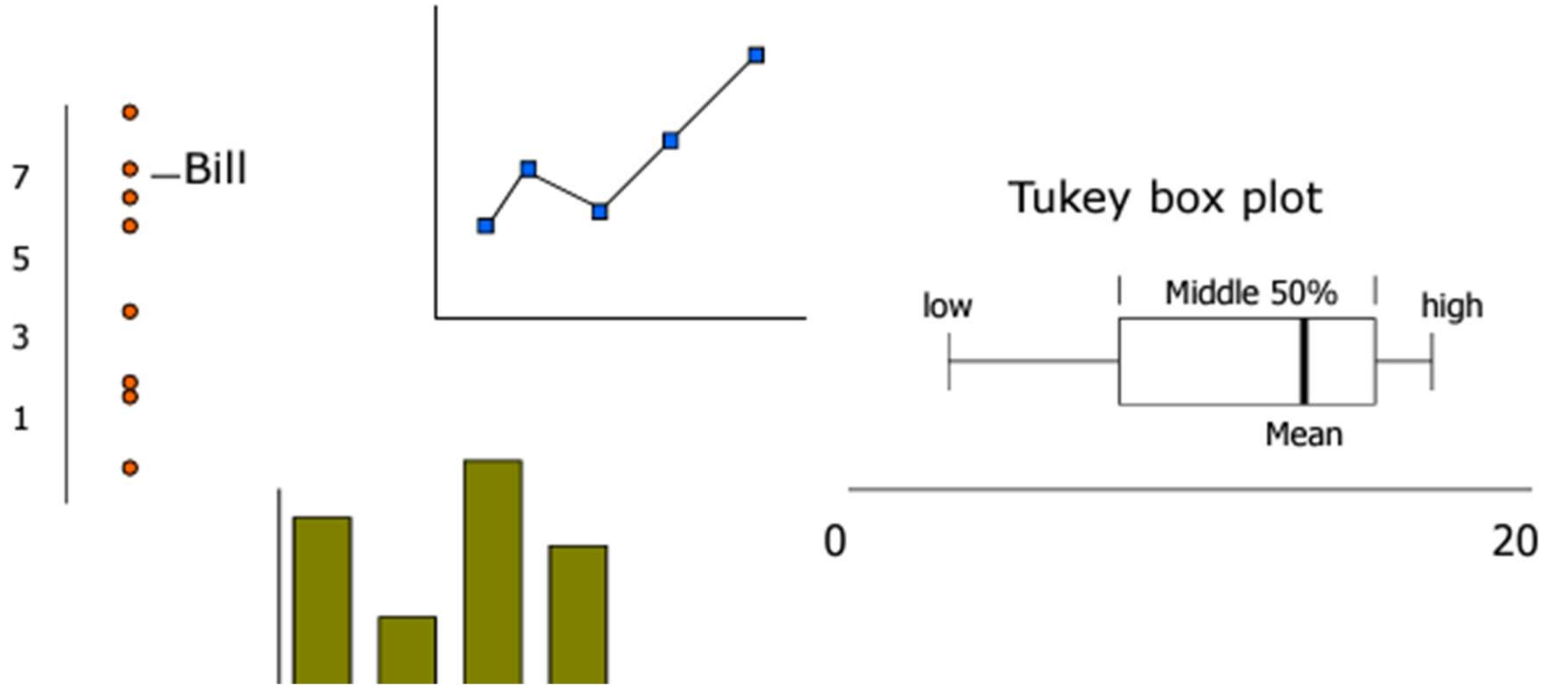
VISUALIZATION

Visualizing Large tables & Multidimensional data

Dimensions

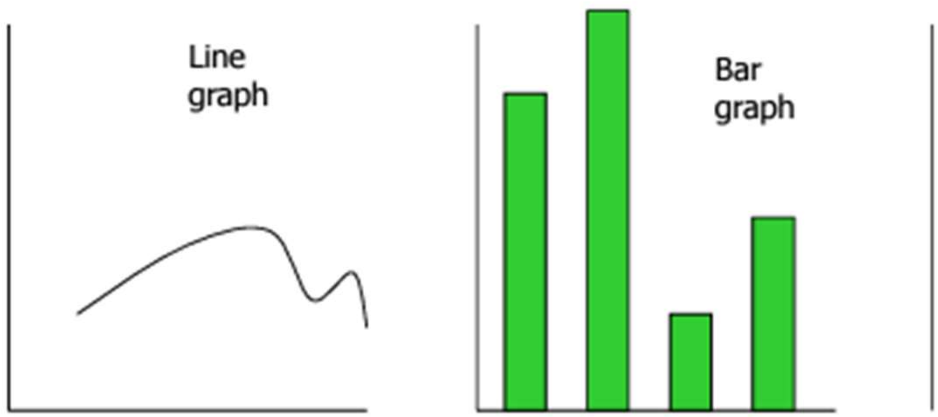
- Data sets of dimensions 1, 2, 3 are common
- Number of variables/attributes per class/item
  - 1 - Univariate data
  - 2 - Bivariate data
  - 3 - Trivariate data
  - >3 - Hypervariate data

Univariate Data



Views

- In univariate representations, we often think of the data case as being shown along one dimension, and the value in another



