text{core-dist}_{\varepsilon,\mathrm{MinPts}}(p) = \begin{cases} \text{undefined}, & |N_\varepsilon(p)| < \mathrm{MinPts}, \\ \text{MinPts-th smallest distance in } N_\varepsilon(p), & \text{otherwise.} \end{cases}$$

Reachability distance

1 Given points p and o, the reachability-distance of o from p is

2 $$\text{reach-dist}_{\varepsilon,\mathrm{MinPts}}(o; p) = \begin{cases} \text{undefined}, & |N_\varepsilon(p)| < \mathrm{MinPts}, \\ \max\Big\{\text{core-dist}_{\varepsilon,\mathrm{MinPts}}(p),\ \mathrm{dist}(p, o)\Big\}, & \text{otherwise.} \end{cases}$$

If p and o are nearest neighbors, this equals the smallest radius $\varepsilon' \le \varepsilon$ sufficient to treat p and o as belonging to the same density-based cluster. Both the core-distance and the reachability-distance are undefined when no sufficiently dense neighborhood (with respect to ε) exists. Setting ε extremely large avoids undefined values, but then every neighborhood query can degenerate to scanning the full data set, which is undesirable computationally. Practically, ε serves as a cutoff that limits neighborhoods to a meaningful scale and speeds up processing; conceptually, OPTICS abstracts away from committing to a single global density threshold by working with the maximum admissible value.

Reachability plot and cluster extraction

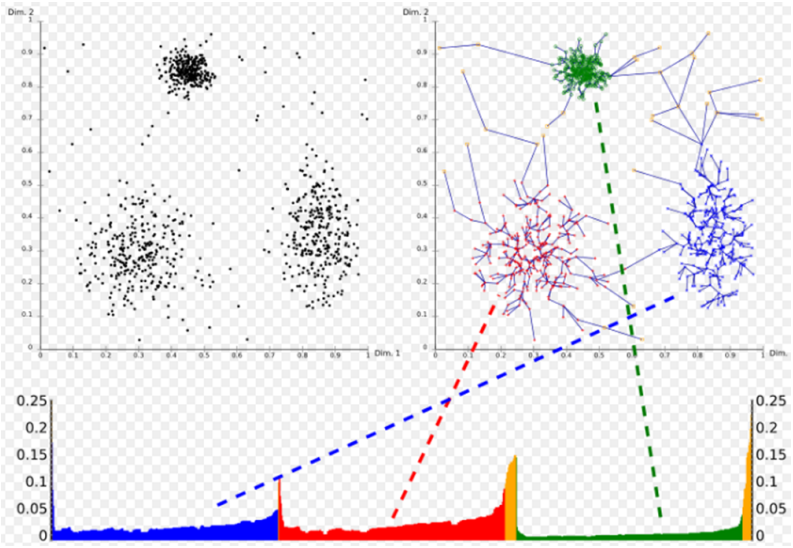


Figure 2.2. Reachability Plot

The algorithm processes the data once, expanding from core points and recording, for each visited point, its reachability-distance relative to a predecessor. Plotting the visitation order on the x-axis and the reachability-distance on the y-axis yields the *reachability plot*. Points within a cluster have small reachability to their nearest processed neighbor, so clusters appear as *valleys*; deeper valleys correspond to denser regions. Points with large reachability do not form valleys and are typically treated as noise (aside from the trivial "all-data" cluster in a hierarchical view).

Clusters can be extracted from this plot in several ways: (i) manual selection of x-ranges after visual inspection; (ii) applying a horizontal threshold on reachability (analogous to running DBSCAN at that level of density); or (iii) using automated procedures that detect valleys via steepness, knee points, or local-maximum heuristics. Care is required at valley boundaries; consulting the recorded predecessor helps assign the last points to the inner or outer cluster. The resulting clusterings are generally hierarchical and cannot be obtained by a single DBSCAN run.

Complexity

Like DBSCAN, OPTICS performs one ε-neighborhood query per point. With a spatial index that supports range queries in expected time $\mathcal{O}(\log n)$, the overall runtime is $\mathcal{O}(n \log n)$ for n points; the worst case is $\mathcal{O}(n^2)$. Choosing ε larger than the maximum interpoint distance makes every neighborhood query return the entire data set, again leading to quadratic behavior and extra overhead (e.g., heap maintenance). Hence ε should be set in line with the data scale and available indexing.

2.3 Engle–Granger test

Cointegration

In econometrics, *cointegration* refers to a long-run equilibrium linkage among two or more time series that are individually non-stationary (i.e., they contain stochastic trends) but share a stationary linear combination. Such series may wander apart in the short run, yet a particular linear aggregate remains $I(0)$, implying that the variables co-move over time around a stable equilibrium.

---

1 Formally, let $z_t = (z_{1t}, \dots, z_{mt})^\top$ be a vector of time series, each integrated of order $d \geq 1$ (they require d differences to be stationary). If there exists a nonzero vector β such that
2 $\beta^\top z_t \sim I(d-b) \qquad \text{with} \quad b > 0,$
3 then the components of z_t are *cointegrated*, and β is a *cointegrating vector*. When $d = 1$ and $b = 1$, the linear combination is $I(0)$ (stationary).

Cointegration is fundamental whenever one studies trending variables—for instance, macroeconomic aggregates or financial prices with persistent growth components.

History

The phenomenon of *spurious* or *nonsense* regression in trending series was already highlighted by Udny Yule (1926). Before the 1980s it was common to run linear regressions on non-stationary data, an approach that Granger and Newbold later showed can produce illusory correlations even after simple detrending. Engle and Granger (1987) formalized the concept of cointegration and demonstrated that, for integrated processes, de-trending alone does not remove the problem: one must test directly for cointegration. Modern time-series practice therefore (i) pre-tests all series for unit roots and (ii) uses cointegration-aware models whenever integrated variables are linked by a genuine equilibrium relation. Regressions that ignore these properties can be misleading.

Testing implications

When variables have unit roots, the possible presence of cointegration must be considered when testing hypotheses about their relationship. Historically, researchers often differenced the data and ran OLS on Δ-transformed variables. This strategy is inappropriate if the levels are cointegrated, because differencing discards the equilibrium relation and biases inference on long-run effects.

Engle–Granger two-step method

Consider two $I(1)$ series x_t and y_t. If they are cointegrated, there exists (α, β) such that the residual from the static regression

$$y_t = \alpha + \beta x_t + \varepsilon_t \quad (2.1)$$

is stationary, $\varepsilon_t \sim I(0)$.

Step 1 (residual-based cointegration test).

If β were known, one could test ε_t for stationarity with an Augmented Dickey–Fuller (ADF) or Phillips–Perron (PP) test. In practice, estimate β by OLS to obtain $\widehat{\varepsilon}_t = y_t - \widehat{\alpha} - \widehat{\beta} x_t$, then run an ADF/PP unit-root test on $\widehat{\varepsilon}_t$. Because $\widehat{\beta}$ is estimated, the critical values are *non-standard* (Engle–Granger) and become more negative in absolute value as additional regressors are included.

Step 2 (error-correction model).

If the null of *no* cointegration is rejected (i.e., $\widehat{\varepsilon}_t$ is stationary), the short-run dynamics can be modeled with an error-correction specification such as

$$\Delta y_t = \gamma\, \widehat{\varepsilon}_{t-1} + \sum_{i=1}^{p} \phi_i\, \Delta y_{t-i} + \sum_{j=1}^{q} \psi_j\, \Delta x_{t-j} + u_t. \quad (2.2)$$

Here γ measures the speed of adjustment toward the long-run equilibrium ($\gamma < 0$ indicates error correction).

No cointegration case. If the residual test fails to reject the null (no cointegration), the error-correction term is omitted and one estimates a differences model (e.g., a VAR in first differences):

$$\Delta y_t = \sum_{i=1}^{p} \phi_i\, \Delta y_{t-i} + \sum_{j=1}^{q} \psi_j\, \Delta x_{t-j} + u_t.$$

Other approaches

Alternative cointegration tests include the Johansen system method (trace and maximum-eigenvalue statistics), the Phillips–Ouliaris residual-based tests, and extensions to *multicointegration*. These provide complementary tools, especially in multivariate settings with more than one cointegrating relation.

2.4 Copulas

What is a Copula?

In probability and statistics, a *copula* is a multivariate cumulative distribution function (CDF) on the unit hypercube $[0,1]^d$ with *uniform* $(0,1)$ marginals. Copulas model the *dependence structure* among random variables, decoupled from their marginal distributions. This separation makes copulas attractive in high-dimensional settings: one can model/estimate marginals and dependence *separately* and then combine them.

Probability Integral Transform

Statement
Suppose that a random variable X has a continuous cumulative distribution function (CDF) F_X.
Define
$\mathrm{Y} := \mathrm{F}_X(X)$.
Then Y has a standard uniform distribution on $[0, 1]$.
Equivalently, if μ denotes the uniform measure on $[0, 1]$, the distribution of X on $\mathbb{R}$ is the pushforward measure $\mu \circ F_X^{-1}$.
Proof
Given any continuous random variable X, define $Y = F_X(X)$. For $y \in [0, 1]$, if the inverse $F_X^{-1}(y)$ exists (i.e., there is a unique x such that $F_X(x) = y$), then

$$\begin{aligned} F_Y(y) &= \mathbb{P}(Y \le y) \\ &= \mathbb{P}(F_X(X) \le y) \\ &= \mathbb{P}(X \le F_X^{-1}(y)) \\ &= F_X(F_X^{-1}(y)) \\ &= y. \end{aligned}$$

If $F_X^{-1}(y)$ does not exist as a single-valued inverse, replace it with the left-continuous quantile function
$\chi(0) = -\infty, \qquad \chi(1) = \infty, \qquad \chi(y) = \inf\{x \in \mathbb{R} : F_X(x) \ge y\}$ for $y \in (0, 1)$,
and the same calculation yields $F_Y(y) = y$. Hence F_Y is the CDF of a Uniform$(0, 1)$ random variable, so Y is uniform on $[0, 1]$.

Definition

Probabilistic
A function $C : [0, 1]^d \to [0, 1]$ is a d-dimensional *copula* iff it is a joint CDF on the unit cube with *uniform* marginals:
C(1,. . . ,1,u,1,. . . ,1)=u for each coordinate u$\in [0, 1]$.
Analytic
Equivalently, C is a d-dimensional copula iff:
(1) **Groundedness:** $C(u_1, \ldots, u_{i-1}, 0, u_{i+1}, \ldots, u_d) = 0$ for any i and any $u_j \in [0, 1]$.
(2) **Uniform margins:** $C(1, \ldots, 1, u, 1, \ldots, 1) = u$ for each coordinate $u \in [0, 1]$.
(3) d**-increasing:** for every hyperrectangle $B = \prod_{i=1}^{d} [x_i, y_i] \subset [0, 1]^d$,
$\int_B dC(u) \ge 0$,
and, by inclusion–exclusion, this C-volume equals
$\sum_{z \in \prod_{i=1}^{d} \{x_i, y_i\}} (-1)^{N(z)} C(z) \ge 0, \qquad N(z) := \#\{\, k : \ z_k = x_k \,\}$.

Bivariate special case ($d = 2$).

A function $C : [0,1]^2 \to [0,1]$ is a copula iff
C(0,u)=C(u,0)=0, C(1,u)=C(u,1)=u for all u$\in [0,1]$,
and for all $0 \le u_1 \le v_1 \le 1$, $0 \le u_2 \le v_2 \le 1$,
$C(v_1, v_2) - C(u_1, v_2) - C(v_1, u_2) + C(u_1, u_2) \ge 0$.

Sklar's Theorem)

Let X and Y be real-valued random variables with joint cumulative distribution
$F(x,y) = P(X \le x,\ Y \le y)$,
and marginal CDFs $F_X(x) = \mathbb{P}(X \le x)$ and $F_Y(y) = \mathbb{P}(Y \le y)$.
Define the probability–integral transforms
$u = F_X(x), \qquad v = F_Y(y)$.
Sklar's theorem states that there exists a copula $C : [0,1]^2 \to [0,1]$ such that
$F(x,y) = C(F_X(x),\ F_Y(y)) = C(u, v)$.
Indeed, if we set $U = F_X(X)$ and $V = F_Y(Y)$, then

$$\begin{aligned} C(u,v) &= \mathbb{P}(U \le u,\ V \le v) \\ &= \mathbb{P}(F_X(X) \le F_X(x),\ F_Y(Y) \le F_Y(y)) \\ &= \mathbb{P}(X \le x,\ Y \le y) \\ &= F(x,y). \end{aligned}$$

If F is differentiable with joint density f, and the marginals admit densities f_X and f_Y, and if the copula C is absolutely continuous with density
$c(u,v) = \partial^2 \overline{\partial u\, \partial v\, C(u,v),}$
then differentiating $F(x,y) = C(F_X(x), F_Y(y))$ yields the well-known factorization:
$\partial F(x,y) \overline{\partial x = \frac{\partial C(u,v)}{\partial u}\, \frac{\partial u}{\partial x} = \frac{\partial C(u,v)}{\partial u}\, f_X(x),}$
and differentiating again with respect to y,
$f(x,y) = \partial \overline{\partial y\big(\frac{\partial C(u,v)}{\partial u}\, f_X(x)\big) = \frac{\partial^2 C(u,v)}{\partial u\, \partial v}\, f_X(x)\, f_Y(y) = c(u,v)\, f_X(x)\, f_Y(y).}$

The lowercase c denotes the *copula density* (not the copula CDF). This identity shows how the joint density decomposes into the product of the marginal densities and the copula density, which captures all cross-sectional dependence between X and Y after marginal effects are removed.

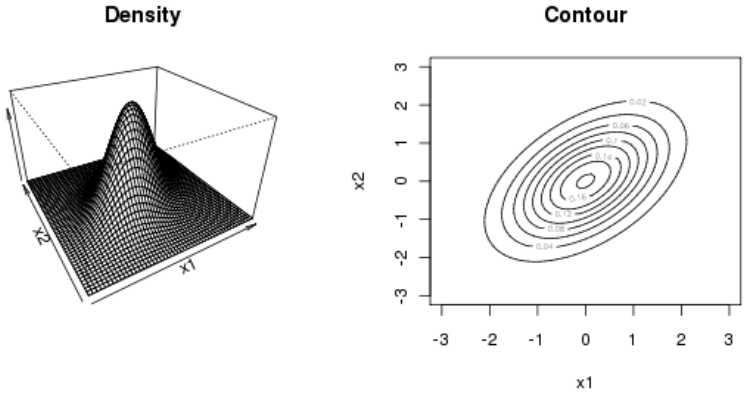


Figure 2.3. Bivariate Normal Distribution (rho=0.5)

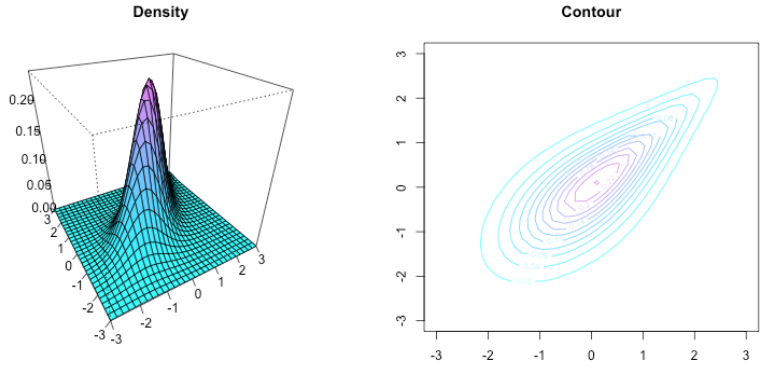


Figure 2.4. Gumble copula (param=2) with normal marginals

Practical Considerations for Time Series

When modeling cross-series dependence with copulas, serial autocorrelation, trends, and seasonality can confound inference. A common workflow transforms each series to approximately i.i.d. innovations (e.g., via ARMA/GARCH/filtering) before fitting the copula; otherwise, apparent cross-dependence may reflect within-series persistence rather than genuine cross-sectional structure.

Gaussian Copula

Let R be a $d \times d$ correlation matrix and Φ the standard normal CDF. The Gaussian copula is

$$\mathrm{C}_R(u_1, \ldots, u_d) = \Phi_R\big(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\big),$$

where Φ_R is the CDF of $\mathcal{N}(0, R)$. Its density is

$$\mathrm{c}_R(u) = \frac{1}{\sqrt{\det R}} \exp\Big\{-\tfrac{1}{2}\, z^\top (R^{-1} - I)\, z\Big\}, \qquad z_k = \Phi^{-1}(u_k),\ I = \mathrm{Id}.$$

The Gaussian copula flexibly captures linear correlation but has zero tail dependence unless $|R_{ij}| = 1$.

Archimedean Copulas

An Archimedean copula has the form
$\mathrm{C}(\mathrm{u}_1, \ldots, u_d) \;=\; \psi^{-1}\big(\psi(u_1) + \cdots + \psi(u_d)\big)$,
where the *generator* $\psi : [0,1] \to [0,\infty]$ is continuous, strictly decreasing, convex, and satisfies $\psi(1) = 0$. The pseudo-inverse is
$\psi^{-1}(t) \;=\; \inf\{\, u \in [0,1] :\; \psi(u) \le t \,\}$.
This representation yields a copula in d dimensions if and only if ψ is d-monotone on $[0,\infty)$ (suitable smoothness and alternating-sign derivative conditions). Archimedean families are popular because they admit closed forms in any d and often use a single parameter controlling dependence strength.

Common bivariate Archimedean copulas (with generator)

Clayton $C(u,v) = \big(u^{-\theta} + v^{-\theta} - 1\big)^{-1/\theta}$, $\theta \in [-1,\infty) \setminus \{0\}$,
$\psi(u) = \frac{u^{-\theta}-1}{\theta}, \quad \psi^{-1}(t) = (1+\theta t)^{-1/\theta}$;

Frank $C(u,v) = -\frac{1}{\theta}\log\!\Big(1 + \frac{(\mathrm{e}^{-\theta u}-1)(\mathrm{e}^{-\theta v}-1)}{\mathrm{e}^{-\theta}-1}\Big)$, $\theta \in \mathbb{R} \setminus \{0\}$,
$\psi(u) = -\log\!\Big(\frac{\mathrm{e}^{-\theta u}-1}{\mathrm{e}^{-\theta}-1}\Big), \quad \psi^{-1}(t) = -\frac{1}{\theta}\log\big(1 + \mathrm{e}^{-t}(\mathrm{e}^{-\theta}-1)\big)$;

Gumbel $C(u,v) = \exp\!\Big(-\big[(-\log u)^{\theta} + (-\log v)^{\theta}\big]^{1/\theta}\Big)$, $\theta \in [1,\infty)$,
$\psi(u) = (-\log u)^{\theta}, \quad \psi^{-1}(t) = \exp(-t^{1/\theta})$.

Expectations Under Copula Models and Monte Carlo

Let H be a joint CDF that factors as $H(x) = C(F_1(x_1), \ldots, F_d(x_d))$. For a response function $g : \mathbb{R}^d \to \mathbb{R}$,
$\mathrm{E}\,[\mathrm{g}(\mathrm{X})] \;=\; \int_{\mathbb{R}^d} g(x)\, dH(x) \;=\; \int_{[0,1]^d} g\big(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d)\big)\, c(u)\, du$,
where c is the copula density (if it exists). If in addition each F_k has density f_k,
$\mathrm{E}\,[\mathrm{g}(\mathrm{X})] \;=\; \int_{\mathbb{R}^d} g(x)\, c\big(F(x)\big) \prod_{k=1}^{d} f_k(x_k)\, dx$.
Monte Carlo approximation.
(1) Sample $U^{(1)}, \ldots, U^{(n)} \sim C$ on $[0,1]^d$.
(2) Set $X_k^{(i)} = F_k^{-1}(U_k^{(i)})$ for $k = 1, \ldots, d$.
(3) Approximate $\mathbb{E}[g(X)]$ by $\frac{1}{n}\sum_{i=1}^{n} g\big(X^{(i)}\big)$.

Empirical Copulas and Pseudo-Observations

Given data $\{(X_1^i, \ldots, X_d^i)\}_{i=1}^n$ with continuous marginals, define empirical CDFs

$$\mathrm{F}_k^{(n)}(x) \;=\; \tfrac{1}{n}\sum_{j=1}^n \mathbf{1}\{X_k^j \le x\}, \qquad k = 1, \ldots, d,$$

and pseudo-observations $\tilde{U}_k^i = F_k^{(n)}(X_k^i)$. Equivalently, with ranks $R_k^i = \sum_{j=1}^n \mathbf{1}\{X_k^j \le X_k^i\}$, one may use $\tilde{U}_k^i = R_k^i/n$ (or $(R_k^i - 0.5)/n$, $R_k^i/(n+1)$). The *empirical copula* is

$$\mathrm{C}^{(n)}(u_1, \ldots, u_d) \;=\; \tfrac{1}{n}\sum_{i=1}^n \mathbf{1}\{\tilde{U}_1^i \le u_1, \ldots, \tilde{U}_d^i \le u_d\}.$$

For $d = 2$, rank measures such as Spearman's ρ are functionals of $C^{(n)}$:

$$\widehat{\rho}_S \;=\; \tfrac{12}{n^2-1}\sum_{i=1}^n \sum_{j=1}^n \left[C^{(n)}\left(\tfrac{i}{n}, \tfrac{j}{n}\right) - \tfrac{i}{n}\cdot\tfrac{j}{n}\right].$$

Quantitative Finance Applications

Copulas are used to separate marginal behaviors (individual asset returns, losses, or risk factors) from their dependence. Typical tasks include:

- **Risk management and stress testing:** modeling joint tail behavior and contagion; scenario design for downside or crisis regimes (flight-to-quality), where cross-asset correlations tend to rise.
- **Portfolio construction:** optimizing allocations using dependence beyond linear correlation (e.g., tail dependence asymmetries).
- **Derivatives and credit:** joint default modeling, tranche pricing, and multi-asset option valuation via copula-based dependence.

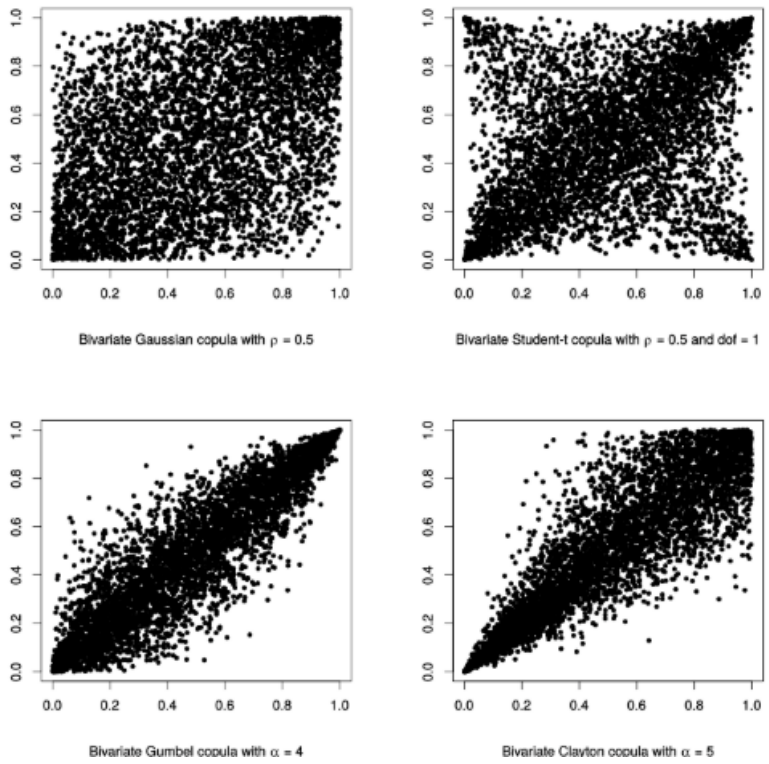


Figure 2.5. Examples of bivariate copulæ used in finance

Selected Copula Families and Reported Signal-Processing Uses

Below are examples of bivariate copula densities $c(u, v)$ and representative application areas where such models have been reported:

- **Gaussian copula:** used in supervised and unsupervised classification of radar/SAR data, biometric validation, and dependence modeling in wind-power integration.
- **Exponential and log-normal copulas:** appear in queueing models and wireless-channel modeling; a bivariate log-normal copula is closely related to the Gaussian copula under log-transformations.
- **Clayton copula:** employed in location estimation and heterogeneous-sensor data fusion with asymmetric (lower-tail) dependence.
- **Frank copula:** applied to quantitative risk assessment in geophysical hazards.
- **Student-t copula:** used in SAR image classification and fusion of correlated sensor decisions; notable for nonzero symmetric tail dependence.

2.5 CUSUM

The *Cumulative Sum* (CUSUM) procedure is a sequential technique for monitoring a process and detecting shifts in its mean. It is widely used in quality control, signal processing, and, more generally, in time–series surveillance.

Setup and Basic Idea

Let $\{X_t\}_{t\geq 1}$ be observations from a process with in-control target mean μ_0 (and, if needed, known scale σ). CUSUM accumulates deviations from the target; small departures that might be invisible to single–sample charts can build up into a detectable signal.

A basic two-sided cumulative sum is

$$S_t \;=\; \sum_{i=1}^{t}(X_i-\mu_0) \qquad (S_0=0),$$

and persistent positive (negative) drift in S_t indicates an upward (downward) shift in the mean.

