

Project 3: Simulating Pure vs. Dirty Dispersion Trading

Quantitative Research Assignment

November 12, 2025

Objective

The aim of this project is to simulate and analyse a dispersion trading strategy under both *pure* (theoretical) and *dirty* (realistic) conditions. You will build a full model linking stock-level volatilities, correlations, and option-implied surfaces, and evaluate how trading frictions affect profitability. The assignment should demonstrate a clear understanding of volatility decomposition, correlation exposure, and P&L attribution under realistic market constraints.

Background

Dispersion trading profits from the discrepancy between the volatility of an index and the average volatility of its components. For an index composed of N assets with average volatility $\bar{\sigma}$ and average pairwise correlation ρ , we have:

$$\sigma_{\text{index}}^2 = \bar{\sigma}^2 \left(\rho + \frac{1 - \rho}{N} \right)$$

and for large N :

$$\rho \approx \frac{\sigma_{\text{index}}^2}{\bar{\sigma}^2}.$$

A *pure* dispersion trade involves being short index volatility and long single-name volatilities in vega-neutral amounts. Profit arises when realised correlation ρ_{real} is less than implied correlation ρ_{imp} . However, in real markets (“dirty” dispersion), frictions such as execution costs, liquidity differences, and funding drag reduce or even eliminate this theoretical edge.

Simulation Design

1. Generate Synthetic Stock Returns

- Simulate $N = 50$ stocks with daily log-returns generated from a multivariate normal distribution with pairwise correlation $\rho = 0.20$.
- Individual annualised volatilities $\sigma_i \in [25\%, 35\%]$.
- Generate 1,000 trading days of data.

The covariance matrix is given by:

$$\text{Cov}(r_i, r_j) = \begin{cases} \sigma_i^2, & i = j, \\ \rho \sigma_i \sigma_j, & i \neq j. \end{cases}$$

2. Simulate Index Returns and Realised Volatility

The index return is the equally weighted mean:

$$r_{\text{index},t} = \frac{1}{N} \sum_{i=1}^N r_{i,t}.$$

Compute realised volatilities for both stocks and index using rolling 20-day standard deviation of log-returns.

3. Compute Implied Correlation and Theoretical Edge

Assume market-implied volatilities:

$$\sigma_{\text{index,imp}} = 12\%, \quad \bar{\sigma}_{\text{stocks,imp}} = 28\%.$$

The implied correlation is:

$$\rho_{\text{imp}} = \frac{\sigma_{\text{index,imp}}^2}{\bar{\sigma}_{\text{stocks,imp}}^2}.$$

The theoretical correlation “edge” is:

$$\Delta\rho = \rho_{\text{imp}} - \rho_{\text{real}},$$

and the variance edge is:

$$\Delta\text{Var} = \sigma_{\text{index,imp}}^2 - \bar{\sigma}_{\text{stocks,imp}}^2 \rho_{\text{real}}.$$

Pure Dispersion Simulation

1. Simulate several paths of correlation decay, where $\rho_{\text{real}} \in [0.05, 0.25]$.
2. Compute expected dispersion P&L as a function of $\Delta\rho$:

$$P\&L_{\text{pure}} \propto -\bar{\sigma}^2 \Delta\rho.$$

3. Visualise:

- Line chart: $\Delta\rho$ vs expected P&L.
- Table of correlation edge under varying N and ρ_{imp} .

Dirty Dispersion Simulation

To introduce real-world frictions, apply the following components derived from the attached briefing:

Friction Type	Description	Simulation Approximation
Execution Drag	Bid-ask crossing on 50 options	Random 0.5–2.0 bps per name
Funding Drag	Carry cost on long vega positions	25–50 bps per month
Vega Mismatch	Index vs single-name skew curvature	Add random noise $\sim \mathcal{N}(0, 0.02)$ to ρ_{real}
Liquidity Asymmetry	Uneven single-name liquidity	Inflate cost on tail names by 2–3×

The *dirty* realised P&L is then:

$$P\&L_{\text{dirty}} = P\&L_{\text{pure}} - C_{\text{exec}} - C_{\text{fund}} - C_{\text{liq}}.$$

Scenario Analysis

Perform a 1,000-path Monte Carlo simulation varying:

- $\rho_{\text{real}} \in [0.05, 0.25]$,
- Transaction costs between 0.5 and 5 bps,
- Number of constituents $N \in \{10, 25, 50, 100\}$.

Estimate the *expected net P&L distribution* and compute the break-even friction level:

$$C_{\text{break-even}}^* = \text{friction level where } \mathbb{E}[P\&L] = 0.$$

P&L Attribution

Summarise contributions in the following table (mirroring the structure from the briefing):

Component	Clean (Pure)	Dirty (Realised)
Correlation P&L	Positive (if $\rho_{\text{real}} < \rho_{\text{imp}}$)	Partially retained
Execution Drag	None	-300 bps
Funding Drag	None	-50 bps (6 months)
Vega/Gamma Mismatch	None	Random \pm impact
Total P&L	+X%	+Y% after frictions

Mathematical Deliverables

1. Show that the derivative of P&L with respect to correlation is:

$$\frac{\partial P\&L}{\partial \rho} = -\bar{\sigma}^2.$$

2. Derive how correlation exposure scales with N :

$$\text{Exposure} \propto \frac{N-1}{N}.$$

3. Quantify the sensitivity of dispersion to *volatility of volatility* via small random perturbations to σ_i .

Visualisations

1. Heatmap of P&L vs $(\rho_{\text{real}}, \text{cost})$.
2. Stacked bar chart: Clean vs Dirty P&L decomposition.
3. Histogram: Distribution of net P&L from Monte Carlo.
4. Line chart: Cumulative simulated dispersion P&L.
5. Correlation term-structure curve (replicating Figure 1 of the briefing).

Expected Insights

- Dispersion profit is driven primarily by the gap between implied and realised correlation.
- Execution and funding drag erode the theoretical edge quickly.
- Larger baskets reduce idiosyncratic noise but increase cost sensitivity.
- The idealised (“clean”) model overstates achievable returns in practice.

Recommended Project Structure

```
projects/26_dirty_dispersion/  
  data/  
    synthetic_dispersion.csv  
  notebooks/  
    01_dirty_dispersion_analysis.ipynb  
  src/  
    simulate_dispersion.py  
    compute_pnl.py
```

```

visualize_results.py
reports/
  figures/
    corr_term_structure.png
    pnl_distribution.png
    clean_vs_dirtyBars.png
requirements.txt

```

Evaluation Criteria

Aspect	Weight	Expectation
Simulation accuracy	25%	Realistic correlation and volatility dynamics
Clarity of pure vs dirty modelling	25%	Equations and logic clearly separated
Visualisation	20%	Insightful, clearly labelled figures
Quantitative reasoning	20%	Strong link between model and economics
Code structure	10%	Modular, readable Python or notebook design

Extensions (Optional Second Day)

- Introduce regime-dependent correlation dynamics (e.g. stress vs calm regimes).
- Fit rolling realised correlation from the simulated data.
- Add a transaction-cost optimiser for adaptive sizing.
- Replicate the slope of the implied correlation term structure using synthetic vol surfaces.

Conclusion

This assignment consolidates applied understanding of volatility, correlation, and real-world trading frictions. It bridges theoretical dispersion pricing with the messy reality of execution, funding, and surface asymmetries, mirroring the challenges faced by sell-side and hedge fund volatility desks.