Y-axis is quantitative variable

See changes over consecutive values

Y-axis is quantitative variable

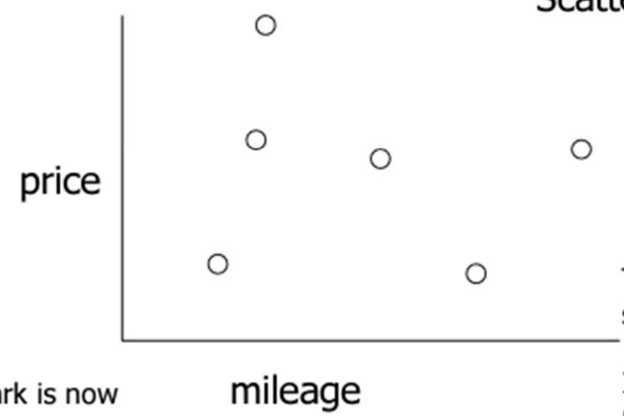
Compare relative point values

- We may think of graph as representing independent (data case) and dependent (value) variables
  - Independent on x-axis
  - Resultant dependent variables along y-axis

Bivariate Data

- Representations

Scatter plot is common

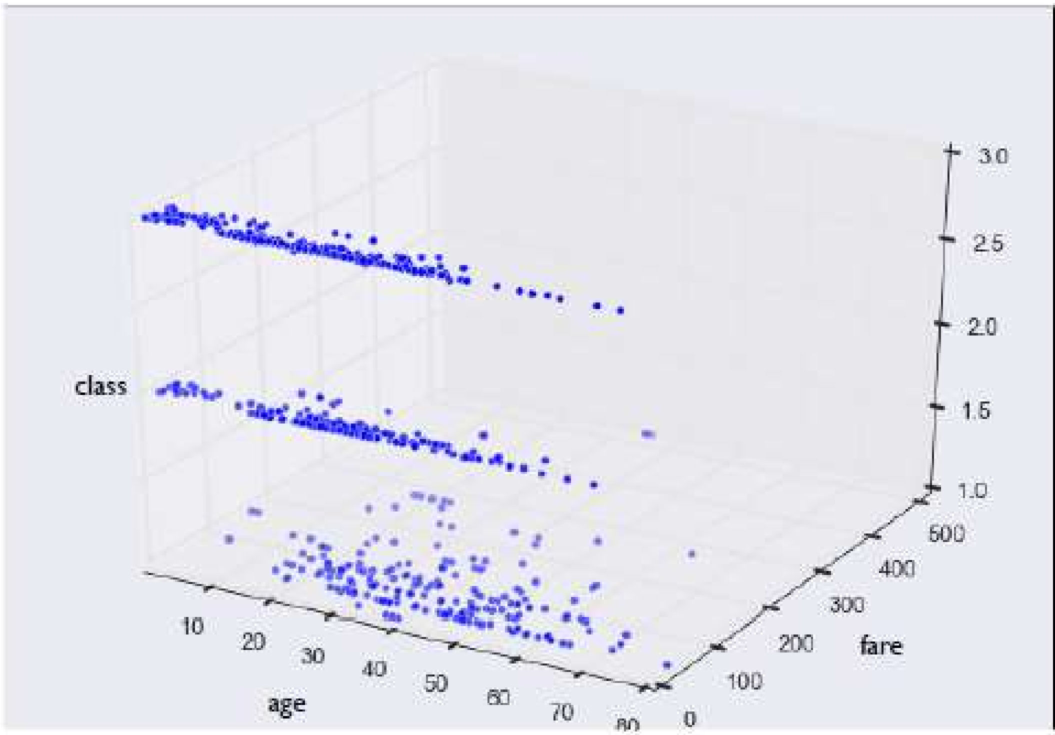


Two variables, want to see relationship

Is there a linear, curved or random pattern?

