Grouping Investors

- To perform KMeans clustering on the dataset you provided, w
first need to organize and preprocess the data. Assuming you
working with a typical setup in Python, you'd generally use
libraries like Pandas for data manipulation and scikit-learn for
applying the KMeans algorithm.

- Python code example that imports the data, preprocesses it, and applies KMeans clustering:
- **Explanation**:
- **Elbow Method**: Helps determine the optimal number of clusters plotting the within-cluster sum of squares (inertia) against the nur of clusters.
- **KMeans Clustering**: Segments investors into three clusters bas on the features provided.
- **Analysis**: By examining the mean values of each cluster, you ca identify characteristics such as average income, investment size, age, and risk tolerance, which can help tailor products or service
- This process can be adapted based on real data and actual inve features to generate actionable insights for financial services firm

Hierarchical Risk Parity (HRP)

- Hierarchical Risk Parity (HRP) is an innovative portfolio construction methodology introduced by Marcos Lopez de Prado. It seeks to overcome some of the issues found in traditional risk parity approaches, like the Markowitz Mean-Variance Optimization, which can be highly sensitive t estimation errors in the input data (expected returns, covariances). HRP uses a hierarchical clustering method to structure the assets into a dendrogram and then allocates capital based on this hierarchical structure, ensuring diversification across different clusters of assets.

- **Implementation Steps:**

1. **Correlation and Distance Calculation**: Calculate the correlation matrix of asset returns and then convert these correlations into distances.
2. **Hierarchical Clustering**: Use the distance matrix to perform hierarchical clustering on assets.
3. **Quasi-Diagonalization of Covariance Matrix**: Order the covariance matrix according to the hierarchical clustering.
4. **Recursive Bisection**: Allocate weights recursively, beginnin with the split between two broadest clusters and then within each sub-cluster.

Implement this with Python using a hypothetical set of assets:

You'll need **numpy**, **pandas**, **scipy**, and **matplotlib**. Ensure these are installed with: