Big Data in Finance

Week 5: Unsupervised Learning

Dr Daniele Bianchi

Spring 2026

Today's Agenda

From Supervised to Unsupervised Learning

Principal Component Analysis (PCA)

Clustering Methods

Recap: PCA and Clustering

PCA for Statistical Factor Models

Clustering for Portfolio Construction

Regime Detection with Clustering

PCA for Dimensionality Reduction in Prediction

Summary and Next Steps

From Supervised to Unsupervised Learning

Recap: What We've Learned So Far

Supervised learning: We had a target variable Y to predict

- **Regression** (Weeks 2–3): Predict returns, continuous Y
  - Ridge, Lasso, Random Forest, Boosting
- **Classification** (Week 4): Predict default, binary Y
  - Logistic regression, classification trees, ensembles

Common Structure

Given (X_i, Y_i) pairs, learn $\widehat{f}$ such that $\widehat{Y} = \widehat{f}(X)$ predicts Y well.

What Is Unsupervised Learning?

Key difference: There is **no target variable** Y

We only observe features $X_1, X_2, \ldots, X_p$ for each observation. The goal is to discover **structure** in the data.

Two main tasks:

1. **Dimensionality reduction:** Find a smaller set of variables that captures most of the information
   - Principal Component Analysis (PCA)
2. **Clustering:** Group similar observations together
   - K-means, hierarchical clustering

Today's Focus

PCA for dimensionality reduction; clustering for segmentation

Why Does This Matter in Finance?

Dimensionality reduction (PCA):

- *Factor models:* Extract common factors from many asset returns
- *Feature reduction:* 100+ characteristics $\rightarrow$ few principal components
- *Noise reduction:* Remove idiosyncratic variation
- *Multicollinearity:* Predictors often highly correlated

Clustering:

- *Portfolio construction:* Group similar stocks
- *Regime detection:* Identify market states (bull/bear/crisis)
- *Client segmentation:* Risk profiles, investment preferences
- *Peer analysis:* Find comparable firms

Key Insight

Unsupervised methods help us understand **structure** before (or instead of) prediction

A Motivating Example: Stock Returns

Setting: We observe monthly returns for 500 stocks over 10 years

- Data matrix: 500×120 (stocks $\times$ months)
- Each row is a stock's return time series

Observation: Stock returns are highly correlated!

- When the market goes up, most stocks go up
- Tech stocks move together, banks move together, etc.

Questions we can ask:

1. What are the common factors? (PCA)
2. Can we group stocks into similar "clusters"? (Clustering)

Principal Component Analysis (PCA)

The Core Idea of PCA

Goal: Find new variables ("principal components") that:

1. Are **linear combinations** of the original variables
2. Capture as much **variance** as possible
3. Are **uncorrelated** with each other

Why variance?

- Variance = information (variables that don't vary carry no information)
- PCA finds directions of maximum spread in the data

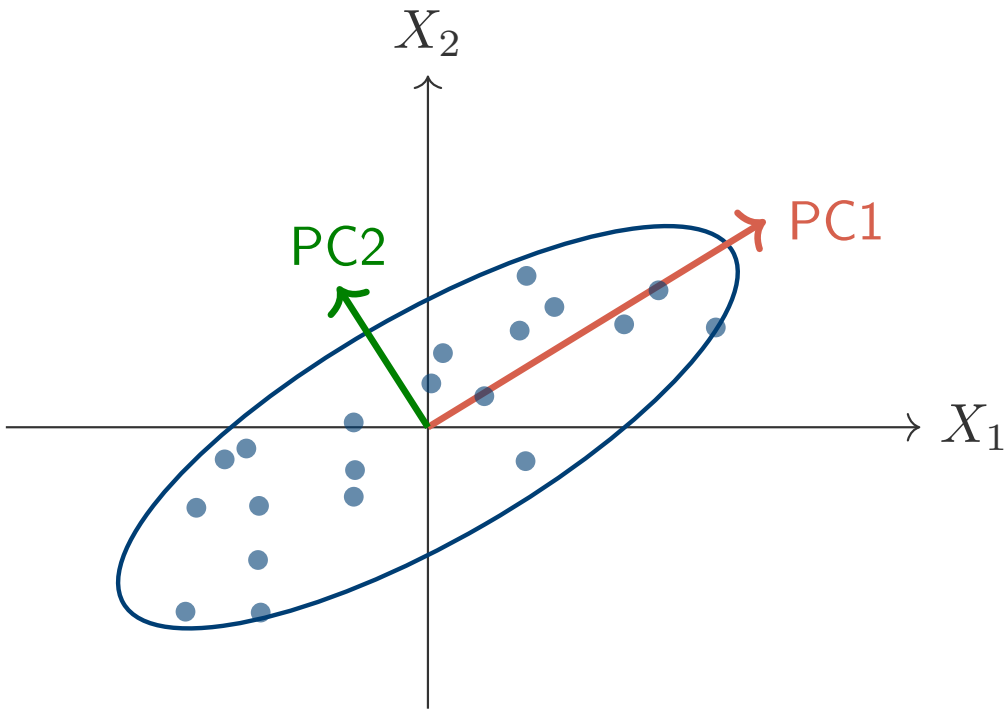
Intuition: If we can only keep $k < p$ dimensions, which ones preserve the most information about the original data?