Basic CUSUM Formula

1 In practice one uses one-sided statistics with a *reference value* K (sometimes called the allowance or slack) and a *decision interval* (threshold) $h>0$. Working either on raw deviations or on standardized scores $Z_t=(X_t-\mu_0)/\sigma$, the recursive form is

2 $$C_t^{+}=\max\{0,\; C_{t-1}^{+}+(X_t-\mu_0)-K\}, \qquad C_0^{+}=0,$$
$$C_t^{-}=\max\{0,\; C_{t-1}^{-}+(\mu_0-X_t)-K\}, \qquad C_0^{-}=0.$$

3 A signal is raised when either statistic crosses its control limit:

4 Signal at time t if $\mathrm{C}_t^{+}\geq h$ or $C_t^{-}\geq h$.

5 The usual heuristic is $K\approx\frac{\delta\sigma}{2}$ when targeting a shift of size $\delta\sigma$; larger h gives fewer false alarms but slower detection.

Average Run Length (ARL)

The *Average Run Length* quantifies the expected number of samples until a signal occurs. If α denotes the per-sample false-alarm probability under the in-control regime, a common summary is

$$\mathrm{ARL}_0 \;=\; \frac{1}{\alpha}.$$

Higher ARL_0 means greater resistance to false alarms; lower ARL under a shifted mean indicates faster detection.

Advantages

- **High sensitivity to small shifts:** cumulative evidence builds quickly even when single observations are only mildly deviant.
- **Cumulative effect:** persistent bias produces a clear trend in $C_t^{\pm}$ that other charts may miss.
- **Versatility:** readily adapted to many domains (manufacturing QC, finance, healthcare monitoring).

Limitations

- **Tuning required:** performance depends on informed choices of K and h relative to the shift sizes of interest.
- **Complexity vs. simple charts:** conceptually and computationally richer than, e.g., Shewhart charts.
- **False alarms if mistuned:** poor parameter choices can lead to excessive signaling.

Example Use Cases

- **Quality control:** detect drifts in a production line's mean output.
- **Finance:** flag structural changes in price series or indicators.
- **Healthcare:** monitor vital signs for early departures from baseline.

Chapter 3

Python Implementation

3.1 Overview and Design Principles

This chapter documents the Python implementation that operationalizes the methodology presented in the thesis. Rather than reporting low-level code, the discussion ties each software block to its statistical rationale, clarifies key hyperparameters, and explains how results are produced and audited.

Universe and time span

The empirical analysis focuses on the **400 largest companies by market capitalization listed on NASDAQ**, the U.S. electronic stock exchange historically dominated by technology and growth-oriented firms, but now hosting a diversified set of industries.
For each issuer we assembled three synchronized data sources:

- (i) daily *price histories*,
- (ii) firm-level *financial statements* (balance sheet, income statement, and cash-flow items),
- (iii) a composite *ESG score*

Because the trading and modeling pipeline requires complete information, we retained only the firms for which *all three* datasets were available across the sample period; after this integrity check the final investable universe consists of **343 companies**.

Code output (dataset integrity check)

```
====================[ Detected ZIP files ]====================
 - Balance sheets : /content/NASDAQ400 - BALANCE SHEET.zip
 - ESG            : /content/ESG Table for NDAQ400.zip
 - Prices         : /content/Price History - NVDA400.zip
------------------------------------------------------------
 Counts           :  Balance sheets = 354 | ESG = 377 | Prices = 387
 Intersection     :  Tickers common to ALL THREE = 343
============================================================
```

The analysis period spans from FY2017 to FY2023.

Pipeline

The implementation is organized in four macro components:

1. *Data curation and pair pre-selection*: robust ingestion of price series, harmonization of tickers and calendars, integration of firm descriptors (with special emphasis on ESG scores), and unsupervised clustering for candidate discovery (baseline *k-means* and final *OPTICS*).

2. *Dependence modeling*: estimation and selection of univariate marginals (Normal, Student-t, Logistic, GED) and bivariate copulas (Gaussian, Student-t, Frank, Clayton, Gumbel) on a dedicated *training* window, with AIC/BIC model choice.

3. *Trading logic*: mean-reversion on a normalized spread, with entries gated by a *rare-event* criterion derived from copula log-likelihoods and an optional *CUSUM* filter for regime shifts.

4. *Portfolio reporting*: daily equity aggregation, trade-level and daily metrics, cost-aware performance in both currency and percentage terms, CUSUM intervention counts, and side-by-side comparison of CUSUM vs. NO-CUSUM portfolios.

All figures and tables referenced in this chapter are produced directly by the implemented Python code.

3.2 ESG Variables, Data and Pre-processing

3.2.1 Why ESG matters for correlation and dependence

A distinctive feature of our implementation is the integration of firm-level ESG (Environmental, Social, Governance) scores alongside market data. ESG metrics condense slow-moving, structural characteristics of firms (e.g., transition risk, governance quality, human-capital practices) that shape exposure to common risk drivers and thus inform *persistent* co-movement patterns.

ESG information and a "rational sustainability" lens

In our implementation the ESG score is *not* treated as a return-predictive factor or a screening label. Instead, it plays a strictly informational role that enlarges the investor's state space in a disciplined, evidence-based way. We use this scores during pair discovery, as an auxiliary feature that helps the unsupervised algorithm avoid spurious proximity driven by transient price co-movements.

The rationale is "rational sustainability": ESG expands the information set and can be exploited when it is *material* for long-term co-movement, without presuming that higher ESG is inherently "better" or that it should mechanically drive portfolio tilts. In practice this yields two benefits. First, it provides an economically grounded prior for *structural relatedness*: firms with similar governance practices, stakeholder policies, or environmental exposures are more likely to share persistent cash–flow and risk channels, which can manifest as both linear and tail dependence captured by copulas. Second, it enhances the *interpretability* of trading signals and interventions (e.g., CUSUM blocks can often be traced back to ESG-relevant corporate events, policy shifts, or supply-chain shocks).

Crucially, ESG is never a direct trigger for trades and never overrides price-based evidence. It serves as a structural covariate to guide discovery, to rationalize dependence estimates, and to contextualize performance, consistent with an outcome-oriented, evidence-driven approach rather than a label-driven one.

3.3 Price Series Ingestion, Annualization, and Balance-Sheet Feature Selection

End-of-day close prices for the NASDAQ sample are ingested through a defensive reader that (i) normalizes tickers (uppercasing, removal of common vendor suffixes, stripping non–alphanumerics), (ii) auto-detects `Date` and `Close` columns across heterogeneous CSV/Excel headers, and (iii) enforces strict date parsing and chronological sorting. When two tickers are merged, the code takes the *intersection* of trading days to avoid calendar misalignment.

Annualization for pair pre-selection

because pair selection is performed on a *yearly* basis, daily prices are aggregated to two annual metrics per ticker and per year: the *annualized mean log-return* and the *annualized volatility*.
Let P_t denote the close price and $r_t = \Delta \log P_t = \log P_t - \log P_{t-1}$ the one-step log-return. For a given ticker i and year y with n_y trading days, Because the number of trading days varies across years and tickers (holidays, suspensions, data gaps), annualization uses the *effective* count of observations n_y rather than a fixed 252:

$$\bar{r}_{i,y}^{(\mathrm{ann})} = n_y \cdot \frac{1}{n_y} \sum_{t \in y} r_{i,t} \; = \; \sum_{t \in y} r_{i,t}, \tag{3.1}$$

$$\sigma_{i,y}^{(\mathrm{ann})} = \sqrt{n_y} \; \cdot \; \mathrm{sd}(\{r_{i,t}\}_{t \in y}) \, , \tag{3.2}$$

Using n_y avoids small biases when $n_y \neq 252$ and is robust to missing days. This choice aligns the price-based signals to the other yearly covariates used in pre-selection (balance-sheet items and ESG scores).

Code output — first 15 tickers for the annualized mean return pivot:

```
=== Pivot: Annualized Mean Return (Ticker x Year) ===
Year         2017       2018      2019      2020      2021      2022      2023
Ticker
AAL      0.112941  -0.382856 -0.106820 -0.450139  0.138871 -0.291759  0.080189
AAON     0.132716  -0.044687  0.409298  0.348512  0.192106 -0.051744 -0.019251
AAPL     0.410250  -0.067896  0.861608 -0.548136  0.338232 -0.268289  0.481798
ABNB          NaN        NaN       NaN  0.014443  0.134128 -0.486456  0.592281
ACGL     0.046340  -0.705630  0.605165 -0.159011  0.232326  0.412373  0.183020
ADBE     0.622593   0.291029  0.457788  0.516388  0.133848 -0.406535  0.772799
ADI      0.231738  -0.035943  0.384597  0.243100  0.189806 -0.066792  0.210510
ADP      0.136994   0.118867  0.300336  0.033431  0.399432 -0.031308 -0.024659
ADSK     0.333376   0.226843  0.426483  0.664341 -0.079092 -0.335432  0.302938
AEP      0.152954   0.015903  0.264517 -0.118929  0.068452  0.067214 -0.144602
AGNC     0.064873  -0.131253  0.007982 -0.117647 -0.035897 -0.311835 -0.052174
```

```
AGNCM         NaN        NaN  0.037842 -0.035299  0.035786 -0.229425  0.184383
AKAM    -0.068061  -0.060886  0.414211  0.215443  0.114773 -0.279733  0.403915
ALGN     1.330501  -0.057428  0.332378  0.915066  0.229799 -0.679083  0.299194
ALNY     2.324176  -0.426131  0.579619  0.128506  0.304763  0.401403 -0.194572
```

Console dump (truncated) — first 15 tickers for the annualized volatility pivot:

```
=== Pivot: Annualized Volatility (Ticker x Year) ===
Year       2017     2018     2019     2020     2021     2022     2023
Ticker
AAL    0.281636 0.402336 0.356715 1.012193 0.468802 0.562767 0.350591
AAON   0.245041 0.320241 0.309289 0.507623 0.245777 0.387453 0.559742
AAPL   0.174904 0.287215 0.263392 1.433285 0.250945 0.355632 0.201933
ABNB        NaN      NaN      NaN 0.182766 0.526770 0.608783 0.420023
ACGL   0.138021 1.139798 0.174743 0.578868 0.227342 0.276658 0.255399
ADBE   0.212340 0.353868 0.238078 0.479077 0.296812 0.456415 0.316601
ADI    0.200987 0.274083 0.288506 0.505593 0.272255 0.366507 0.260608
ADP    0.192959 0.227713 0.182209 0.452760 0.192559 0.283269 0.211212
ADSK   0.299557 0.405993 0.293170 0.506030 0.349829 0.473019 0.314378
AEP    0.106419 0.174265 0.137648 0.385892 0.166704 0.234775 0.208176
AGNC   0.133620 0.136800 0.129645 0.504175 0.173627 0.354970 0.291651
AGNCM       NaN      NaN 0.046618 0.648936 0.085541 0.214090 0.169553
AKAM   0.342102 0.317202 0.220060 0.382191 0.257482 0.300072 0.237977
ALGN   0.336086 0.478575 0.488403 0.692434 0.377758 0.606412 0.558045
ALNY   0.641365 0.515288 0.422876 0.446743 0.488924 0.597854 0.341851
```

Balance-sheet variable set and Spearman-based feature selection

From firm financial statements we initially extracted the following yearly variables:

- *Capital Lease Maturities – Total*
- *Debt - Long-Term - Maturities – Total*
- *Debt including Finance and Operating Lease Liabilities*
- *Debt – Total*
- *Employees - Full-Time/Full-Time Equivalents - Period End*
- *Investments – Total*
- *Loans & Receivables – Total*
- *Operating Lease Payments – Total*
- *Payables & Accrued Expenses*
- *Property, Plant & Equipment - excluding Right of Use Tangible Assets & Capital Leases – Net*
- *Right of Use Tangible Assets - Total - Net – Supplemental*
- *Total Assets*
- *Total Liabilities*

