Hierarchical Risk Parity Explained from First Principles

A friendly guide for readers with a high school math background

Version: October 27, 2025

Contents

1 Why Study Risk Allocation?

Imagine sharing chores among a group of friends. If everyone does everything, some tasks get repeated while others are ignored. Investing works the same way. Instead of chores, we split up "risk" so that no single asset shocks the whole portfolio. Hierarchical Risk Parity (HRP) is a modern method for sharing risk fairly, especially when we only have noisy data.

1.1 From Everyday Intuition to Finance

Consider three friends biking to school. If they all ride on the same narrow street, a single traffic jam delays everyone. Diversifying their routes reduces the chance that all are late simultaneously. HRP searches for independent "routes" in the market, aiming to keep the group (the portfolio) on time (stable).

1.2 What Makes HRP Different?

Classical portfolio theory, pioneered by Harry Markowitz, directly uses the inverse of a covariance matrix. That matrix captures how each pair of assets co-moves. Unfortunately, when we do not have much data relative to the number of assets, inverting the matrix magnifies estimation errors, the same way dividing by a tiny number blasts small errors into huge ones. HRP manages to avoid inversion altogether. It scans the natural structure of correlations, groups similar assets, and assigns weights level by level.

1.3 The Guiding Principles

HRP follows two rules that we can state without calculus:

a) **Only equalize things people actually feel.** The "thing" in finance is risk (volatility), not just invested dollars.

b) **Respect the buddy system.** Assets that move together form a team. HRP measures how tight each team is and ensures no single team dominates the portfolio.

Throughout this document we build each ingredient needed to implement those rules, starting from raw prices and ending with a complete algorithm.

2 Building Blocks: Prices and Returns

Before we use covariance matrices, we need to describe how price data becomes the math objects we manipulate.

2.1 Price Series

Suppose we track prices $P_{i,t}$ for asset $i = 1, \ldots, N$ over days $t = 0, \ldots, T$. Prices are positive numbers, typically expressed in dollars. High school math lets us manipulate these using ratios and differences.

2.2 Simple (Arithmetic) Returns

The simple return from day $t-1$ to t is

$$r_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}} = \frac{P_{i,t}}{P_{i,t-1}} - 1.$$

This is the percentage gain or loss over one day. For small movements (which is typical for daily data), arithmetic and logarithmic returns are almost the same. We use arithmetic returns in most of this note because they are easy to average.

2.3 Collecting Returns into a Matrix

Organize the data into a table (matrix) called R. Each row is a day, each column is an asset. Entry $R_{t,i} = r_{i,t}$. This matrix is the starting point for computing averages, variances, and covariances.

2.4 Sample Mean of Returns

The average return for asset i over T days is

$$\bar{r}_i = \frac{1}{T} \sum_{t=1}^{T} r_{i,t}.$$

This is the same as computing a grade point average. Although HRP emphasizes risk rather than return, we still need means to compute centered values.

3 Variance: Measuring Individual Noise

Variance tells us how much an asset wiggles around its mean.

3.1 Definition from First Principles

Given numbers $x_1, \dots, x_T$, their variance is

$$\mathrm{Var}(x) = \frac{1}{T-1} \sum_{t=1}^{T} (x_t - \bar{x})^2.$$

This formula reuses skills from algebra: subtract, square, and average. The $T - 1$ in the denominator is called the Bessel correction; it keeps the estimate unbiased when data are drawn from a larger population.

3.2 Interpretation

Variance grows when observations spread out. If stock A moves $+2\%, -2\%, +2\%, -2\%$ while stock B stays at 0%), then stock A has positive variance but stock B has zero variance. Standard deviation is the square root of variance and has the same units as the original data (percentage changes).

3.3 Annualizing Variance

Daily data can be scaled to yearly risk by multiplying by 252, the approximate number of trading days in a year. Specifically,

$$\Sigma_{ii} = 252 \times \widehat{\Sigma}_{ii}$$

where $\widehat{\Sigma}$ is the daily covariance matrix. This is a simple rule-of-thumb that matches the fact that variances add for independent increments.

4 Covariance and Correlation

Diversification is only meaningful when we examine how assets interact, not just how they behave alone.