Key Result

PCA gives us the *optimal* linear dimensionality reduction in terms of preserved variance

Geometric Intuition: 2D Example

Imagine data in 2D: Points scattered in an ellipse



- **PC1:** Direction of maximum variance (major axis)
- **PC2:** Orthogonal direction, captures remaining variance
- If we project onto PC1 only, we lose the least information

The Mathematics of PCA

Setup: Data matrix $\mathbf{X}$ is $n \times p$ (observations $\times$ variables)

Assume variables are **centered** (mean zero).

First principal component:

- Find loadings $\mathbf{w}_1 = (w_{11}, w_{12}, \ldots, w_{1p})^\top$ with $\|\mathbf{w}_1\| = 1$
- Such that the new variable $Z_1 = \mathbf{X}\mathbf{w}_1$ has **maximum variance**

$$\mathbf{w}_1 = \arg\max_{\|\mathbf{w}\|=1} \mathrm{Var}(Z_1) = \arg\max_{\|\mathbf{w}\|=1} \mathbf{w}^\top \mathbf{S} \mathbf{w}$$

where $\mathbf{S} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}$ is the sample covariance matrix.

Second principal component: Same, but require $Z_2 \perp Z_1$ (uncorrelated with first)

And so on for PC3, PC4, ...

How Does PCA Find the Components?

The algorithm: PCA analyzes the covariance matrix of your data

What you need to know:

1. PCA finds the **directions** (principal components) where data varies most
2. Each direction comes with a **number** (eigenvalue) measuring its importance
3. Directions are ranked: PC1 captures most variance, PC2 second-most, etc.
4. Directions are **perpendicular** to each other (uncorrelated)

The outputs:

- **Loadings:** How much each original variable contributes to each PC
- **Eigenvalues:** How much variance each PC captures
- **Scores:** The new transformed data points

Good News

Python handles all the math — you just need to interpret the results!

How Much Information Do We Keep?

Question: If we reduce from p variables to k PCs, how much do we lose?

Think of it this way:

- Total variance = total "information" in your data
- Each PC captures some fraction of this information
- PC1 captures the most, PC2 second-most, and so on

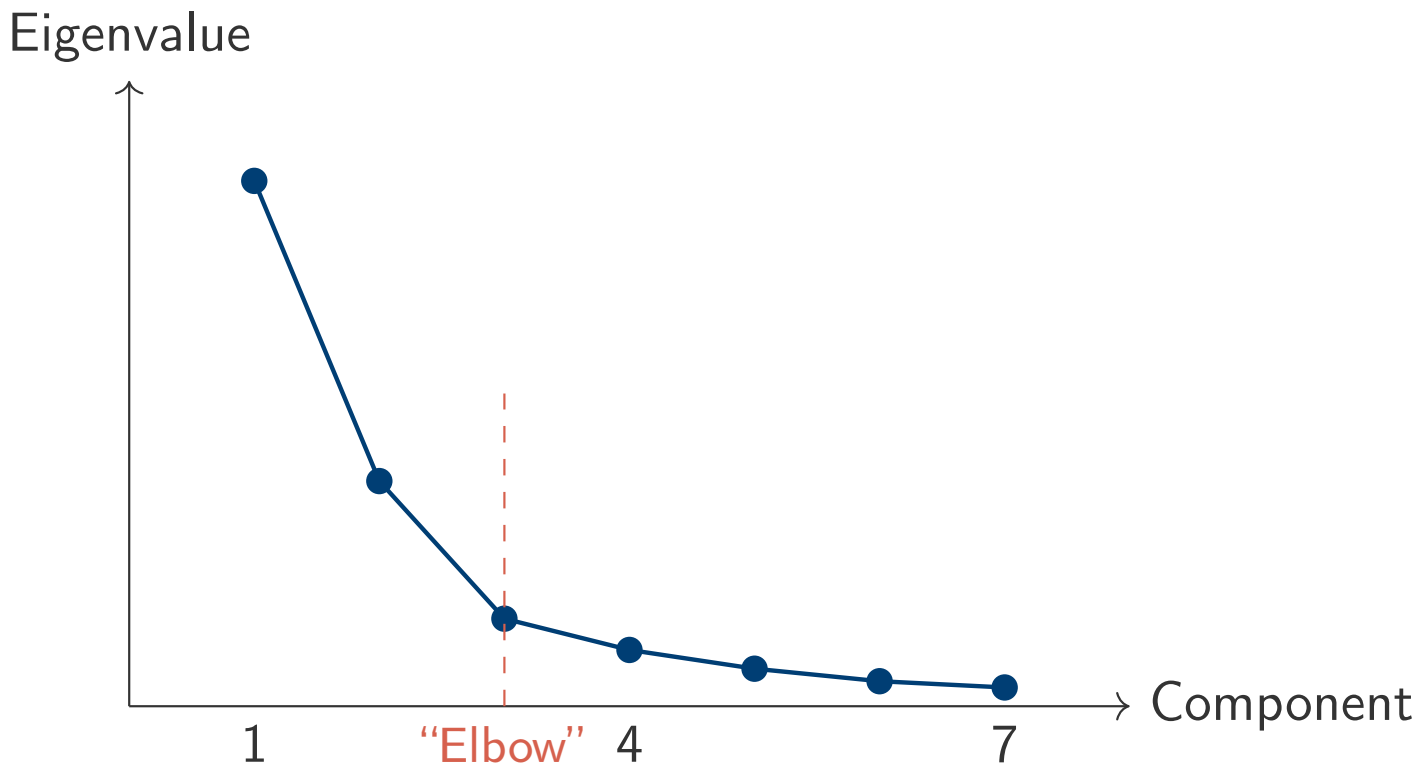
Example: 10 stock returns, applying PCA

	Variance Explained	**Cumulative**
PC1	45%	45%
PC2	20%	65%
PC3	10%	75%
PCs 4–10	25%	100%

Interpretation: Using just 3 PCs instead of 10 variables, we retain 75% of the information. The last 7 PCs add little — mostly noise.

The Scree Plot: Choosing Number of Components

How many PCs should we keep?



Common approaches:

1. *Elbow method:* Look for "elbow" where curve flattens
2. *Cumulative variance:* Keep enough for 80–90% CPVE
3. *Cross-validation:* If using PCs for prediction, tune k via CV

Standardization: An Important Choice

Question: Should we standardize variables before PCA?

Without standardization:

- Use covariance matrix $\mathbf{S}$
- Variables with larger variance dominate
- Appropriate when scale is homogeneous (e.g., all returns in %)

With standardization:

- Use correlation matrix $\mathbf{R}$ (standardize each variable to variance 1)
- All variables treated equally regardless of scale
- Appropriate when scales differ (e.g., mixed financial ratios)

Rule of Thumb

If variables are in different units or have very different variances, **standardize first**. If all variables are comparable (e.g., stock returns), covariance may be preferred.

Interpreting Principal Components

The loadings (eigenvector entries) tell us what each PC represents

Example: PCA on stock returns

Stock	PC1 Loading	PC2 Loading	PC3 Loading
Apple	0.15	0.35	-0.10
Microsoft	0.14	0.32	-0.12
JPMorgan	0.12	-0.25	0.40
Goldman Sachs	0.11	-0.28	0.38
ExxonMobil	0.10	-0.05	-0.30
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Interpretation:

- **PC1:** All positive loadings $\approx$ "market factor"
- **PC2:** Tech positive, banks negative $\approx$ "growth vs. value"
- **PC3:** Banks positive, energy negative $\approx$ "sector rotation"

PCA as a Factor Model

Connection to finance: PCA extracts **statistical factors**

Factor model representation:

$$R_{it} = \alpha_i + \beta_{i1} F_{1t} + \beta_{i2} F_{2t} + \cdots + \beta_{ik} F_{kt} + \varepsilon_{it}$$

where:

- F_{jt}: Factor j at time t (the principal component scores)
- β_{ij}: Loading of asset i on Factor j
- ε_{it}: Idiosyncratic return

Key insight: PCA on *returns* gives us the factors that best explain return *covariances*

Statistical vs. Economic Factors

PCA factors are **statistical** (maximize variance explained), not necessarily **economic** (like Fama-French). We'll compare them in the next lecture.

Limitations of PCA

PCA is powerful, but has limitations:

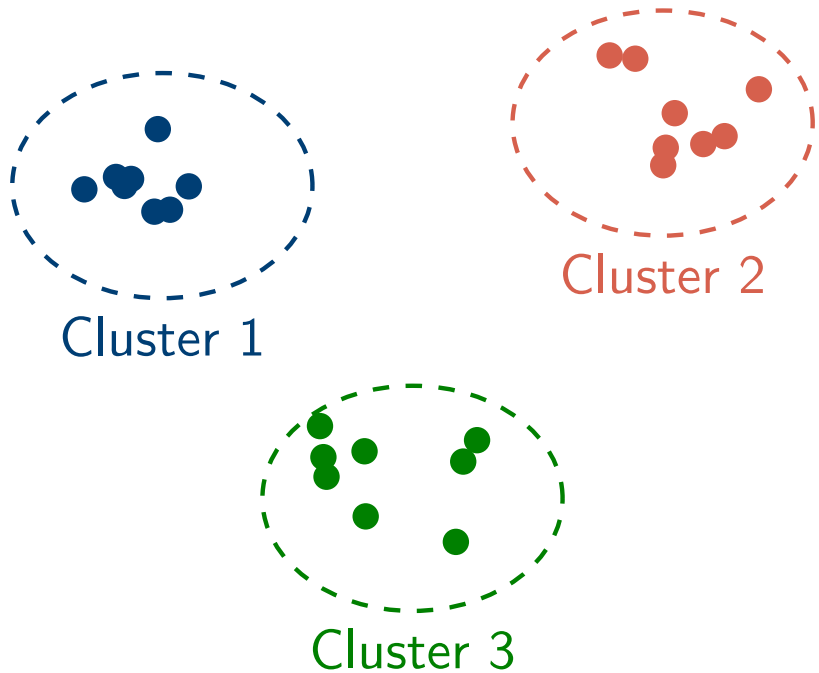
1. **Linearity:** Only finds linear combinations
   - Kernel PCA can capture non-linear structure
2. **Orthogonality:** Components must be uncorrelated
   - Real factors may be correlated (e.g., value and momentum)
3. **Interpretability:** PCs are linear combinations of *all* variables
   - Sparse PCA forces some loadings to zero
4. **Sample dependence:** Factors are estimated from data
   - Can be unstable with small samples or noisy data

Clustering Methods

From Dimensionality Reduction to Clustering

PCA: Reduce the number of *variables*

Clustering: Group *observations* into similar clusters



Goal: Find groups where observations within a cluster are similar, and observations in different clusters are dissimilar.