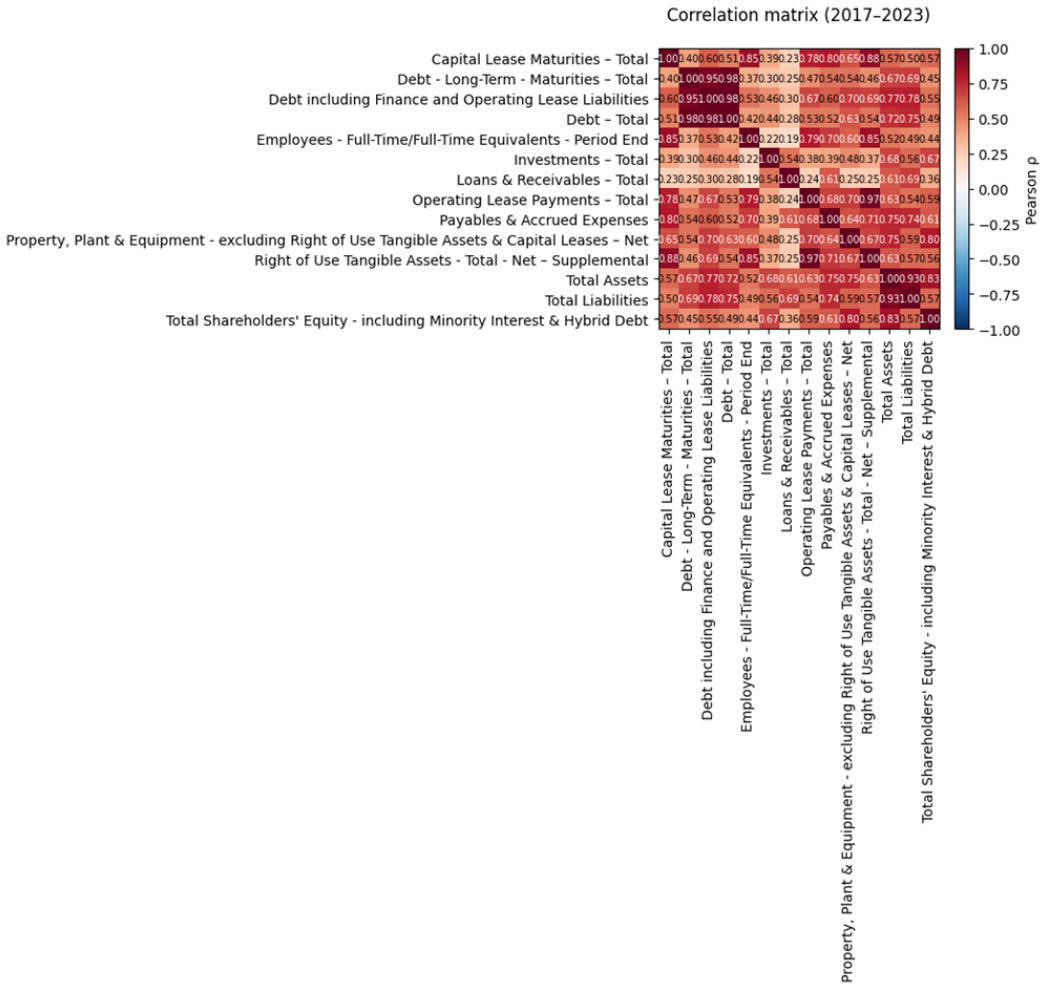
- *Total Shareholders' Equity - including Minority Interest & Hybrid Debt*

To reduce redundancy and improve cluster interpretability, we applied a *Spearman rank-correlation* screen across features. Recall that Spearman's ρ between two random variables X and Y is

$$\rho_s(X, Y) \;=\; \mathrm{corr}\big(\mathrm{rank}(X),\, \mathrm{rank}(Y)\big), \tag{3.3}$$

and, when no ties are present, can be written as

$$\rho_s \;=\; 1 \;-\; \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, \qquad d_i \equiv \mathrm{rank}(X_i) - \mathrm{rank}(Y_i). \tag{3.4}$$



We computed the pairwise Spearman matrix on the pooled (ticker, year) panel and retained variables with absolute correlation ≤ 0.75. The selected set is:

- *Debt - Long-Term - Maturities – Total*
- *Employees - Full-Time/Full-Time Equivalents - Period End*

- *Investments – Total*
- *Loans & Receivables – Total*
- *Payables & Accrued Expenses*
- *Total Liabilities*
- *Total Shareholders' Equity (incl. Minority Interest & Hybrid Debt)*

3.4 Pair Discovery: from k-means to OPTICS

3.4.1 Baseline: k-means clustering

Given the data preparation described above, we now dispose of **ten yearly features per firm** with which to perform the first, purely unsupervised, pair pre–selection via k-means. The ten features are:

1. Annualized mean return (from daily log–returns).
2. Annualized volatility (from daily log–returns).
3. *Debt – Long-Term – Maturities – Total.*
4. *Employees — Full-Time/Full-Time Equivalents — Period End.*
5. *Investments – Total.*
6. *Loans & Receivables – Total.*
7. *Payables & Accrued Expenses.*
8. *Total Liabilities.*
9. *Total Shareholders' Equity — including Minority Interest & Hybrid Debt.*
10. **ESG Combined score**.

All variables are aligned at the *yearly* frequency (2017–2023) and standardized before clustering to prevent scale effects.

Choosing the number of clusters k

We determine k by combining two complementary diagnostics:

- **Elbow method:** For $k = 2, 3, \ldots,$ we compute the within–cluster sum of squares (inertia)
$$\mathrm{WCSS}(k) = \sum_{c=1}^{k} \sum_{i \in \mathcal{C}_c} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2,$$
and select k near the *elbow*, i.e., where marginal reductions in WCSS become negligible.

- **Silhouette score:** For each observation i, with $a(i)$ the mean intra–cluster distance and $b(i)$ the smallest mean distance to any other cluster, the silhouette is

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}} \in [-1, 1].$$

We choose k that maximizes the average $\bar{s}(k) = \frac{1}{n}\sum_i s(i)$, favouring compact and well–separated clusters. (Figure to be inserted.)

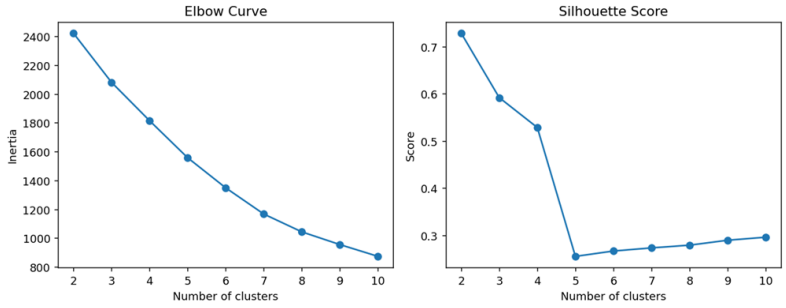


Figure 3.1. Elbow Curve and Silhouette Score

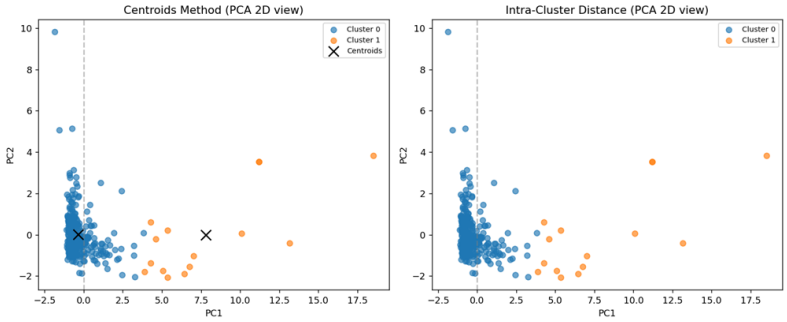
Both diagnostic curves point to a very coarse partition.
Elbow: the within–cluster inertia exhibits its most pronounced drop by moving to $k = 2$; beyond $k = 2$ the curve decays almost linearly with no clear secondary elbow, indicating diminishing returns from additional clusters.
Silhouette: the average silhouette score attains its *global maximum* at $k = 2$ (substantially higher than for $k \geq 3$) and decreases steadily thereafter, signaling poorer separation and coherence when more clusters are imposed. On the joint evidence of parsimony (elbow) and separability (silhouette), we therefore **set** $k = 2$ for the k-means pre–selection stage used in our analysis.

Pairing rules used with k-means

In our baseline we employ *two* pairing criteria *within* each k-means cluster; both operate on Euclidean distances in the 10-dimensional feature space:

1. **Centroid rule:** Assets are ranked by distance to the cluster centroid $\boldsymbol{\mu}_c$; pairs are formed by greedily matching the closest names around the centroid (disjoint matching). This yields *centroid* portfolios.

2. **ICD rule (Intra–Cluster Distance):** We compute all pairwise distances inside the cluster and select disjoint pairs that minimize total intra–cluster distance (nearest–neighbour matching). These are the *ICD* portfolios.



Code output — k-means cluster sizes ($k = 2$):

```
Cluster sizes:
Cluster
0    325
1     15
```

Observed limitations

The k-means output exhibits *highly unbalanced* cluster sizes (as visible in the accompanying plots): a few large, heterogeneous clusters coexist with many small ones. In practice this may (i) bias discovery toward dense regions of the feature space, (ii) mix assets with different local densities into the same large cluster (hurting pair homogeneity), and (iii) make the centroid rule sensitive to outliers. These observations motivate the transition to a density–based approach (OPTICS) in the subsequent section, which is better suited to non-convex, variable-density structures commonly encountered in equity universes.

3.4.2 Final choice: OPTICS

To overcome the limitations observed with k-means (notably the two highly unbalanced groups and its spherical, equal-density bias), we adopt *OPTICS* (Ordering Points To Identify the Clustering Structure), a density-based method that reveals multi-scale cluster structure and explicitly separates noise. Two hyperparameters govern the extraction of clusters from the reachability plot:

- the **steepness** parameter $\boldsymbol{\xi} \in \{0.05, 0.10\}$, which controls how sharp a descent in reachability distance must be to delimit a valley (cluster). Larger ξ yields fewer, coarser clusters; smaller ξ admits finer structure but can elevate the number of noise points;
- the **minimum cluster size MCS** $\in \{0.05, 0.03\}$, specified as a fraction of the sample, which regularizes the smallest admissible basin on the reachability landscape. Higher MCS suppresses thin/sparse groups in favor of more stable aggregates.

Crossing ξ and MCS produces the four portfolio tags used throughout the trading experiments: `ICD_XI005_MCS005`, `ICD_XI010_MCS003`, `MEDOID_XI005_MCS005`, `MEDOID_XI010_MCS003`.

Code output

Below we report the console snippets produced by the implementation for the two configurations; the counts include per-cluster cardinalities, number of noise points, and the resulting silhouette score.

Run with $\xi = 0.10$, MCS $= 0.03$ *(tag:* `xi010_mcs003`*)*

```
=== Running OPTICS -> tag=xi010_mcs003 ===
clusters=4 | noise=91 | silhouette=0.3504
cluster sizes (members):
  - Cluster 0 : 168
  - Cluster 1 : 55
  - Cluster 2 : 17
  - Cluster 3 : 12
  ----------------------------------------
```

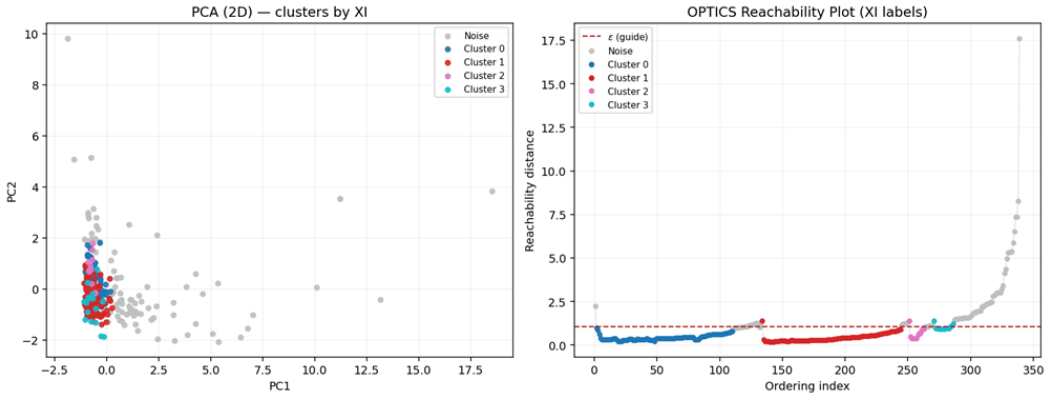


Figure 3.2. OPTICS with $\xi = 0.10$, MCS $= 3\%$: PCA embedding with cluster labels (left) and reachability plot (right).

Run with $\xi = 0.05$, MCS $= 0.05$

```
=== Running OPTICS - tag=xi005_mcs005 ===
clusters=3 | noise=212 | silhouette=0.4110
cluster sizes (members):
  - Cluster 0 : 96
  - Cluster 1 : 23
  - Cluster 2 : 12
  ----------------------------------------
```

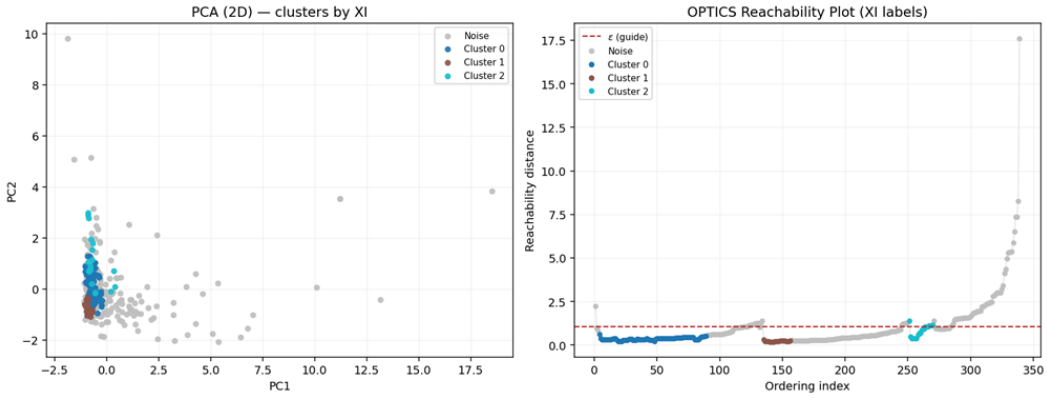


Figure 3.3. OPTICS with $\xi = 0.05$, MCS $= 5\%$: PCA embedding with cluster labels (left) and reachability plot (right).