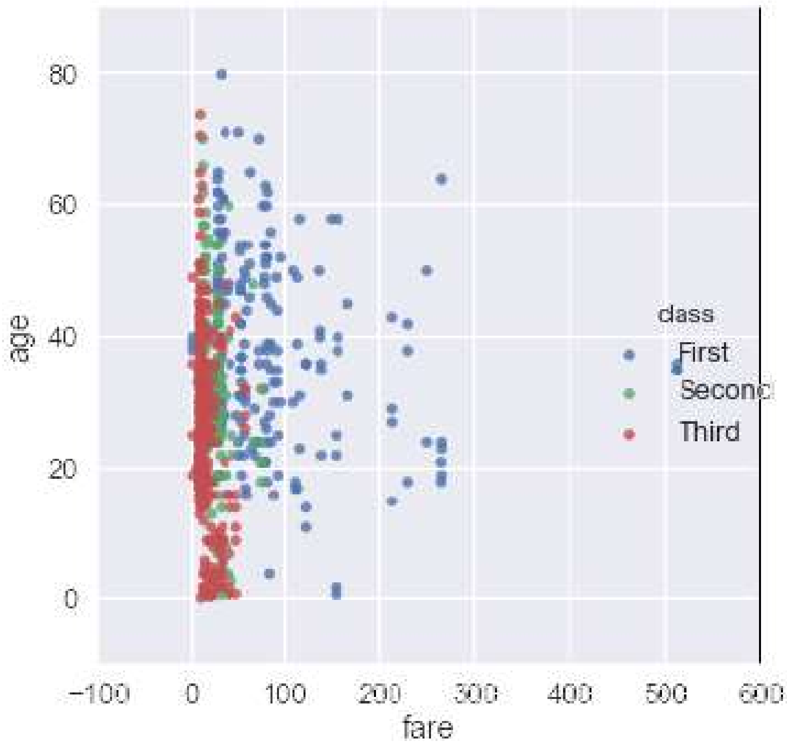
Each mark is now a data case

Trivariate Data



3D Scatterplots are difficult to understand

Trivariate Data



Map the 3rd dimension to some visual attributes

Visualizing Large Tables

- Tabular data, containing
  - Rows (items)
  - Columns (attributes or dimensions)
- How many records?
  - ~ 1000 – “just” visualization is fine
  - >> 10,000 – need analytical methods
- How many dimensions?
  - ~50 – tractable with “just” visualization
  - ~1000 – need analytical methods

Table Lens

- Spreadsheet is certainly one hypervariate data presentation
- Idea: Make the text more visual and symbolic
- Just leverage basic bar chart idea
- Problems:
  - Showing Categorical data

Table Lens

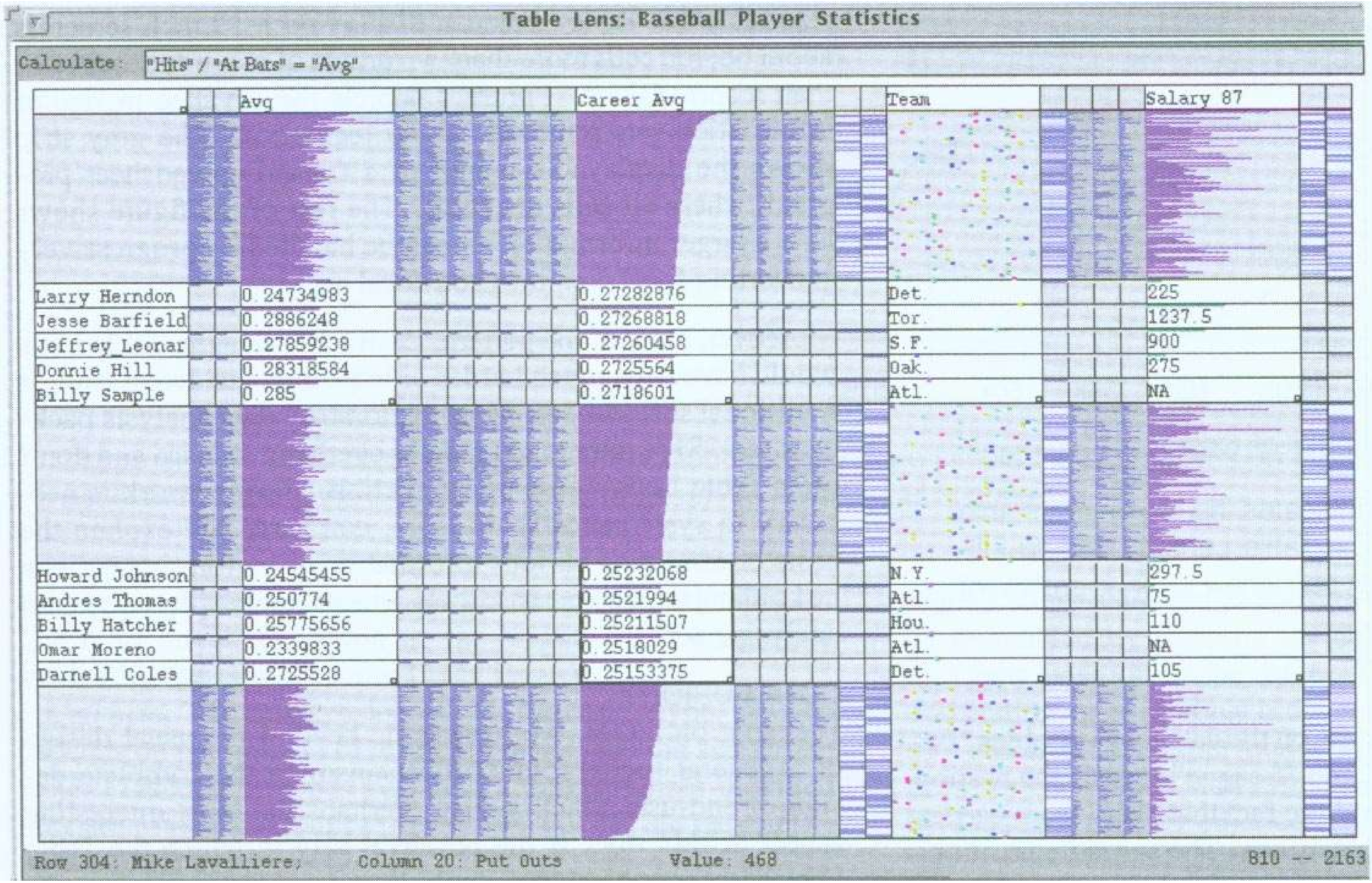


Table Lens: Baseball Player Statistics
Calculate: "Hits" / "At Bats" = "Avg"
Avg
Career Avg
Team
Salary 87
Larry Herndon 0.24734983 0.27282876 Det. 225
Jesse Barfield 0.2886248 0.27268818 Tor. 1237.5
Jeffrey_Leonar 0.27859238 0.27260458 S.F. 900
Donnie Hill 0.28318584 0.2725564 Oak. 275
Billy Sample 0.285 0.2718601 Atl. NA
Howard Johnson 0.24545455 0.25232068 N.Y. 297.5
Andres Thomas 0.250774 0.2521994 Atl. 75
Billy Hatcher 0.25775656 0.25211507 Hou. 110
Omar Moreno 0.2339833 0.2518029 Atl. NA
Darnell Coles 0.2725528 0.25153375 Det. 105
Row 304: Mike Lavalliere; Column 20: Put Outs Value: 468 810 -- 2163

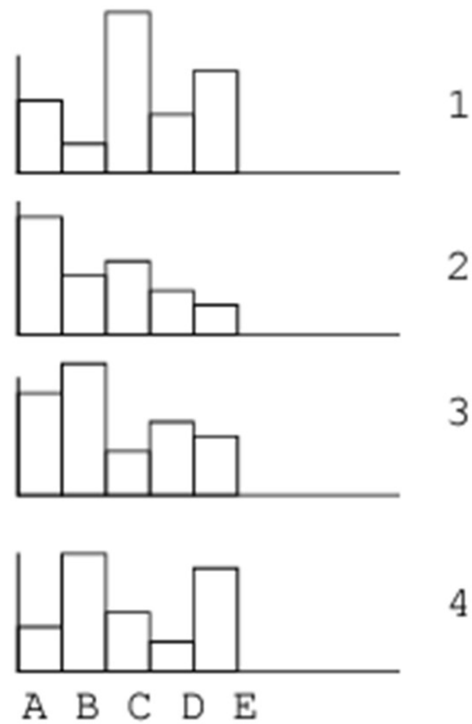
Visualizing Hypervariate Data

- Fundamentally, we have 2 geometric (position) display dimensions
- Various techniques for visualizing hypervariate data:
  - For data sets with >2 variables, we must project data down to 2D
  - Come up with visual mapping that locates each dimension into 2D plane
  - Computer graphics: 3D->2D projections
- Many ôther techniques have also been proposed

