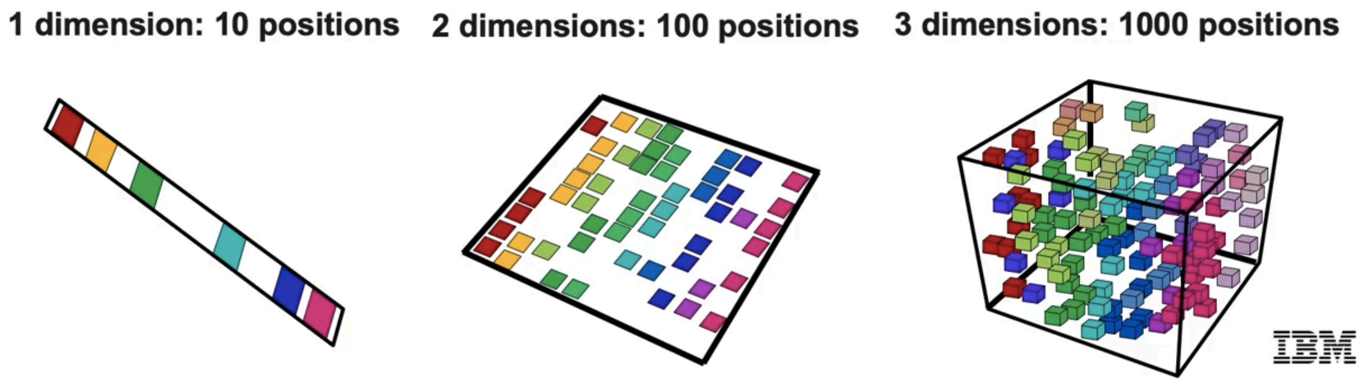
13) Dimensionality Reduction

- Part of unsupervised learning
- "Finding ways of representing our dataset in lower dimensions"
- This can help improve both, the performance and interpretability
- Dimensionality reduction can be done by:
    - Data can be represented by fewer dimensions/features
    - Reduce dimensionality by selecting subset (feature elimination)
    - Combine with linear and non-linear transformations (PCA)
- Common use cases:
    - Compressing high resolution images
    - Image processing & tracking

Curse of dimensionality

- In theory, increasing features should improve performance. However, in practice, too many features leads to worse performance. This can be due to there being uncorrelated features, the features creating more noise than signal, and so on
- Number of training examples required increases EXPONENTIALLY with dimensionality
- More dimensions lead to more expensive calculations
- In high dimensional space, points tend to be far apart. Thus, the distance measures perform poorly and incidence of outliers increases

If we have a 1D space (of 10 units), we need only need 6 observations to cover up 60% of the space. But in a 2D space (w/ each side being 10 units), we need 10x more observations (ie 60) to cover up 60% of the space. In a 3D space, the number needed will increase to 600



Rules of thumb for selecting an approach:

Method	Use case
PCA	Identify small number of transformed variables with different effects, preserving variance
KPCA	Useful for situations with non-linear PCA, but requires more computation than PCA
MDS	Like PCA, but new (transformed) features are determined based on preserving distance between points, rather than explaining variance
NMF	Useful when you want to consider only positive values (like word matrices and images)