

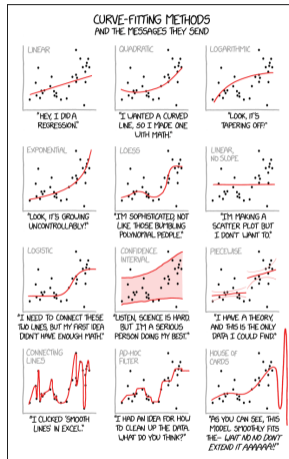
# PCA & t-SNE

Mini-Lecture: Seeing the Big Picture in High Dimensions

Methods & Algorithms

MSc Data Science – Spring 2026

# When You Have Too Many Dimensions to Plot



XKCD #2048 by Randall Munroe (CC BY-NC 2.5)

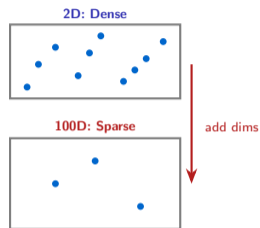
# The Curse of Dimensionality

## Why High-Dimensional Data Is Hard

- In high dimensions, all points become equally distant – nearest-neighbor methods break down
- A dataset that seems dense in 2D becomes sparse in 100D
- Visualization is impossible beyond 3 dimensions
- Many features are redundant or correlated – we need to compress without losing structure

### Key Question

Can we reduce 100 features to 2–3 while preserving the important patterns?



Bellman (1961) coined “curse of dimensionality” – the volume of space grows exponentially with dimensions

# PCA: Find the Directions of Maximum Variance

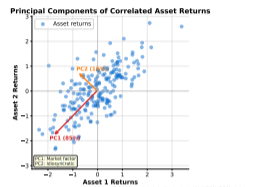
## Principal Component Analysis

- **Step 1:** Center the data (subtract the mean)
- **Step 2:** Find the direction of maximum variance – that is PC1
- **Step 3:** Find the next orthogonal direction of maximum variance – that is PC2
- Project data onto these new axes to reduce dimensions

Mathematically: eigendecomposition of the covariance matrix  $\Sigma = \frac{1}{n-1} X_c^T X_c$  (centered data).

## Insight

PCA is a rotation – it does not discard data, it reorders it by importance (variance).

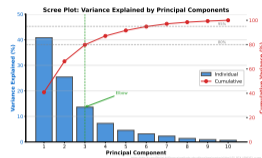


PCA preserves global structure (variance) – it finds the best linear summary of your data

# How Many Components Do We Keep?

## The Scree Plot

- Plot explained variance ratio for each component
- Look for the “elbow” – where additional components add little
- Rule of thumb: keep enough components to explain 90–95% of total variance
- Alternatively, use cross-validation on a downstream task



## Insight

If 3 components explain 95% of variance in 50 features, you have compressed 50D to 3D with minimal information loss.

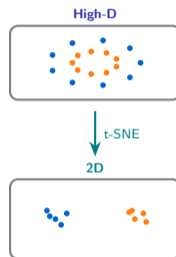
The scree plot is named after the geological term for rubble at the base of a cliff – you stop where the “cliff” flattens

## When Linear Projection Fails

- **t-SNE** (van der Maaten & Hinton, 2008) is a non-linear method for visualization
- Converts high-D distances to probabilities, then matches them in 2D
- Preserves **local structure**: nearby points stay nearby
- Warning: global distances are not meaningful in t-SNE plots

## Insight

t-SNE is for looking at your data, not for feeding into a model. Use PCA for preprocessing, t-SNE for visualization.



t-SNE clusters are real (local structure is preserved), but cluster distances and sizes are not – never interpret gaps

## Balancing Local vs Global Structure

- **Perplexity**  $\approx$  effective number of neighbors each point considers
- Low perplexity (5–10): tight local clusters, noisy global layout
- High perplexity (30–50): smoother layout, may merge distinct groups
- Always try multiple perplexity values before drawing conclusions

## Insight

There is no “correct” perplexity. If your conclusions change with perplexity, they are artifacts of the method, not real structure.

Perplexity	Behavior
5	Very local, fragmented clusters
30	Balanced (default in sklearn)
50	Smoother, global trends visible
100+	Over-smoothed, clusters merge

**Rule of thumb:** perplexity should be smaller than  $n/3$  where  $n$  is the number of data points.

Wattenberg et al. (2016) “How to Use t-SNE Effectively” – essential reading before interpreting t-SNE plots

# PCA vs t-SNE: When to Use Which?

## Comparison Table

Property	PCA	t-SNE
Type	Linear	Non-linear
Preserves	Global variance	Local neighborhoods
Invertible	Yes (reconstruct data)	No
Speed	Fast ( $O(np^2)$ )	Slow ( $O(n^2)$ )
Deterministic	Yes	No (random initialization)
Use for modeling	Yes (preprocessing)	No (visualization only)
Interpretable axes	Yes (loadings)	No

## Practical Advice

Run PCA first to reduce to 30–50 dimensions, then apply t-SNE for visualization. This is faster and more stable than running t-SNE on raw high-D data.

PCA + t-SNE pipeline: PCA for compression and denoising, t-SNE for the final 2D visualization

## The Three Factors of Interest Rates

- PCA on yield curve data (1Y, 2Y, 5Y, 10Y, 30Y rates) reveals 3 dominant components
- **PC1 (~85%)**: Level – parallel shift of all rates
- **PC2 (~10%)**: Slope – short vs long rates diverge
- **PC3 (~3%)**: Curvature – belly of the curve moves

## Insight

3 components explain 98% of yield curve variation. Fixed income risk management is built on this PCA decomposition.

## Portfolio Risk with PCA:

Instead of tracking 10 maturities, track 3 factor exposures:

$$\begin{aligned}\Delta P &\approx D_1 \cdot \Delta PC1 \\ &+ D_2 \cdot \Delta PC2 \\ &+ D_3 \cdot \Delta PC3\end{aligned}$$

$D_k$ : factor duration (sensitivity to factor  $k$ )

---

Litterman & Scheinkman (1991) showed that 3 PCA factors explain >98% of US Treasury yield curve movements

## Summary: Dimensionality Reduction in 4 Takeaways

1. **Curse of dimensionality**: high-D data is sparse, distances become meaningless, and visualization is impossible without reduction
2. **PCA**: linear, fast, invertible – finds directions of maximum variance. Use for preprocessing and interpretable compression.
3. **t-SNE**: non-linear, slow, for visualization only – preserves local neighborhoods but not global distances. Always try multiple perplexity values.
4. **Finance**: yield curve PCA (level, slope, curvature) reduces 10+ maturities to 3 factors explaining 98% of variance

### Next Steps

Explore the deep dive slides for eigenvalue derivations, kernel PCA, and UMAP. Try the Jupyter notebook to run PCA on real financial data.

---

**PCA is a tool; t-SNE is a lens. Use PCA to simplify your data, use t-SNE to look at it.**