Applications of Clustering in Finance

1. **Portfolio construction:**
   - Group stocks by return behavior (not just sector labels)
   - Diversify by picking from different clusters
2. **Regime detection:**
   - Cluster time periods by market conditions
   - Identify bull markets, bear markets, crisis periods
3. **Client segmentation:**
   - Group investors by risk tolerance, behavior
   - Tailor products and advice
4. **Peer group analysis:**
   - Find comparable companies for valuation
   - Data-driven alternative to SIC codes

K-Means Clustering: The Idea

Goal: Partition n observations into K clusters

Objective: Minimize within-cluster variation

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} W(C_k)$$

where $W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|x_i - x_j\|^2$ is the within-cluster sum of squares.

Equivalently:

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

where $\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$ is the centroid of cluster k.

Intuition

Find K centroids and assign each point to nearest centroid

K-Means Algorithm

The algorithm:

1. **Initialize:** Randomly assign each observation to one of K clusters
2. **Repeat until convergence:**
   (a) **Update centroids:** Compute the mean of each cluster

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

   (b) **Reassign:** Assign each observation to the cluster with the closest centroid

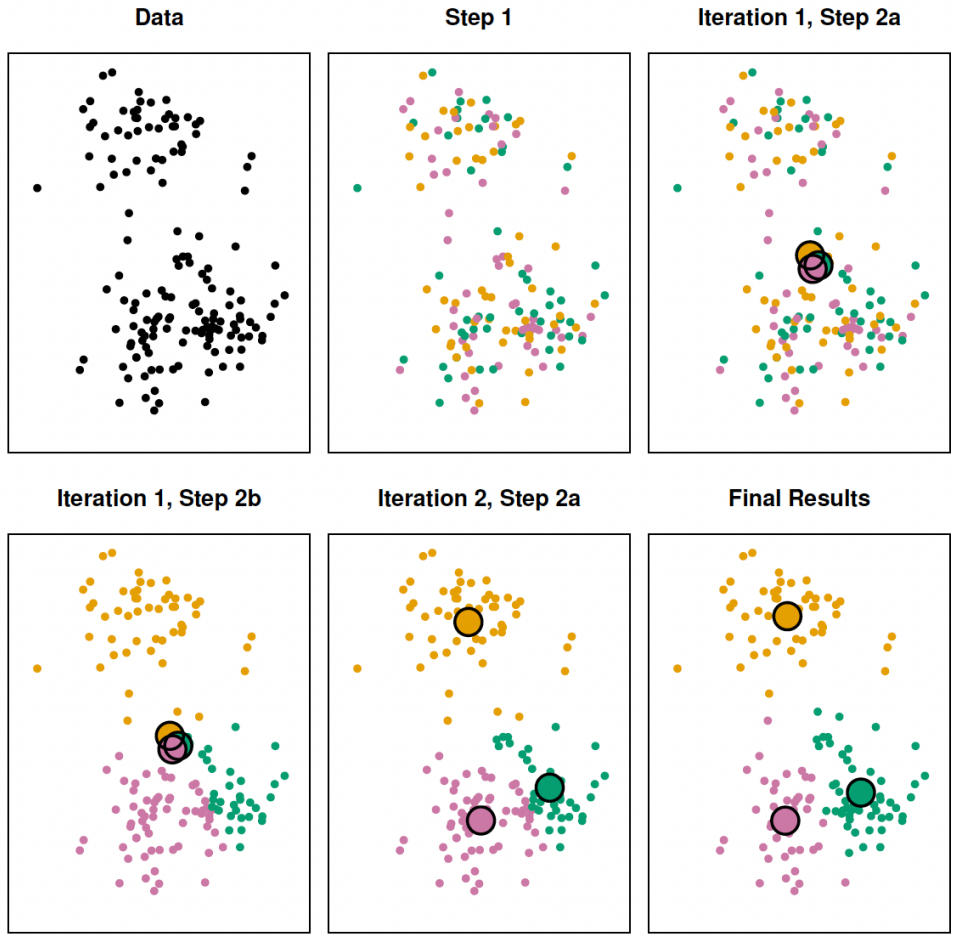
$$C(i) = \arg\min_k \|x_i - \mu_k\|^2$$

3. **Stop:** When assignments no longer change

Properties:

- Guaranteed to decrease objective at each step
- Converges to a **local** minimum (not necessarily global)
- Run multiple times with different initializations