Interpretation tied to the hyperparameters

With $\xi = 0.10$ and MCS $= 3\%$, OPTICS extracts four relatively compact groups (252 names) and isolates 91 securities as noise. The moderate silhouette (≈ 0.35) reflects clearly separated dense cores plus transitional regions (visible as gentle slopes in the reachability plot). Tightening the minimum cluster size to 5% and lowering the steepness to $\xi = 0.05$ enforces stricter density and shape requirements: only three clusters survive (131 names) while many borderline names migrate to noise (212), and cohesion improves (silhouette ≈ 0.41). In practice:

- **Higher** MCS filters out thin, potentially unstable groups, trading interpretability for coverage.
- **Lower** ξ recognizes finer valleys but, combined with larger MCS, will only keep valleys that are both sharp and sufficiently populated.
- **Noise points** (gray in the PCA panels; high reachability) capture idiosyncratic or regime-switching tickers that would otherwise distort centroid/medoid selection.

Representative selection for trading.

As in the k-means stage, we form two portfolios per OPTICS setting by changing the representative rule:

1. **ICD portfolios**: pairs anchored around the cluster *centers of mass* in feature space, favoring prototypical names.
2. **MEDOID portfolios**: pairs built around *medoids*, i.e., actual tickers minimizing average in-cluster dissimilarity, which improves robustness against outliers and non-spherical shapes.

These four base universes feed the subsequent dependence estimation and trading stage; their different balance between coverage and cohesion (as quantified by the counts and silhouette values above) directly influences the stability of fitted marginals/copulas and the reliability of rare-event gates used for entries.

3.5 Cointegration testing and its role

Pairs trading is traditionally motivated by the hypothesis that two price processes share a stable long–run equilibrium: if (y_t^A, y_t^B) are cointegrated, some linear combination is stationary and the deviation from equilibrium (the *spread*) tends to mean–revert. After forming candidate sets with OPTICS, we subject every pair to an Engle–Granger (EG) two–step check. This stage is not a theoretical end in itself; rather, it is a *sanity filter* that prioritizes pairs for which the subsequent copula layer and trading logic are most informative.

Construction of the spread and normalization.

Let P_t^A and P_t^B be end-of-day closes. We work in logs

$$y_t^A = \log P_t^A, \qquad y_t^B = \log P_t^B.$$

On the *training window* $[t_0, t_1]$ we estimate the hedge ratio by OLS on log–prices,

$$y_t^A = \alpha + \beta\, y_t^B + \varepsilon_t,$$

and *construct* the spread without intercept as

$$s_t \;=\; y_t^A - \beta\, y_t^B.$$

We then standardize it using training moments,

$$z_t \;=\; \frac{s_t - \mu_s}{\sigma_s}, \qquad \mu_s = \mathbb{E}[s_t]_{t\in[t_0,t_1]}, \quad \sigma_s = \mathrm{sd}(s_t)_{t\in[t_0,t_1]},$$

which is the parsimonious proxy for mean–reversion used by the trading engine.

Why an Engle–Granger test, and why two criteria.

The EG procedure (i) estimates β as above and (ii) tests the residual/spread for a unit root with an ADF regression,

$$\Delta s_t \;=\; \gamma\, s_{t-1} + \sum_{i=1}^{p} \phi_i\, \Delta s_{t-i} + u_t, \quad H_0 : \gamma = 0 \text{ (unit root)},\ H_1 : \gamma < 0.$$

Implementationally, the library returns an EG t–statistic and p–value based on MacKinnon critical values. We *also* compute a plain ADF p–value on s_t as a

cross–check. Requiring EG $p < 0.10$ *and* ADF $p < 0.05$ reduces false positives: the first exploits EG's residual–based distribution, the second demands conventional stationarity evidence on the same spread. Pairs passing both are promoted to the copula layer, which then governs *when* deviations are informative via rare–event gating.

Code output

The following console snippets summarize the EG screen for each candidate universe. For each portfolio we show the top 15 pairs by EG p–value, and report the number of passing pairs under the joint criterion. *ICD_XI005_MCS005*

```
=== Top 15 by EG p-value - ICD_XI005_MCS005 ===
TickerA TickerB  EG_pval  EG_tstat  ADF_resid_pval     beta
  LBRDA   LBRDK 0.000008 -5.679371        0.000001 0.973841
   FAST      TW 0.000076 -5.179247        0.000010 0.590295
   DSGX    NDSN 0.000219 -4.932011        0.000030 1.524431
   CIGI    LAMR 0.000263 -4.888253        0.000034 1.476464
   DKNG    PLTR 0.000423 -4.771711        0.000062 1.013670
   RVMD      ZG 0.000900 -4.580678        0.000142 0.414047
   CIGI     STX 0.001303 -4.483590        0.000208 1.012979
   CHKP    ODFL 0.001392 -4.466101        0.000229 0.161200
   NVMI    SAIA 0.001749 -4.404747        0.000292 0.936042
   SSNC    ZBRA 0.002000 -4.368135        0.000339 0.378485
   ALNY    AMKR 0.002450 -4.312261        0.000424 0.694167
   CHKP    DSGX 0.002915 -4.263693        0.000521 0.172710
   MTSI    UFPI 0.003035 -4.252282        0.000539 1.146911
   CHKP    JBHT 0.003188 -4.238373        0.000571 0.250152
   CORT    SAIA 0.003269 -4.231299        0.000586 0.404718

=== ICD_XI005_MCS005: Cointegration test results ===
Passed (EG<0.1 & ADF<0.05): 356 / 2000
```

MEDOID_XI005_MCS005.

```
=== Top 15 by EG p-value - MEDOID_XI005_MCS005 ===
TickerA TickerB  EG_pval  EG_tstat  ADF_resid_pval     beta
   FAST      TW 0.000076 -5.179247        0.000010 0.590295
   DSGX    NDSN 0.000219 -4.932011        0.000030 1.524431
   CIGI    LAMR 0.000263 -4.888253        0.000034 1.476464
   CHKP    COKE 0.000367 -4.807009        0.000053 0.158352
   CHKP    MANH 0.000392 -4.790910        0.000057 0.140464
   NXST    PAYX 0.000589 -4.689048        0.000089 1.389318
   ADSK    NICE 0.000652 -4.663345        0.000099 0.780758
   CHKP    RMBS 0.000669 -4.656716        0.000102 0.123887
   CHKP    IESC 0.000679 -4.652930        0.000104 0.146544
   CHKP    ENSG 0.000726 -4.635930        0.000112 0.137905
   ALNY    JBHT 0.001076 -4.534056        0.000171 1.257243
   CIGI     STX 0.001303 -4.483590        0.000208 1.012979
   CHKP    FTAI 0.001347 -4.474693        0.000219 0.203240
   CHKP    ODFL 0.001392 -4.466101        0.000229 0.161200
   CHKP    ERIE 0.001747 -4.405026        0.000292 0.238113
```

```
=== MEDOID_XI005_MCS005: Cointegration test results ===
Passed (EG<0.1 & ADF<0.05): 287 / 2000
```

ICD_XI010_MCS003.

```
=== Top 15 by EG p-value - ICD_XI010_MCS003 ===
TickerA TickerB  EG_pval  EG_tstat  ADF_resid_pval      beta
  LBRDA   LBRDK 0.000008 -5.679371        0.000001  0.973841
   DSGX    NDSN 0.000219 -4.932011        0.000030  1.524431
   CIGI    LAMR 0.000263 -4.888253        0.000034  1.476464
   PCTY    RGEN 0.000295 -4.860152        0.000042  0.871276
   INSM       Z 0.000468 -4.746928        0.000069  0.401309
   ENTG    MRVL 0.000514 -4.723488        0.000076  1.148226
   ICLR    MORN 0.000618 -4.676934        0.000093  0.762265
   BMRN    PEGA 0.000678 -4.653471        0.000103 -0.091496
    EXE    QRVO 0.001067 -4.536364        0.000171 -0.778305
   DDOG      ZS 0.001088 -4.531224        0.000174  0.775667
   BMRN    TROW 0.001235 -4.497823        0.000200 -0.109020
   CIGI     STX 0.001303 -4.483590        0.000208  1.012979
   BMRN    TRMB 0.001505 -4.445098        0.000248 -0.086643
   BMRN    LOPE 0.001674 -4.416567        0.000279  0.108036
   NVMI    SAIA 0.001749 -4.404747        0.000292  0.936042

=== ICD_XI010_MCS003: Cointegration test results ===
Passed (EG<0.1 & ADF<0.05): 380 / 2000
```

MEDOID_XI010_MCS003.

```
=== Top 15 by EG p-value - MEDOID_XI010_MCS003 ===
TickerA TickerB  EG_pval  EG_tstat  ADF_resid_pval      beta
   CIGI    LAMR 0.000263 -4.888253        0.000034  1.476464
   NXST    PAYX 0.000589 -4.689048        0.000089  1.389318
   ADSK    NICE 0.000652 -4.663345        0.000099  0.780758
    EXE    SWKS 0.000943 -4.568617        0.000149 -0.845145
    EXE    QRVO 0.001067 -4.536364        0.000171 -0.778305
   CIGI     STX 0.001303 -4.483590        0.000208  1.012979
     EA     EXE 0.001825 -4.393191        0.000309 -0.159889
   CASY    MANH 0.001833 -4.391994        0.000308  0.531354
   CIGI    JBHT 0.001878 -4.385360        0.000312  1.095151
   AAON    HOLX 0.002611 -4.294516        0.000456  1.034245
   CYBR    ROAD 0.003473 -4.214060        0.000627  0.594079
   IDXX    POOL 0.004493 -4.139756        0.000836  0.921293
     EA    STEP 0.005204 -4.096601        0.000991  0.178723
    EXE    OLED 0.005218 -4.095819        0.000997 -0.863402
   LFUS    NWSA 0.005549 -4.077607        0.001058  0.720807

=== MEDOID_XI010_MCS003: Cointegration test results ===
Passed (EG<0.1 & ADF<0.05): 239 / 2000
```

Many top entries are economically intuitive (e.g. share classes `LBRDA` / `LBRDK` with $\beta \approx 1$), while others reflect tighter cross-industry links or vendor/customer relations. Universes with $\xi = 0.10$ and smaller MCS (`XI010_MCS003`) admit more passing pairs (380/2000 in ICD), consistent with broader coverage. MEDOID selections are more

conservative (e.g. 239/2000 in `MEDOID_XI010_MCS003`), reflecting their robustness to non-spherical geometry and tendency to exclude borderline names. Crucially, the EG layer is a *filter* rather than a trigger: pairs that look cointegrated supply a spread with reasonable mean–reversion, while the copula layer determines when a deviation is statistically rare enough to justify entry in real time.

3.6 Marginals, Copulas, and Rare-Event Gating

3.6.1 Marginal selection and Probability Integral Transform

To enable honest out-of-sample evaluation, each pair's time series is split into a 70% training segment and a 30% test segment (no shuffling or leakage). All modeling choices for marginals and copulas are learned on the training window; the trading is then evaluated on the held-out test window. On the training window $[t_0, t_1]$ we model the one–day log–returns Δy_t^A and Δy_t^B for each selected stock with four competing univariate distributions. Parameters are estimated by maximum likelihood on the training sample, and the preferred specification is chosen by information criteria. The resulting fitted cdfs, F_A and F_B, are then used to compute the *Probability Integral Transform* (PIT),

$$u_t \;=\; F_A(\Delta y_t^A), \qquad v_t \;=\; F_B(\Delta y_t^B),$$

which are approximately $\mathsf{Uniform}(0,1)$ under correct marginal specification.

Normal (Gaussian)

The Gaussian distribution is a classical baseline due to its tractability and symmetry. For $X \sim \mathcal{N}(\mu, \sigma^2)$ the density is

$$f(x \mid \mu, \sigma) \;=\; \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right].$$

It is symmetric around μ, has kurtosis 3 ("thin" tails), and often under-represents the frequency of large moves in financial returns.