4.1 Covariance Defined

For two assets i and j, the covariance is

$$\widehat{\Sigma}_{ij} = \frac{1}{T-1}\sum_{t=1}^{T}(r_{i,t} - \bar{r}_i)(r_{j,t} - \bar{r}_j).$$

This number is positive when the assets often rise and fall together, negative when they usually move in opposite directions, and near zero when they act independently.

4.2 Correlation as a Standardized Covariance

Correlation rescales covariance to lie between -1 and $+1$:

$$\rho_{ij} = \frac{\widehat{\Sigma}_{ij}}{\sqrt{\widehat{\Sigma}_{ii}}\sqrt{\widehat{\Sigma}_{jj}}}.$$

A value of $+1$ means perfect teamwork, -1 means perfect tug-of-war, and 0 means no predictable relationship. Notice the denominator is the product of the standard deviations. This makes correlation dimensionless, which is convenient for judging relationships across assets with different volatilities.

4.3 Correlation Matrix

Collect all ρ_{ij} into a matrix ρ. This matrix is symmetric ($\rho_{ij} = \rho_{ji}$). The diagonal entries are all ones. Thinking of ρ as a map of friendships helps: tightly knit groups have correlations near 1, while unfamiliar pairs hover near zero.

4.4 Example with Three Assets

Suppose we have three stocks A, B, and C. After processing returns, we estimate

$$\rho = \begin{bmatrix} 1.00 & 0.80 & 0.10 \\ 0.80 & 1.00 & 0.15 \\ 0.10 & 0.15 & 1.00 \end{bmatrix}.$$

Stocks A and B almost move together, so they should not both receive large weights if we want diversification. Stock C hardly moves with the others, so it provides diversification value.

5 Distances Built from Correlations

To apply clustering algorithms, we convert correlation into a distance that obeys geometry-like rules.

5.1 Defining the Distance

The most common mapping is

$$d_{ij} = \sqrt{\frac{1 - \rho_{ij}}{2}}.$$

This formula has several nice properties:

- If $\rho_{ij} = 1$, then $d_{ij} = 0$ (no distance between perfectly correlated assets).
- If $\rho_{ij} = -1$, then $d_{ij} = 1$ (maximum distance for perfectly opposite movers).
- The mapping ensures d_{ij} is a genuine metric, meaning it satisfies the triangle inequality.

5.2 Why Distances Matter

Clustering algorithms expect a distance matrix, not a correlation matrix. The distance matrix allows us to apply geometry-based intuition: assets that are close in distance belong to the same neighborhood.

5.3 Condensed Distance Vector

For programming, we often flatten the upper triangle of the distance matrix into a vector with $N(N-1)/2$ entries. This is the format consumed by functions such as `scipy.cluster.hierarchy.linkage`.

6 Agglomerative Hierarchical Clustering

Now we turn the distance matrix into a tree. The process mirrors merging study groups based on shared interests.

6.1 Bottom-Up Merging

1. Start with each asset as its own cluster.
2. Find the two clusters with the smallest distance between them.
3. Merge them into a new cluster and record the distance at which this happened.
4. Repeat until all assets join a single mega-cluster.

The sequence of merges can be visualized with a dendrogram (a tree diagram).

6.2 Linkage Choices

There are multiple ways to define the distance between two multi-asset clusters:

- **Single linkage:** use the smallest pairwise distance between members of the clusters.
- **Complete linkage:** use the largest pairwise distance.
- **Average linkage:** average all pairwise distances.

HRP commonly uses single linkage because it emphasizes tight relationships, ensuring that the resulting order respects the strongest dependencies. However, when markets have noisy relationships, average linkage can be more stable.

6.3 Ultrametric Property

The tree structure satisfies the ultrametric inequality: for any three leaves i, j, k, the distances obey an even stronger condition than the triangle inequality. This property justifies the recursive splitting later.

7 Quasi-Diagonalization

Once we have a dendrogram, we read off the leaf order via a depth-first traversal (walk down the left branch before the right). Reordering the covariance matrix according to this leaf order produces a blocky structure.

7.1 Permutation Matrix

Let p be the list of leaf indices encountered in the traversal. Construct the permutation matrix P by placing ones at positions $P_{i,p_i} = 1$ and zeros elsewhere. Then

$$\Sigma^* = P\Sigma P^\top$$

rearranges rows and columns simultaneously.

7.2 Interpretation

If two assets belong to the same tight cluster, they now appear next to each other, and their covariance estimates concentrate along the diagonal. Visually, the matrix looks almost block-diagonal, hence the term "quasi-diagonalization." This step requires no matrix inversion; it only shuffles entries.

7.3 Energy Analogy

Think of Σ as a heat map. Quasi-diagonalization moves the hottest zones (high covariance blocks) onto the diagonal so we can treat them group by group without cross-talk.

8 Recursive Bisection: Sharing Risk Level by Level

This is the heart of HRP. After reordering, we split the tree at its root and assign risk budgets to the left and right children.

8.1 Inverse-Variance Portfolio (IVP) Inside a Cluster

Within any cluster C, we define preliminary weights using the inverse of individual variances:

$$w_i^{\mathrm{IVP}} = \frac{1/\Sigma_{ii}^*}{\sum_{j \in C} 1/\Sigma_{jj}^*}.$$

These IVP weights down-weight volatile assets and are easy to compute because they only need diagonal entries.

8.2 Cluster Variance

We estimate the aggregate variance of cluster C by

$$\sigma^2(C) = (w^{\mathrm{IVP}})^\top \Sigma_C^* w^{\mathrm{IVP}}$$

where Σ_C^* is the submatrix of Σ^* corresponding to the members of C.

8.3 Allocating Between Siblings

If a parent cluster splits into children C_L and C_R, HRP assigns them weights

$$\alpha_L = 1 - \frac{\sigma(C_L)}{\sigma(C_L) + \sigma(C_R)}, \qquad \alpha_R = 1 - \alpha_L.$$

Intuitively, the child with larger variance receives a smaller share of the parent's weight. Then we multiply existing weights in C_L by α_L and in C_R by α_R, and recurse until every leaf is a single asset.

8.4 Algorithmic Pseudocode

1. Input: ordered covariance matrix Σ^* and tree structure.
2. Initialize all weights to 1.
3. For each split from root to leaves:
    (a) Identify left and right clusters.

   (b) Compute IVP weights within each cluster.
   (c) Obtain cluster variances.
   (d) Allocate parent weight between children using the inverse-variance rule.

4. Normalize weights so they sum to one.

9 Risk Contributions at the Leaves

After recursion we obtain final weights w. To check fairness we compute each asset's risk contribution.

9.1 Marginal Risk Contribution

The marginal contribution of asset i is

$$\mathrm{MRC}_i = \frac{(\Sigma w)_i}{\sqrt{w^\top \Sigma w}}.$$

This formula mirrors the derivative of the portfolio standard deviation with respect to w_i.

9.2 Total Risk Contribution

Multiply by the weight to get the total contribution:

$$\mathrm{TRC}_i = w_i \times \mathrm{MRC}_i.$$

Summing over all assets recovers the portfolio standard deviation. HRP aims for clusters to have similar aggregate TRCs.

9.3 Visualization

Plot TRCs by cluster to ensure no cluster dominates. In the slides repository this is implemented in `notebooks/plotting.py`. Visual checks help detect when the clustering collapsed into one giant group, which can happen in highly correlated universes.

10 Worked Mini Example

To make the procedure concrete, assume four assets with annualized covariance matrix (after quasi-diagonalization)

$$\Sigma^* = \begin{bmatrix} 0.040 & 0.032 & 0.010 & 0.008 \\ 0.032 & 0.050 & 0.009 & 0.007 \\ 0.010 & 0.009 & 0.030 & 0.020 \\ 0.008 & 0.007 & 0.020 & 0.045 \end{bmatrix}.$$

Assets 1 and 2 form the left cluster, assets 3 and 4 form the right cluster.

10.1 Cluster Variances

- Left IVP weights: proportional to $1/0.040 = 25$ and $1/0.050 = 20$, so normalized weights are $(0.556, 0.444)$.
- Left cluster variance: substitute into $w^\top \Sigma^* w$ to get 0.0436.
- Right IVP weights: proportional to $1/0.030 = 33.3$ and $1/0.045 = 22.2$, giving $(0.600, 0.400)$.
- Right cluster variance: computed similarly as 0.0350.