K-Means: Visualization



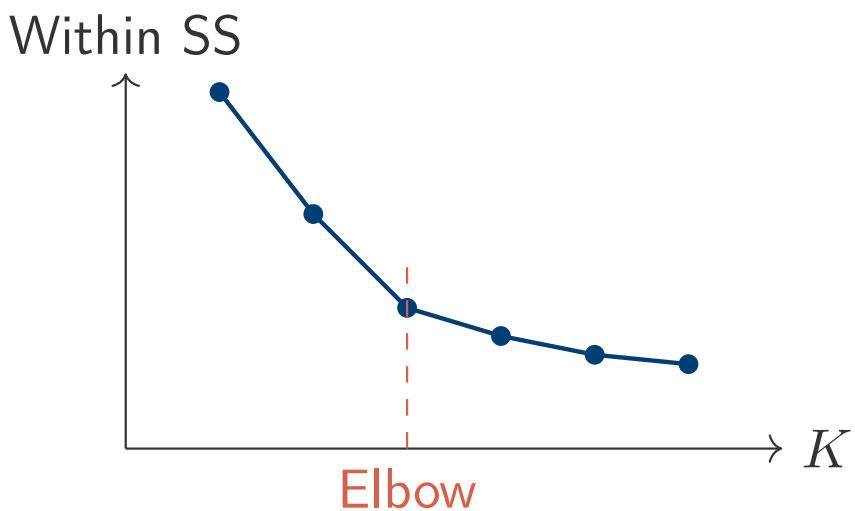
The algorithm alternates between:

- Assigning points to the nearest centroid
- Recomputing centroids based on assignments

Choosing the Number of Clusters K

Problem: K-means requires us to specify K in advance

1. **Elbow method:**
   - Plot total within-cluster sum of squares vs. K
   - Look for "elbow" where improvement slows



2. **Domain knowledge:** In finance, often have natural groupings (sectors, regimes, anomalies)

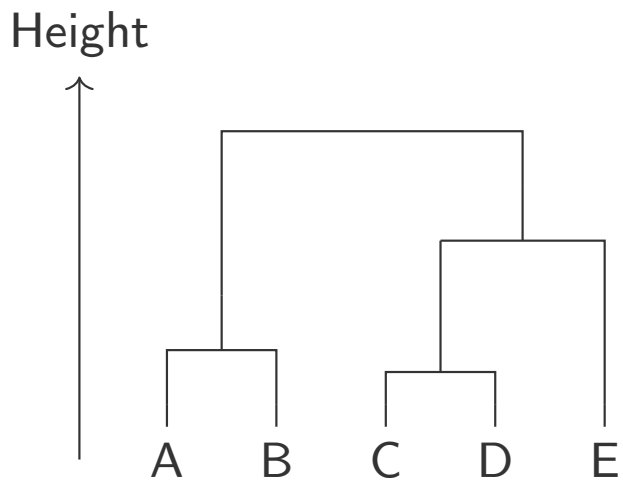
Hierarchical Clustering

Alternative approach: Build a hierarchy of clusters

Agglomerative (bottom-up):

1. Start: Each observation is its own cluster
2. Merge: Combine the two closest clusters
3. Repeat until all observations are in one cluster

Result: A **dendrogram** showing the merging sequence



Advantage: Don't need to pre-specify K; cut dendrogram at desired height

Linkage Methods

How do we define "distance" between clusters?

Linkage	Definition
Single	Minimum distance between any two points
Complete	Maximum distance between any two points
Average	Average distance between all pairs
Ward	Increase in within-cluster variance after merging

Recommendations:

- **Ward:** Tends to produce compact, equal-sized clusters (often best)
- **Complete:** Produces compact clusters
- **Single:** Can create long "chains" (often problematic)

K-Means vs. Hierarchical: Trade-offs

	K-Means	**Hierarchical**
Specify K?	Yes, in advance	No (cut dendrogram)
Scalability	$O(nKt)$ — fast	$O(n^2)$ or $O(n^3)$ — slow
Cluster shape	Spherical	Flexible
Deterministic?	No (random init)	Yes
Hierarchical structure?	No	Yes (dendrogram)

In practice:

- K-means: Large datasets, when K is known or tunable
- Hierarchical: Smaller datasets, want to explore structure, interpretability
- Often: Use hierarchical for exploration, K-means for final clustering

Important Considerations for Clustering

1. **Feature scaling:**
   - Clustering uses distances — scale matters!
   - Usually standardize features before clustering
2. **Dimensionality:**
   - High dimensions $\Rightarrow$ distances become less meaningful
   - Consider PCA before clustering ("spectral clustering" approach)
3. **Outliers:**
   - K-means sensitive to outliers (affects centroids)
   - Consider robust alternatives or remove outliers first
4. **Validation:**
   - No "ground truth" in unsupervised learning
   - Use multiple methods, check stability, domain validation

Recap: PCA and Clustering

What We Learned in Part I

Principal Component Analysis:

- Find linear combinations that maximize variance
- Eigenvalue decomposition of covariance matrix
- First few PCs often capture most variation
- Standardization matters when scales differ

Clustering:

- K-means: Partition into K groups, minimize within-cluster variance
- Hierarchical: Build dendrogram, cut at desired level
- Validation: Elbow method, silhouette score

Today's Focus

How do we apply these methods to **real financial problems**?

PCA for Statistical Factor Models

The Factor Model Framework

Classic asset pricing: Returns are driven by common factors

$$R_{it} - R_f = \alpha_i + \beta_{i1} F_{1t} + \beta_{i2} F_{2t} + \cdots + \beta_{iK} F_{Kt} + \varepsilon_{it}$$

Two approaches to identifying factors:

Economic factors:

- Fama-French: MKT, SMB, HML
- Based on economic theory
- Interpretable
- May miss important information

Statistical factors:

- PCA on returns
- Data-driven
- Optimal variance capture
- Less interpretable

Key Question

How do statistical factors compare to economic factors?