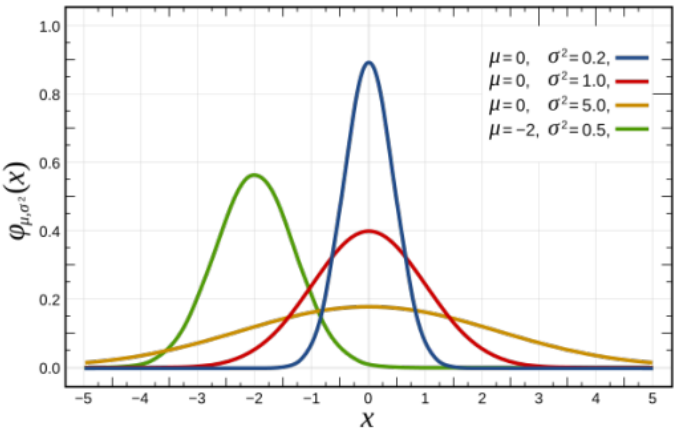


Figure 3.4. Normal distribution

Student's t

To accommodate heavy tails, we fit a centered Student's-t with $\nu > 2$ degrees of freedom; for $X \sim t_\nu$,

$$f(x \mid \nu) \;=\; \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}.$$

Smaller ν implies fatter tails and higher kurtosis. While still symmetric, the t-law typically provides a markedly better fit to return tails than the Gaussian.

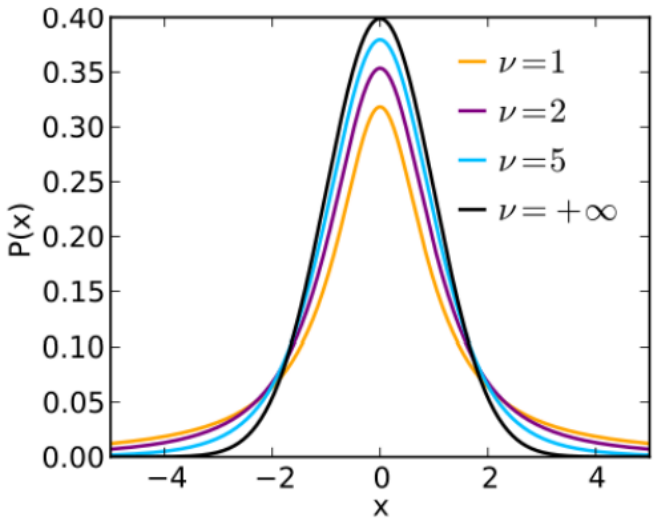


Figure 3.5. Student's t distribution

Logistic

The logistic law preserves a bell shape while allowing somewhat heavier tails than the Gaussian. With location $\mu \in \mathbb{R}$ and scale $s > 0$,

$$f(x \mid \mu, s) \;=\; \frac{e^{-(x-\mu)/s}}{s\left(1 + e^{-(x-\mu)/s}\right)^2}, \qquad x \in \mathbb{R}.$$

Here μ locates the distribution and s scales its dispersion; the tails are moderately heavier than Normal, improving the description of large (but not extreme) moves.

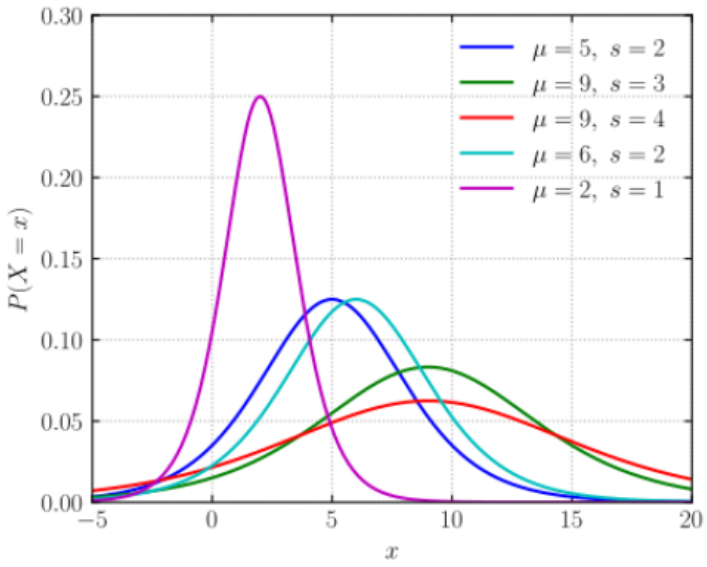


Figure 3.6. Logistic distribution

Generalized Extreme Value (GEV)

To flexibly capture asymmetric tail behavior, we also consider the GEV family from extreme–value theory. For $X \sim \mathrm{GEV}(\mu, \sigma, \xi)$ with location $\mu \in \mathbb{R}$, scale $\sigma > 0$, and shape $\xi \in \mathbb{R}$, the density is

$$f(x \mid \mu, \sigma, \xi) \;=\; \frac{1}{\sigma}\Big(1 + \xi\,\tfrac{x-\mu}{\sigma}\Big)^{-1/\xi-1} \exp\Big\{-\Big(1 + \xi\,\tfrac{x-\mu}{\sigma}\Big)^{-1/\xi}\Big\},$$

defined on the support $\{x : 1 + \xi(x-\mu)/\sigma > 0\}$. The shape parameter governs tail thickness and skewness: $\xi = 0$ corresponds to the Gumbel type (limit of light tails), $\xi > 0$ to Fréchet (heavy right tail), and $\xi < 0$ to Weibull (bounded right tail). Although developed for extremes, in practice GEV can parsimoniously accommodate return asymmetries.

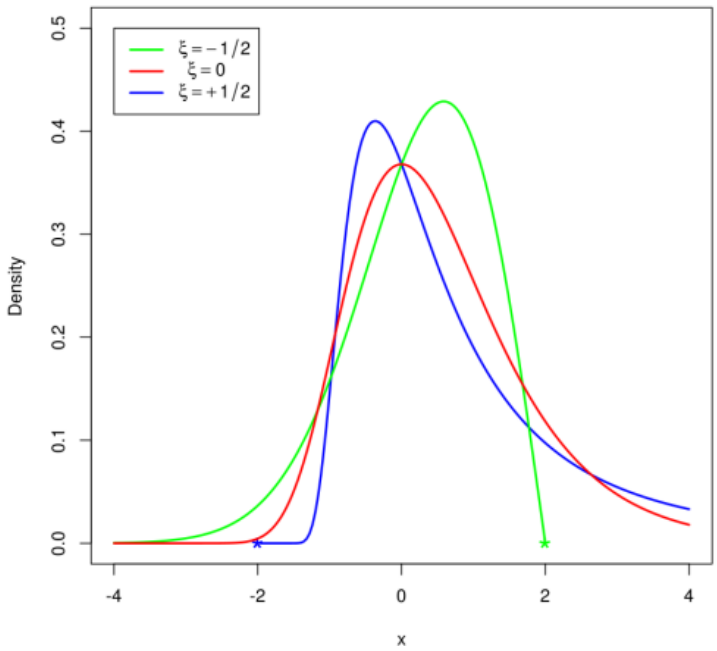


Figure 3.7. GED distribution

Model selection via AIC/BIC

Let $\hat{\theta}$ denote the maximum–likelihood estimate for a candidate marginal with k free parameters, and let $\ell(\hat{\theta})$ be the maximized log–likelihood on n observations from $[t_0, t_1]$. We rank the four candidates by:

$$\text{AIC} \;=\; 2k \;-\; 2\,\ell(\hat{\theta}), \qquad \text{BIC} \;=\; k\log n \;-\; 2\,\ell(\hat{\theta}).$$

Both penalize complexity while rewarding in–sample fit; BIC imposes a stronger penalty for additional parameters as n grows. In our implementation AIC is the default selector (BIC is reported for robustness), and the winning marginals F_A, F_B feed the PIT step used downstream by the copula layer.

Code output — (top 10 rows).

```
=== Preview: marginals_summary.csv (top 10) ===
          Source TickerA TickerB BestByBIC_A_Family BestByBIC_B_Family
ICD_XI005_MCS005   LBRDA   LBRDK          Student-t          Student-t
ICD_XI005_MCS005    FAST      TW          Student-t          Student-t
ICD_XI005_MCS005    DSGX    NDSN          Student-t          Student-t
ICD_XI005_MCS005    CIGI    LAMR          Student-t          Student-t
ICD_XI005_MCS005    DKNG    PLTR          Logistic           Student-t
ICD_XI005_MCS005    RVMD      ZG          Logistic           Student-t
ICD_XI005_MCS005    CIGI     STX          Student-t          Student-t
ICD_XI005_MCS005    CHKP    ODFL          Student-t          Student-t
ICD_XI005_MCS005    NVMI    SAIA          Student-t          Student-t
ICD_XI005_MCS005    SSNC    ZBRA          Student-t          Student-t
```

In this preview the Student's-t distribution is overwhelmingly preferred by both AIC and BIC for *both* legs, signalling heavy-tailed return behavior. A few short-history cases (e.g., `DKNG-PLTR`, `RVMD-ZG`) admit a logistic fit on one marginal under BIC, but Student's-t remains dominant overall.

3.6.2 Copula fitting and training objective

On the pseudo-observations (u_t, v_t) we fit a bivariate copula C_θ from the Gaussian, Student-t, Frank, Clayton, and Gumbel families. Parameters (correlation and degrees-of-freedom for elliptical copulas; scalar θ for Archimedean) are estimated by maximum likelihood on the training window; the best family is selected by AIC/BIC. Let $c_\theta(u, v)$ denote the copula density and

$$\ell_t \equiv \log c_\theta(u_t, v_t)$$

the per-period log-likelihood. Denote by μ_ℓ and σ_ℓ its mean and standard deviation on $[t_0, t_1]$.

Code output — (AIC selection)

```
=== Preview: copula_results_best.csv ===
          Source TickerA TickerB   Copula     Theta       LogLik
ICD_XI005_MCS005   LBRDA   LBRDK  Student-t    2.964156   2219.019688
ICD_XI005_MCS005    FAST      TW  Student-t    4.997897     51.741680
ICD_XI005_MCS005    DSGX    NDSN  Student-t    4.351313    101.965858
ICD_XI005_MCS005    CIGI    LAMR  Student-t    3.859953    117.471260
ICD_XI005_MCS005    DKNG    PLTR     Frank     4.329316    116.677661
```

The AIC-driven selection yields predominantly heavy–tailed *Student-t* copulas—consistent with the non-Gaussian co-movement often observed in equities—plus one *Frank* copula capturing monotone dependence. Here `Theta` denotes the family-specific scalar parameter, while `LogLik` is the maximized log-likelihood for the fitted copula; less-negative (or positive) values indicate a comparatively better fit on the same data window.

3.6.3 Rare-event gate

After estimating, on the training window, the best-fitting univariate marginals F_A and F_B and the copula $C_{\hat\theta}$ for each selected pair (A, B), the test-window returns are mapped to uniforms via the Probability Integral Transform (PIT):

$$u_t \;=\; F_A(\Delta y_t^A), \qquad v_t \;=\; F_B(\Delta y_t^B),$$

with values clipped to $[\varepsilon,\, 1-\varepsilon]$ for numerical stability. The joint dependence at time t is then evaluated through the copula *log-density*

$$\ell_t \;=\; \log c\!\left(u_t, v_t \,|\, \hat\theta\right),$$

where $c(\cdot,\cdot \,|\, \hat\theta)$ is the density associated with $C_{\hat\theta}$.

Definition of the gate

Let μ_ℓ and σ_ℓ denote, respectively, the mean and standard deviation of $\{\ell_t\}$ computed on the training sample of the pair. In the test window, a *rare-event* flag is raised whenever

$$\ell_t \;\le\; \mu_\ell - q\,\sigma_\ell,$$

with $q = \texttt{RARE\_Q\_SIGMA} = 1.0$ in our baseline runs. Only when this gate is open do we allow spread signals to trigger entries.

Operational view: the copula as a gate

Intuitively, the copula plays the role of a *gatekeeper* on the entry logic. The spread signal (e.g., a large $|Z_t|$) proposes a trade, but the copula checks whether the *joint* return realization $(\Delta y_t^A, \Delta y_t^B)$ is sufficiently unusual *under the trained dependence*. Formally, we use the indicator

$$G_t = 1_{\{\ell_t \le \mu_\ell - q\sigma_\ell\}}.$$

and we enter only if

$$G_t = 1 \quad \text{and} \quad |Z_t| \ge Z_{\text{enter}}.$$

In this sense, the copula-based gate opens only on *rare joint events*: it lets through spread deviations that are more likely to reflect a transient dislocation (mean reversion) rather than a persistent regime change.

Why it helps (decoupling marginals and dependence)

The statistic ℓ_t scores how plausible the *joint* outcome is under the dependence fitted in-sample, after factorizing away the marginals via the PIT. Requiring ℓ_t to be q standard deviations below its training mean acts as a low-probability filter on the *dependence* layer. This reduces false positives where $|Z_t|$ is large but the two assets move in a way still typical for the fitted copula (i.e., not a true dislocation).