10.2 Allocating Between Clusters

The right cluster is less volatile, so it receives a larger share of weight:

$$\alpha_L = 1 - \frac{\sqrt{0.0436}}{\sqrt{0.0436} + \sqrt{0.0350}} \approx 0.46,$$

$$\alpha_R = 0.54.$$

Multiplying down the tree yields final weights: $w_1 = 0.256, w_2 = 0.204, w_3 = 0.324, w_4 = 0.216$.

10.3 Checking Risk Contributions

Using w and Σ^*, we compute TRCs and confirm that the left cluster contributes about 49% of total risk while the right contributes 51%, validating the HRP logic.

11 Connecting to the Repository Implementation

Section 11 of the slide deck mentions how each theoretical step appears in code. Here is a more detailed map.

11.1 Data Handling

- `data/financials prices.csv`: historical price data.
- `src/data loader.py`: converts prices to returns and applies cleaning (forward-filling, removing obvious outliers).

11.2 Covariance Estimation

- `src/risk/covariance.py`: functions for sample covariance, exponential weighting, and Ledoit-Wolf shrinkage.
- Code adheres closely to Equation (3) in the slides.

11.3 Clustering and Ordering

- `src/clustering/hrp.py`: builds the condensed distance matrix, applies SciPy linkage, and extracts the depth-first order.
- The permutation matrix is implemented implicitly by reorganizing index arrays rather than forming P explicitly.

11.4 Recursive Bisection

- `src/risk/allocation.py`: contains the recursive allocation function. It computes IVP weights, cluster variances, and weight splits exactly as in Equations (7)-(8).
- Numerical stability tricks: clipping extremely small variances and adding 10^{-8} to diagonals to avoid division by zero.

11.5 Reporting

- `notebooks/hrp walkthrough.ipynb`: demonstrates the pipeline with plots for dendrograms, TRCs, and performance metrics.
- CSV outputs in the `outputs/` folder correspond to the summary tables in the slides.

12 Statistical Considerations in Plain Language

Advanced slide sections discuss shrinkage, Bayesian estimators, and statistical tests. Here we restate their meaning for a broader audience.

12.1 Why Shrinkage Helps

Sample covariance matrices can be unstable when T is not much larger than N. Shrinkage pulls extreme estimates toward a structured target, such as a diagonal matrix. Think of it as blending your noisy measurement with a conservative guess to reduce overfitting.

12.2 Ledoit-Wolf Shrinkage

The Ledoit-Wolf procedure automatically picks the blend weight by estimating how noisy the sample covariance is. It balances bias and variance so that the final matrix is a better predictor out of sample.

12.3 Bayesian Covariance Estimation

Bayesian approaches (like the Bayes-Stein estimator) treat the covariance matrix as a random object with a prior distribution. Observed data updates this prior. The result is similar to shrinkage but framed probabilistically.

12.4 Bootstrap Sharpe Tests

When comparing strategies, we cannot rely solely on observed differences; randomness might explain the gap. Bootstrap methods resample return paths to build a distribution of Sharpe differences. If zero lies comfortably within the confidence interval, we cannot claim a real performance gap.

13 Common Pitfalls and Practical Checks

Even with a clear algorithm, real-world implementation demands discipline.

13.1 Pitfall 1: Misaligned Data

Ensure all return series share the same calendar. Missing dates create artificial spikes in covariance. Aligning timestamps and forward-filling short gaps avoids this.

13.2 Pitfall 2: Over-Shrinkage

Shrinking too aggressively toward a diagonal target erases meaningful correlation structure. Monitor how shrinkage affects cluster shapes; if the dendrogram collapses to equal distances, ease the shrinkage intensity.

13.3 Pitfall 3: Ignoring Transaction Costs

HRP tends to trade less than minimum variance but more than equal weighting. Always compute turnover (average absolute change in weights) and set a rebalance schedule compatible with trading costs.

13.4 Diagnostic Checklist

1. Plot the dendrogram and verify that clusters align with economic intuition (e.g., banks grouped with banks).
2. Compare HRP weights to inverse-variance weights to detect extreme skews.
3. Re-run the process under small perturbations (e.g., add noise to returns) to see if weights remain stable.