PCA on Stock Returns

Setup:

- n stocks, T time periods (e.g., 500 stocks, 120 months)
- Each stock has a time series of returns
- Stocks are correlated — they move together

What PCA does:

- Finds the common patterns driving returns across all stocks
- PC1: The single direction that explains most co-movement
- PC2: The next most important pattern (uncorrelated with PC1)
- And so on...

What we get:

- **Factor returns:** A time series for each PC (like a portfolio return)
- **Factor loadings:** How much each stock is exposed to each factor
- **Variance explained:** How important each factor is

Intuition

PCA extracts the "common factors" that drive stock returns

Empirical Evidence: How Many Factors?

Typical finding for US equities:

Component	Variance Explained	Cumulative
PC1	35–45%	35–45%
PC2	8–12%	45–55%
PC3	4–6%	50–60%
PC4	2–4%	55–65%
PC5	2–3%	58–68%

Key observations:

- First PC dominates — this is the "market factor"
- First 3–5 PCs explain majority of covariation
- Remaining PCs capture idiosyncratic noise

Implication

Stock return covariances can be well-approximated by a **low-rank** structure

What Do the PCs Represent?

Examining the loadings reveals economic interpretation:

Characteristic	PC1	PC2	PC3
Mean loading	$+$ (uniform)	≈ 0	≈ 0
Size correlation	Low	High	Low
Value correlation	Low	Low	High

Typical interpretation:

- **PC1:** Market factor (all stocks load positively and similarly)
- **PC2:** Often correlates with size (small vs. large)
- **PC3:** Often correlates with value (high vs. low B/M)

Insight: Statistical factors (PCA) and economic factors (Fama-French) are related but not identical!

PCA Factors vs. Fama-French Factors

How correlated are PCA factors with FF factors?

	MKT	SMB	HML
PC1	0.95+	0.10	0.05
PC2	0.15	0.60–0.80	0.20
PC3	0.10	0.20	0.50–0.70

Key findings:

- PC1 $\approx$ MKT (very high correlation)
- PC2 and PC3 are related to SMB/HML, but not identical
- PCA factors capture size/value *plus* other sources of covariation

Trade-off

PCA: Optimal variance capture, less interpretable
FF: Economically motivated, may miss some covariation

Application: Using PCA Factors for Risk Management

Why use PCA factors?

1. **More stable covariance estimates:**
   - Problem: With 500 stocks and 60 months, sample covariance is noisy
   - Solution: Estimate covariance using only the first few PCs
   - Result: Smoother, more reliable risk estimates
2. **Simpler portfolio optimization:**
   - Instead of estimating 500×500 covariances...
   - ...estimate exposures to 5 factors
   - Far fewer parameters = less estimation error
3. **Risk decomposition:**
   - "How much of my portfolio risk comes from the market?"
   - "How much from sector bets? From stock-specific risk?"
   - PCA separates systematic from idiosyncratic risk

Rule of Thumb

When you have more stocks than time periods, PCA-based methods are more reliable than using raw sample covariances.

Clustering for Portfolio Construction

Why Cluster Stocks?

Traditional approach: Group stocks by sector (GICS, SIC codes)

Problems:

- Sectors are static, markets evolve
- Amazon: Tech? Retail? Cloud? Streaming?
- Correlations within sectors vary widely
- Miss stocks that behave similarly across sectors

Data-driven alternative: Cluster stocks by **return behavior**

Benefits:

- Groups reflect actual return correlations
- Updates automatically as relationships change
- Can reveal hidden structure

What Features to Use for Clustering?

Option 1: Raw returns

- Cluster based on return time series
- Problem: High dimensional, noisy

Option 2: Return correlations

- Use correlation matrix as distance
- Distance $= 1 - \rho_{ij}$
- Groups stocks that move together

Option 3: PCA scores

- First apply PCA to reduce dimension
- Then cluster based on factor loadings
- Denoises the data, focuses on systematic variation

Recommendation

PCA + clustering often works best: reduce noise first, then cluster

Correlation-Based Distance: Measuring Similarity

The challenge: Clustering algorithms need a "distance" between stocks

Intuition: Stocks that move together should be "close"

Solution: Convert correlation ρ_{ij} to distance $d_{ij} = \sqrt{2(1 - \rho_{ij})}$

If two stocks have...	Correlation	Distance
Identical returns	$\rho = 1.0$	0 (very close)
Similar movements	$\rho = 0.8$	0.63
No relationship	$\rho = 0.0$	1.41
Opposite movements	$\rho = -1.0$	2.0 (very far)

Why this makes sense:

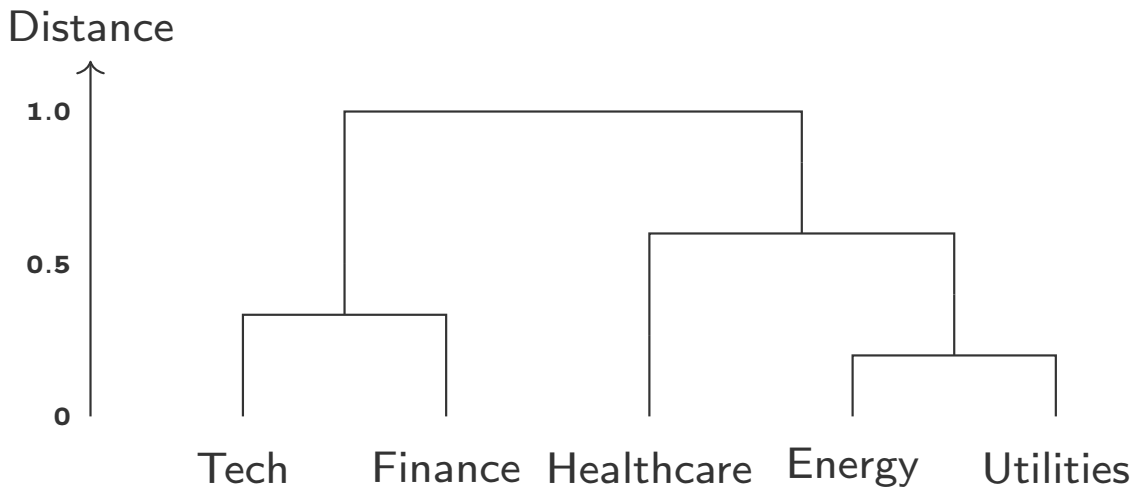
- Apple and Microsoft (high correlation) $\rightarrow$ small distance $\rightarrow$ same cluster

Key Point

We're grouping stocks by **how they behave**, not by industry labels

Hierarchical Clustering of Stocks

Example: S&P 500 stocks clustered by return correlation



Typical findings:

- Stocks often cluster by sector (industry effects)
- But not always! Data-driven clusters can cross sector boundaries
- Defensive stocks (utilities, staples) often group together
- Cyclical stocks (finance, industrials) form another group

Portfolio Diversification via Clustering

Traditional approach: Equal-weight across sectors

- Problem: Sectors are arbitrary; stocks within sectors may be very different; ignores actual return correlations

Clustering approach: Diversify across **data-driven** groups

- Group stocks by how they actually move together
- Allocate across clusters, then within clusters
- Result: True diversification based on return behavior

Why this matters:

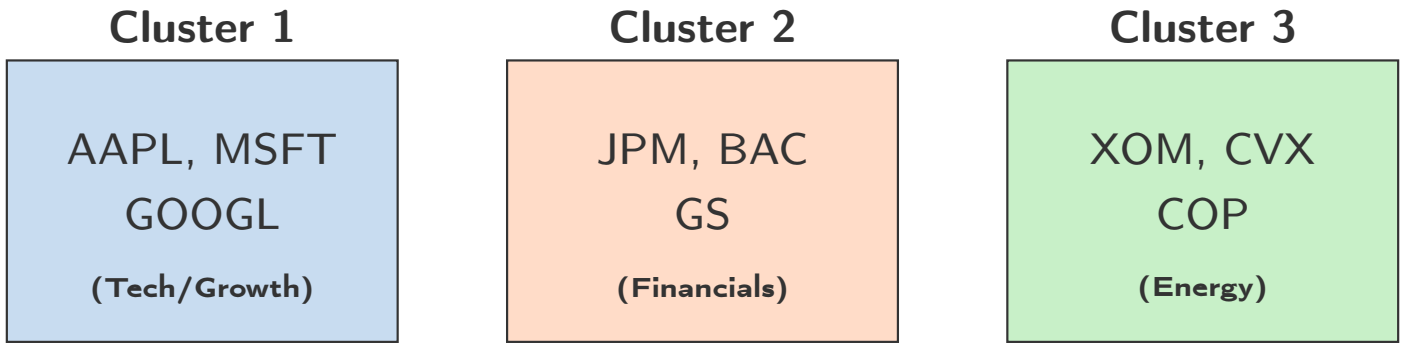
- Two "tech" stocks might behave very differently (Apple vs. Oracle)
- A "consumer" stock might move like tech (Amazon)
- Clustering reveals the *true* groupings in your data

Key Insight

Diversification means spreading risk across **uncorrelated** bets, not just different labels

Example: Cluster-Based Allocation

Suppose clustering reveals 3 groups among 9 stocks:



Simple cluster-based allocation:

1. Allocate equally across clusters: 33.3% each
2. Within each cluster, equal-weight the stocks: 11.1% per stock

Compare to naive equal-weight (11.1% each):

- If Tech cluster has 3 stocks but is highly correlated internally...
- ...you're putting 33% in one "bet" without realizing it
- Cluster-aware allocation makes this explicit

Regime Detection with Clustering

Market Regimes

Observation: Market behavior changes over time

- **Bull markets:** Low volatility, positive drift, low correlations
- **Bear markets:** High volatility, negative drift, high correlations
- **Crisis periods:** Extreme volatility, correlations spike to 1

Why does this matter?

- Risk management: Volatility and correlations are regime-dependent
- Trading strategies: What works in one regime may fail in another
- Asset allocation: Defensive positioning in crisis, aggressive in recovery

Idea

Use clustering to identify **time periods** with similar market conditions

Clustering Time Periods

Approach: Cluster based on market characteristics

Features for each time period (e.g., month):

- Market return
- Realized volatility
- Average correlation among stocks
- Spread between high and low beta stocks
- Credit spread, VIX, term spread

Process:

1. Compute features for each time period
2. Standardize features
3. Apply K-means or hierarchical clustering
4. Label periods by cluster ("regimes")

Example: Regime Detection Results

Typical 3-regime structure:

Feature	**Regime 1** (Bull)	**Regime 2** (Normal)	**Regime 3** (Crisis)
Market return	High (+)	Moderate	Low (−)
Volatility	Low	Medium	High
Avg correlation	0.2–0.3	0.3–0.5	0.6–0.8
Frequency	30%	55%	15%

Applications:

- **Conditional strategies:** Different model/allocation per regime
- **Risk budgeting:** Reduce risk when crisis regime detected
- **Factor timing:** Some factors work better in certain regimes

Regime Detection Using Goyal-Welch Data

Our data: Same macroeconomic predictors from Weeks 2–3

Features we can use:

- Market return (MktRf)
- Volatility proxy (svar — stock variance)
- Term spread (tms)
- Default spread (dfy)
- Dividend yield (d_y)

Question: Can we identify distinct market "states" in the data?

Exercise in Tutorial

Apply K-means clustering to monthly observations of GWZ variables. Do the clusters correspond to economically meaningful regimes?

PCA for Dimensionality Reduction in Prediction

The Curse of Dimensionality Revisited

Recall from Week 2: Many predictors create problems

- Overfitting: Model fits noise
- Multicollinearity: Correlated predictors, unstable estimates
- Computational cost: Large covariance matrices

Solutions we've seen:

- Regularization (Ridge, Lasso): Shrink or select coefficients
- Tree ensembles: Automatic feature selection

Alternative approach: **Reduce dimensions first**, then predict

PCA for Prediction

Replace p original predictors with $k \ll p$ principal components

Principal Component Regression (PCR)

Procedure:

1. Apply PCA to predictors $\mathbf{X}$, get k principal components $\mathbf{F}$
2. Regress target $\mathbf{y}$ on $\mathbf{F}$ instead of $\mathbf{X}$

$$y_t = \gamma_0 + \gamma_1 F_{1t} + \gamma_2 F_{2t} + \cdots + \gamma_k F_{kt} + \varepsilon_t$$

Why does this help?

- Fewer parameters: k instead of p
- PCs are uncorrelated: No multicollinearity!
- Noise reduction: Drop PCs with small eigenvalues

Important caveat:

- PCA maximizes *variance in* $\mathbf{X}$, not prediction of $\mathbf{y}$
- High-variance directions may not be the most predictive!

PCR vs. Ridge: A Comparison

Both shrink the effective number of parameters, but differently:

	PCR	**Ridge**
Approach	Drop less relevant PCs	Shrink all directions
Shrinkage pattern	Discrete (keep or drop)	Continuous
Uses y?	No (unsupervised)	Yes (supervised)
Computation	Fast (closed form)	Fast (closed form)
Interpretation	PCs may be interpretable	Original variables

Key insight: Ridge shrinks *more* in low-variance directions, which is often what we want. PCR drops them entirely.

In practice, Ridge often outperforms PCR unless the factor structure in the data is very clear.

Application: Diffusion Index Forecasting

Stock & Watson (2002): Use PCA for macroeconomic forecasting

Setting:

- Many macro predictors (100+ series)
- Want to forecast inflation, output, etc.
- Too many predictors for standard regression

Diffusion index approach:

1. Apply PCA to macro predictors
2. Keep first few PCs ("diffusion indices")
3. Use PCs as predictors in forecasting regression

Why "diffusion"?

- PCs summarize "diffuse" information across many series
- Each series contributes, none dominates

Finance application: Rapach, Strauss & Zhou (2013) use diffusion indices for international return prediction

Summary and Next Steps

Key Takeaways from Today

1. **PCA for factor models:**
   - Statistical factors from return covariances
   - PC1 $\approx$ market, other PCs capture size/value/etc.
   - Useful for covariance estimation, risk decomposition
2. **Clustering for portfolio construction:**
   - Data-driven alternative to sector classification
   - Correlation-based distance captures return co-movement
   - Hierarchical clustering reveals structure
3. **Regime detection:**
   - Cluster time periods by market characteristics
   - Identifies bull/bear/crisis states
   - Useful for conditional strategies
4. **PCR for prediction:**
   - Reduce dimensions before regression
   - Often inferior to Ridge (unsupervised vs. supervised)

Readings and Next Steps

Readings for Today's Lecture:

- James et al. (2023), Chapter 12 *[Recomm.]*
- Hastie, Tibshirani & Friedman (2009), Ch. 14.3, 14.5 *[Supp.]*
- López de Prado (2016), "Building Diversified Portfolios that Outperform Out-of-Sample," *JPM* *[Supp.]*
- Stock & Watson (2002), "Forecasting Using Principal Components," *JASA* *[Supp.]*
- Rapach, Strauss & Zhou (2013), "International Stock Return Predictability," *JF* *[Supp.]*

Coming up next Week:

- Model interpretability and explainability
  - Feature importance methods
  - SHAP values
  - Partial dependence plots
- Why interpretability matters for finance and regulation

Questions?