Link to a confidence level

If ℓ_t is approximately Gaussian, the q–sigma rule corresponds to a one-sided tail-confidence $c \approx \Phi(-q)$. For $q = 1$, $c \approx 16\%$: we only accept spread signals occurring with joint likelihood in roughly the worst 16% of training scenarios. This is conceptually akin to fixing a high confidence (e.g., 84–90%) for the gate, implemented via a simple, scale-free z-score on ℓ_t.

Because the gate operates on the copula log-density, it naturally adapts to different forms of tail dependence (e.g., lower-tail for Clayton, upper-tail for Gumbel, symmetric/heavy-tail for Student-t). This makes the gate sensitive to those rare co-movements that the chosen copula emphasizes, rather than to arbitrary large moves in either marginal.

Interaction with the spread signal and CUSUM

Direction and size of the trade come from the normalized spread z_t of the OLS-hedged pair; an entry is opened only if *both* conditions hold: (i) the rare-event gate is open and (ii) $|z_t| \ge \texttt{Z\_ENTER}$. Exits follow the spread rules (take-profit / stop-loss on $|z_t|$), while in CUSUM-enabled portfolios an additional regime-break

filter can prevent entries or force exits upon structural shifts. In this way, the copula layer governs *when* we act (timing/selection), and the spread governs *how* we act (direction and sizing).

3.7 Spread Construction, Signal Logic, and Execution

3.7.1 Hedge ratio and standardized spread

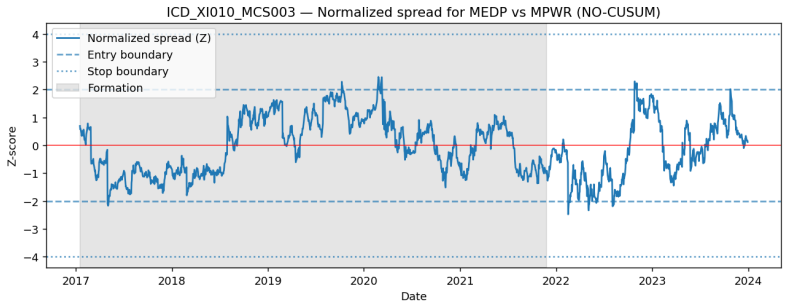
On $[t_0, t_1]$ we estimate the hedge ratio by regressing y_t^A on y_t^B with intercept:

$$y_t^A = \alpha + \beta\, y_t^B + \varepsilon_t, \qquad \hat{\beta} = \arg\min_{\beta} \sum_{t=t_0}^{t_1} (y_t^A - \alpha - \beta y_t^B)^2.$$

The spread is $s_t = y_t^A - \hat{\beta}\, y_t^B$. With training mean μ_s and standard deviation σ_s, the standardized spread ("z-score") is

$$z_t = \frac{s_t - \mu_s}{\sigma_s}.$$

We use thresholds `Z_ENTER` $= 2.0$, `Z_EXIT` $= 0.5$, and `Z_STOP` $= 4.0$.



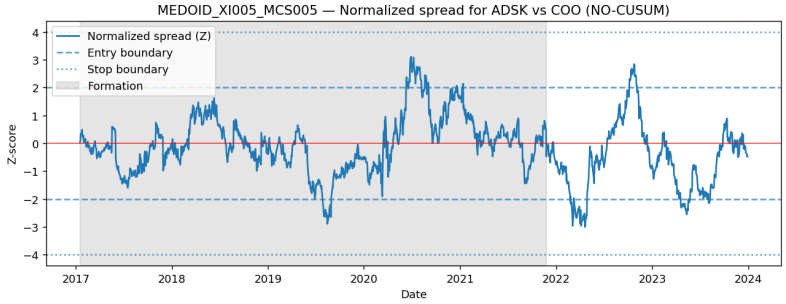
3.7.2 Entry/exit rules

On the test window:

- **Entry**: if the rare-event gate is open and $|z_t| \geq 2.0$. Go long the spread when $z_t \leq -2.0$ (buy A, sell $\hat{\beta}$ units of B); go short when $z_t \geq 2.0$.
- **Exit**: on mean reversion ($|z_t| \leq 0.5$) or emergency stop ($|z_t| \geq 4.0$). In CUSUM-enabled runs we also exit on regime-break detection.

The trade PnL ignoring costs for a position of sign ± 1 is

$$\text{PnL} = \pm\,(P^A_{\text{exit}} - P^A_{\text{entry}}) \mp \hat{\beta}\,(P^B_{\text{exit}} - P^B_{\text{entry}}).$$



3.8 CUSUM Regime-Shift Detection

3.8.1 Definition and parameters

To avoid trading through structural breaks in dependence, we apply a CUSUM filter to centered copula log-likelihoods $x_t = \ell_t - \mu_\ell$:

$$g_t^+ = \max\{0,\ g_{t-1}^+ + (x_t - k)\}, \qquad g_t^- = \max\{0,\ g_{t-1}^- - (x_t + k)\},$$

with $g_0^+ = g_0^- = 0$, drift parameter $k = \texttt{CUSUM\_k} = 0.10$, and threshold $h = \texttt{CUSUM\_h} = 5.0$. If either statistic exceeds h, we flag a regime shift. In **CUSUM** portfolios we block new entries while the filter is tripped and force an exit if a position is open. **NO-CUSUM** portfolios disable this layer.

3.8.2 Empirical effect of the CUSUM gate

In the *test* window we evaluate the CUSUM filter on the same pool of entry candidates (i.e., times t with the rare–event flag open and $|z_t| \geq 2$). The following console-style summaries report (i) how many potential entries are *blocked* by CUSUM and (ii) how many trades are subsequently *exited* by a CUSUM break.

Code output

```
CUSUM filtering of potential entries (TEST window) - per Source:
                    PotentialEntries  BlockedByCUSUM  BlockedRate
Source
ICD_XI005_MCS005                2716             921     0.339000
ICD_XI010_MCS003                3074             987     0.321000
MEDOID_XI005_MCS005             2248             676     0.301000
MEDOID_XI010_MCS003             1665             504     0.303000

Exits triggered by CUSUM (count of trades) - per Source:
Source
ICD_XI005_MCS005       112
ICD_XI010_MCS003       124
MEDOID_XI005_MCS005     84
MEDOID_XI010_MCS003     68
```

Interpretation

- **Blocking rate.** Aggregating across sources, CUSUM screened out 3,088 of 9,703 potential entries, i.e. $\approx 31.8\%$ overall. Rates are slightly higher for ICD

sources (33.9% and 32.1%) and around 30% for MEDOID sources, consistent with ICD clusters being more exposed to density shifts in this dataset.

- **Forced exits.** The filter also triggered 388 break exits in total, split as 236 for ICD ($112 + 124$) and 152 for MEDOID ($84 + 68$)—the same order reported elsewhere—confirming that CUSUM actively cuts positions when the spread behavior becomes inconsistent with the trained regime.

- **Net effect.** Empirically, CUSUM removes roughly one third of otherwise admissible entries and enforces a moderate number of protective exits. This is precisely the intended role of the module: a conservative guard against persistent dependence changes, trading fewer signals while shaping drawdowns.

3.9 Portfolio-level performance and risk (CUSUM vs. NO-CUSUM)

Trade-level distribution metrics

Before turning to equity-curve diagnostics, we examine the cross-section of *per-trade* PnLs (in currency units) generated by each portfolio. The following statistics are reported and interpreted:

- **Mean & Median.** Central tendency of trade PnLs; the median is robust to outliers and indicates the "typical" trade, while the mean captures the contribution of tail winners/losers.

- **Std. Dev.** Dispersion of trade outcomes; larger values imply a wider spread between winners and losers, i.e., higher per-trade risk.

- **Minimum/Maximum.** The most adverse and most favorable single-trade outcomes observed; these give a tangible sense of downside (tail loss) and upside potential.

- **Skewness.** Shape asymmetry of the trade distribution. Positive skew means occasional large winners that pull the mean above the median; negative skew would indicate fatter left tails.

- **t-stat of the mean.** Statistical significance of the average trade PnL, computed as $\bar{x}/(s/\sqrt{n})$ over the sample of trade PnLs; larger values indicate the mean is reliably different from zero.

Code output

ICD_XI005_MCS005 - CUSUM vs. NO-CUSUM

Metric	CUSUM	NO-CUSUM
Information Ratio	0.321	0.354
Maximum	250.353	345.435
Mean	8.040	11.880
Median	1.471	0.645
Minimum	-92.425	-94.560
Skewness	2.871	2.995
Std. Dev.	29.489	37.022
t-stat	5.719	7.827

ICD_XI010_MCS003 - CUSUM vs. NO-CUSUM

Metric	CUSUM	NO-CUSUM
Information Ratio	0.386	0.395
Maximum	130.664	157.665
Mean	6.592	12.170
Median	0.961	2.729
Minimum	-83.829	-182.429
Skewness	1.191	0.947
Std. Dev.	22.733	33.398
t-stat	6.788	9.474

MEDOID_XI005_MCS005 - CUSUM vs. NO-CUSUM

Metric	CUSUM	NO-CUSUM
Information Ratio	0.448	0.519
Maximum	130.664	149.237
Mean	9.702	14.870
Median	3.134	5.902
Minimum	-76.710	-75.639
Skewness	0.973	1.182
Std. Dev.	26.419	32.425
t-stat	7.102	9.793

MEDOID_XI010_MCS003 - CUSUM vs. NO-CUSUM

Metric	CUSUM	NO-CUSUM
Information Ratio	0.355	0.503
Maximum	130.664	149.237
Mean	9.618	17.364
Median	2.109	7.637
Minimum	-76.710	-75.639
Skewness	1.071	1.165
Std. Dev.	26.024	34.146
t-stat	6.272	9.404

Reading the trade-level metrics

Across all four sources, **NO-CUSUM** portfolios exhibit a *higher Mean* per trade than their CUSUM counterparts (e.g., 11.88 vs. 8.04 for `ICD_XI005_MCS005`; 17.36 vs. 9.62 for `MEDOID_XI010_MCS003`), but also a *larger Std. Dev.* (e.g., 37.02 vs. 29.49 for `ICD_XI005_MCS005`; 34.15 vs. 26.02 for `MEDOID_XI010_MCS003`). *Maximum* gains are consistently higher without CUSUM (e.g., 345.44 vs. 250.35 for `ICD_XI005_MCS005`; 149.24 vs. 130.66 for `MEDOID_XI010_MCS003`). The most adverse trade (*Minimum*) is much worse for `ICD_XI010_MCS003` without CUSUM (-182.43 vs. -83.83) and slightly worse for `ICD_XI005_MCS005` (-94.56 vs. -92.43), while it is broadly comparable or marginally *better* in the MEDOID runs (e.g., -75.64 vs. -76.71). Medians are generally larger under NO-CUSUM in three out of four sources (e.g., 2.73 vs. 0.96 for `ICD_XI010_MCS003`; 7.64 vs. 2.11 for `MEDOID_XI010_MCS003`); the exception is `ICD_XI005_MCS005`, where CUSUM yields a higher median (1.47 vs. 0.65). *Skewness* is positive in all cases—profit distributions remain right-tailed—with similar magnitudes (NO-CUSUM slightly higher except for `ICD_XI010_MCS003`). Finally, the mean's *t-stat* is higher without CUSUM across all sources (e.g., 7.83 vs. 5.72 for `ICD_XI005_MCS005`; 9.40 vs. 6.27 for `MEDOID_XI010_MCS003`), and the *Information Ratio* also improves under NO-CUSUM (ranging from 0.35–0.52 vs. 0.32–0.45 with CUSUM).

3.10 Results overview

What we compare

We benchmark *eight* portfolios that differ only in how pairs are *discovered* and *guarded* before trading. They are obtained by crossing two OPTICS configurations with two representative–selection rules, and then running each in two execution modes:

- **OPTICS configurations:**
  - `XI005_MCS005`: extraction steepness $\xi = 0.05$ (sharper split of valleys in the reachability plot) and minimum cluster size MCS = 5% of the sample (more regularization, admits only sufficiently large clusters).
  - `XI010_MCS003`: extraction steepness $\xi = 0.10$ (smoother, merges shallow valleys) and MCS = 3% (allows smaller, local structures).

  Diagnostics recorded for each run: the number of clusters discovered, the count (and share) of points labeled as *noise* by OPTICS, and the *silhouette score* computed on the OPTICS labels (higher is better separation). Intuitively,

ξ controls how aggressively we cut the reachability landscape, while MCS prevents over-fragmentation by requiring a minimum support for each cluster.

- **Cluster representatives:**
  - `ICD` (*in-cluster distance to centroid*): pairs are built around the cluster centroid; representatives are the assets closest (in feature space) to that centroid.
  - `MEDOID`: pairs are anchored to the *medoid*, i.e., the actual observation minimizing within–cluster dissimilarity.
- **Execution modes:**
  - **CUSUM**: a regime–break filter is active. It halts new entries and may force exits when a CUSUM statistic indicates a persistent shift (conservative mode).
  - **NO–CUSUM**: the regime filter is disabled; entries rely only on the copula rare–event gate and spread z-score thresholds (opportunistic mode).