14 Frequently Asked Questions

14.1 Do I Need Calculus?

No. HRP uses algebra, square roots, averages, and basic linear algebra (matrices). High school skills suffice.

14.2 Can HRP Lose Money?

Yes. HRP only redistributes risk; it does not forecast returns. If the entire market drops, HRP loses as well, although usually with less drawdown than naive allocations.

14.3 How Many Assets Are Needed?

HRP can operate with as few as five assets, but clustering becomes more meaningful when there are distinct groups (10+ assets).

14.4 Does HRP Handle Constraints?

Baseline HRP assumes long-only weights that sum to one. Extensions can incorporate leverage, minimum/maximum weights, or turnover penalties. These are discussed in Section 12 of the slides and in the repository's optimization modules.

15 Extended Example: Building HRP in a Spreadsheet

For readers without coding experience, it is helpful to simulate HRP in a spreadsheet.

15.1 Step-by-Step Spreadsheet Plan

1. Input daily prices for four assets in columns B–E.
2. Compute daily returns in rows by dividing each price by the previous day's price minus one.
3. Use built-in functions to compute means, variances, and covariances (e.g., `COVARIANCE.P`).
4. Convert covariance to correlation (`CORREL`).
5. Apply the distance formula manually for each pair.
6. Perform hierarchical clustering using add-ins or by manually ranking distances.
7. Reorder the covariance matrix by the cluster order and run the recursive allocation formulas.

15.2 Educational Value

This exercise reinforces that HRP is not a black box; each calculation is transparent and reproducible with basic tools.

16 Practical Extensions

16.1 Constraint Handling

Real portfolios often impose maximum weight limits. A common extension caps any weight at, say, 10% and redistributes excess proportionally among the remaining assets while preserving the cluster hierarchy.

16.2 Robust Variants

When covariances are unstable, one can replace the sample covariance with a shrinkage estimator or a factor model covariance. The HRP steps remain identical; only the input matrix changes.

16.3 Machine Learning Perspectives

Some researchers interpret HRP as a form of unsupervised learning on the correlation matrix. The dendrogram resembles a decision tree that splits assets according to similarity. This viewpoint connects HRP to clustering methods used in image and signal processing.

17 Glossary

Asset A tradable item like a stock or bond.

Return Percentage change in price from one period to the next.

Variance Average squared deviation from the mean; measures volatility.

Covariance Measures how two assets move together.

Correlation Covariance scaled to lie between -1 and $+1$.

Distance Matrix Symmetric matrix describing pairwise dissimilarities.

Dendrogram Tree diagram showing how clusters merge.

Quasi-Diagonalization Reordering a matrix so that related items sit together.

Recursive Bisection Splitting a set into two parts repeatedly.

Risk Contribution Portion of total portfolio risk attributable to one asset.

18 Practice Problems

18.1 Problem Set

1. Given three assets with pairwise correlations 0.9, 0.5, and 0.2, build the distance matrix and determine which pair the clustering algorithm merges first.
2. Show that if two assets have identical return series, HRP will treat them as one effective asset. Compute the resulting weights.
3. Create a simple HRP allocation for assets with variances 0.04, 0.02, 0.01 and mutual correlations of 0.3. Compare the results to equal weighting.

18.2 Solutions Sketch

Detailed walkthroughs are provided in Appendix A (not shown here) and match the logic introduced earlier. Solving these reinforces the mechanical steps while keeping the arithmetic manageable.

19 Summary and Next Steps

We started with raw prices, defined returns, built covariance and correlation matrices, converted correlations into distances, performed hierarchical clustering, and finally executed the recursive allocation that defines HRP. Each step relied on fundamental algebraic manipulations rather than advanced calculus. With practice, the sequence becomes intuitive:

1. Clean data.
2. Estimate correlations.
3. Cluster and reorder.
4. Allocate risk top-down.
5. Validate with risk contribution charts and performance tests.

Future explorations could involve stress testing HRP under crisis scenarios, mixing it with factor models, or integrating expected returns to tilt weights while preserving risk budgets.