The four base portfolio IDs are therefore `ICD_XI005_MCS005`, `ICD_XI010_MCS003`, `MEDOID_XI005_MCS005`, and `MEDOID_XI010_MCS003`; each is evaluated twice, with **CUSUM** and **NO–CUSUM**, for a total of eight portfolios.

Metrics considered in this section

We report one risk/return diagnostic from the *daily net equity* series (i.e., after costs):

- **Maximum drawdown.** For the cumulative net equity process E_t, define the running peak $P_t = \max_{u \le t} E_u$. The (fractional) drawdown at time t is $D_t = 1 - E_t/P_t$, and the *maximum drawdown* is

$$\mathrm{MDD} \;=\; \max_t D_t \;=\; \max_t \left(1 - \frac{E_t}{\max_{u \le t} E_u}\right).$$

  It measures the worst peak-to-trough loss experienced over the sample; smaller (less negative in currency units) indicates better downside risk protection.

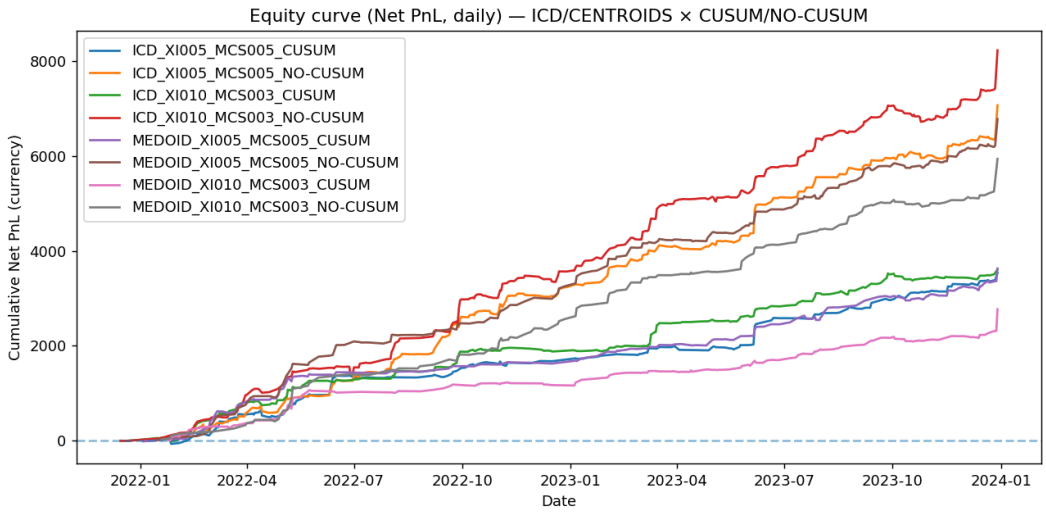
Cost model and aggregation

To compare portfolios on realistic grounds, we apply an ex-post cost model:

$$\text{Cost} = (\texttt{SLIPPAGE\_BPS}+\texttt{FEES\_BPS}) \times 10^{-4} \times \left(|P^A_{\text{entry}}|+|P^A_{\text{exit}}|+|\hat{\beta}|(|P^B_{\text{entry}}|+|P^B_{\text{exit}}|)\right),$$

with `SLIPPAGE_BPS`= 2.0 and `FEES_BPS`= 0.5. Net PnL equals Gross PnL minus Cost. Net PnL is aggregated by exit date to obtain daily equity.

Portfolio	Max Drawdown	Total Gross Return %	Total Net Return %
ICD_XI005_MCS005_CUSUM	-146.912	4.778	3.794
ICD_XI005_MCS005_NO-CUSUM	-143.079	5.710	4.852
ICD_XI010_MCS003_CUSUM	-151.089	4.224	3.207
ICD_XI010_MCS003_NO-CUSUM	-344.144	5.744	4.888
MEDOID_XI005_MCS005_CUSUM	-93.298	4.839	3.977
MEDOID_XI005_MCS005_NO-CUSUM	-96.366	6.127	5.365
MEDOID_XI010_MCS003_CUSUM	-98.181	4.151	3.404
MEDOID_XI010_MCS003_NO-CUSUM	-143.246	6.205	5.528



Reading the results

Three broad patterns emerge:

1. **Return boost without the regime filter.** In all four bases, **NO–CUSUM** attains the higher *Total Net Return %* (about 4.85–5.53%) versus **CUSUM** (about 3.21–3.98%). For example, `ICD_XI010_MCS003` delivers 4.89% without CUSUM vs. 3.21% with CUSUM, and `MEDOID_XI005_MCS005` yields 5.37% vs. 3.98%.

2. **Risk shaping with CUSUM.** Max drawdowns are *tighter with CUSUM in 3 out of 4* sources: e.g., `ICD_XI010_MCS003` −151.09 vs. −344.14 without CUSUM; `MEDOID_XI005_MCS005` −93.30 vs. −96.37; `MEDOID_XI010_MCS003` −98.18 vs. −143.25. The exception is `ICD_XI005_MCS005`, where NO–CUSUM is slightly shallower (−143.08 vs. −146.91).

3. **Drawdown patterns vary across portfolios.** The most conservative drawdown is delivered by `MEDOID_XI005_MCS005_CUSUM` (≈ −93.30). The most severe drawdown occurs for `ICD_XI010_MCS003_NO-CUSUM` (≈ −344.14).

These summaries capture the core trade-off: **NO–CUSUM** yields higher net returns at the expense of deeper drawdowns, whereas **CUSUM** sacrifices some return to stabilize downside risk.

Chapter 4

Conclusions

Aim and scope: This thesis set out to design, implement, and evaluate a *copula–gated mean-reversion* pipeline for equity pairs trading, where *pair discovery* is informed by both market data and firm descriptors (including ESG scores), and *execution* is disciplined by a rare-event likelihood gate and an optional CUSUM regime filter. The empirical universe consisted of the *largest NASDAQ constituents by market capitalization*: from an initial long-list of 400 names we retained **343 stocks** for which *all* the data blocks were available—daily prices, ESG scores, and annual balance-sheet statements—over the period **FY2017–FY2023**.

Statistical screening and modeling: Pairs were vetted for linear equilibrium using the Engle–Granger two-step logic: an OLS hedge ratio on the training window, followed by residual stationarity (ADF) with the EG p-value as the primary ranking device. For execution, we separated marginals from dependence: each leg's returns were fit to candidate distributions (Normal, Student-t, Logistic, GEV) with model choice by AIC (BIC supported); the Probability Integral Transform mapped returns to uniforms, on which we estimated several bivariate copulas (Gaussian, Student-t, and standard Archimedeans), again selected by AIC.

Trade logic and risk controls: *Entries* require a large normalized spread (a z-score threshold) *and* a *rare-event gate* based on copula log-likelihood excursions (threshold $q = 1.0$ standard deviations below the training mean). *Exits* follow spread mean-reversion rules with stop-loss/take-profit safeguards. In the **CUSUM** variants we activate a cumulative-sum detector that (i) blocks entries when a dependence shift is suspected and (ii) can force earlier exits; the **NO-CUSUM** variants trade without this regime-break filter.

Eight portfolios and cost-aware evaluation: Each base construction (`ICD_XI005_MCS005`, `ICD_XI010_MCS003`, `MEDOID_XI005_MCS005`, `MEDOID_XI010_MCS003`) is run in both **CUSUM** and **NO-CUSUM** mode, for a total of **eight portfolios**. We evaluate performance on daily *net* equity, after applying a transparent ex-post cost model (slippage and fees in basis points on both legs, at entry and exit). The main diagnostics are *Total Net Return %* and *Max Drawdown (net)*.

Headline findings: Across the four base constructions, **NO-CUSUM** portfolios realize *higher total net returns* by monetizing more signals: representative net percentages are about $4.85\% - 5.53\%$ (vs. $3.21\% - 3.98\%$ for CUSUM). **CUSUM** variants *consistently shape risk*: they trade fewer opportunities, experience *shallower drawdowns*, and show more stable equity profiles (e.g., large drawdown gaps in favor of CUSUM under `ICD_XI010_MCS003`).

Implications for investors: CUSUM vs. NO-CUSUM. A central message of the study is the *allocative* one. **NO-CUSUM** variants tend to earn *more* in percentage terms because they admit more entries and let spreads resolve without a regime governor; this comes with *deeper* and sometimes more frequent drawdowns, reflecting the classic higher-risk, higher-return trade-off observed in asset pricing. **CUSUM** variants deliberately give up some trades and occasionally truncate winners, yet they *materially lower tail risk* and stabilize the equity path.

Hence the choice is preference-driven:

an investor prioritizing absolute return might favor NO-CUSUM; an investor valuing *capital preservation*, smoother paths, and smaller worst-case losses may rationally prefer CUSUM. In other words, the regime filter acts as a *risk dial*: turning it off increases expected return together with downside amplitude; turning it on reduces drawdowns while keeping risk-adjusted metrics competitive. Thus, the evidence suggests both modes are viable: risk-seeking investors may gravitate toward NO-CUSUM for its higher upside, while more risk-averse investors will often favor CUSUM, accepting slightly lower returns in exchange for stronger risk mitigation and steadier performance

Bibliography

[1] EconStor. *Monitoring Breaks in Fractional Cointegration* (PDF). https://www.econstor.eu/bitstream/10419/307742/1/1912365251.pdf.

[2] V. Sfragara. *Un test CUSUM per modelli a memoria lunga* (Thesis, Università di Padova). https://thesis.unipd.it/retrieve/260ce016-d046-4468-a1ec-a35fb2627429/Sfragara_Valeria.pdf.

[3] Hudson & Thames. *Homepage — Research & Education in Quant Finance.* https://hudsonthames.org/.

[4] A. Rai. *The CUSUM Algorithm: All the Essential Information You Need (with Python examples).* Stackademic. https://blog.stackademic.com/the-cusum-algorithm-all-the-essential-information-you-need-with-python-examples-f6a5651bf2e5.

[5] Wikipedia. *Copula (statistics).* https://en.wikipedia.org/wiki/Copula_(statistics).

[6] GeeksforGeeks. *Elbow Method vs. Silhouette Score — Which is Better?* https://www.geeksforgeeks.org/machine-learning/elbow-method-vs-silhouette-score-which-is-better/.

[7] Wikipedia. *K-means clustering.* https://en.wikipedia.org/wiki/K-means_clustering.

[8] Wikipedia. *Cluster analysis.* https://en.wikipedia.org/wiki/Cluster_analysis.

[9] IBM Think. *Clustering (topic page).* https://www.ibm.com/think/topics/clustering.

[10] Hudson & Thames. *Definitive Guide to Pairs Trading.* https://hudsonthames.org/definitive-guide-to-pairs-trading/.

[11] Wikipedia. *OPTICS algorithm.* https://en.wikipedia.org/wiki/OPTICS_algorithm.

[12] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander. *OPTICS: Ordering Points To Identify the Clustering Structure.* ACM SIGMOD '99. https://dl.acm.org/doi/10.1145/304181.304187.

[13] `scikit-learn`. *sklearn.cluster.OPTICS — API reference.* https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html.

[14] AlgoTrading101 Blog. *Cluster Analysis — Machine Learning for Pairs Trading*. (Blog post).

[15] `scikit-learn`. *A demo of K-Means clustering on the handwritten digits data*. https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html.

[16] Adam (Call For Atlas). *Unsupervised Learning as Signals for Pairs Trading and StatArb*. Medium. (Article).

[17] A. Henrique. *Clustering with K-Means: simple yet powerful*. Medium. (Article).

[18] M.R. Mota. *Pairs Trading: Using Machine Learning for the Selection of Pairs*. Medium / AI Monks. (Article).

[19] Hudson & Thames. *Machine Learning for Trading Pairs Selection*. (Article).

[20] S.S. Galiani. *Introduction to Copulas* (Colab notebook). https://colab.research.google.com/drive/1cHObO4OYyMlHLd09aOZzYZibeK4O8-lP.

[21] W. Xie, Y. Wu. *Copula-based pairs trading strategy*. Asian Finance Association (AsFA) Conference, 2013.

[22] Y. Stander, D. Marais, I. Botha. *Trading strategies with copulas. Journal of Economic and Financial Sciences*.

[23] Investopedia. *Pairs Trade*. https://www.investopedia.com/terms/p/pairstrade.asp.

[24] T. Palomar. *Pairs Trading* (lecture slides, HKUST MAFS5310). https://palomar.home.ece.ust.hk/MAFS5310_lectures/slides_pairs_trading.pdf.

[25] T. Zhu. *Pairs Trading*. Yale University, Department of Economics (notes). https://economics.yale.edu/sites/default/files/2024-05/Zhu_Pairs_Trading.pdf.

[26] H. Cho. *Change-point detection in panel data via double CUSUM statistic*. arXiv:1611.08631. https://arxiv.org/pdf/1611.08631.