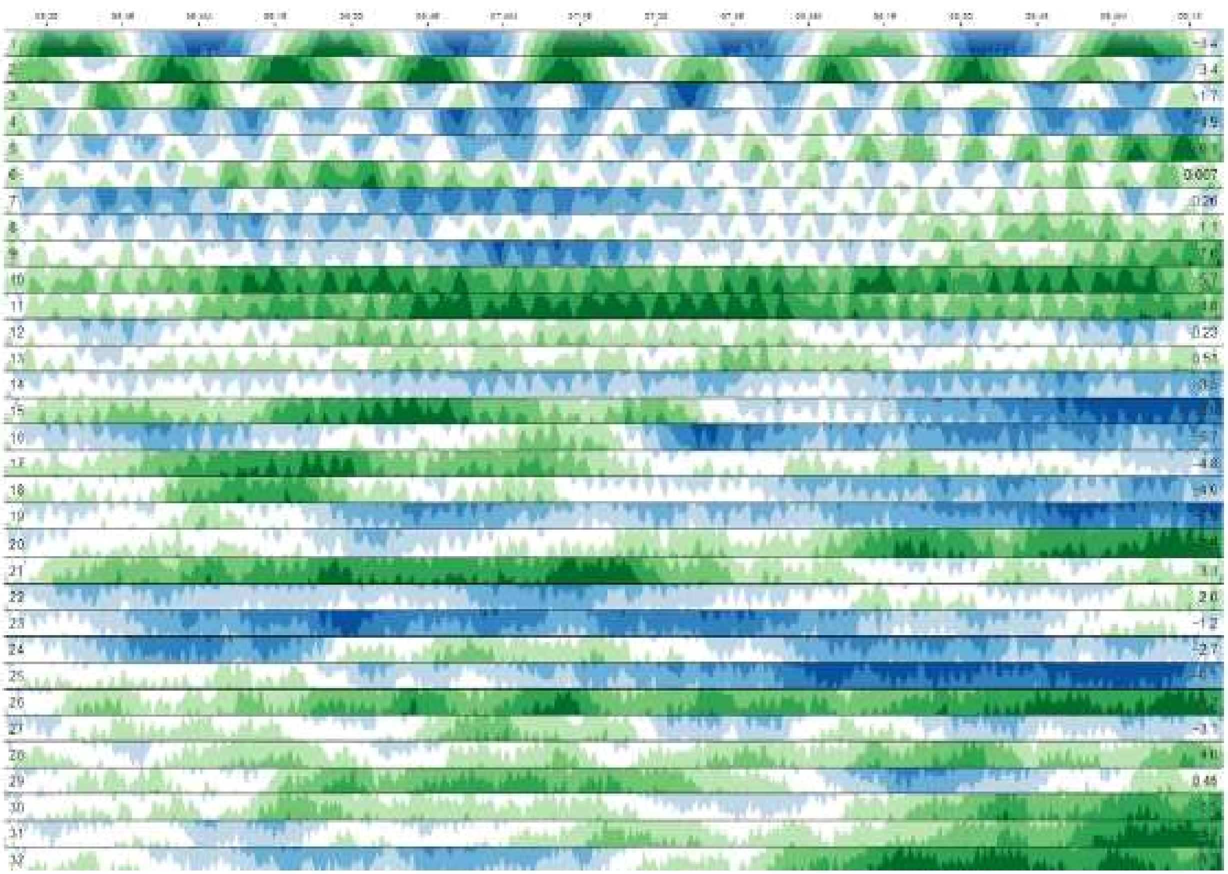
Multiple Views

Give each variable its own display

	A	B	C	D	E
1	4	1	8	3	5
2	6	3	4	2	1
3	5	7	2	4	3
4	2	6	3	1	5

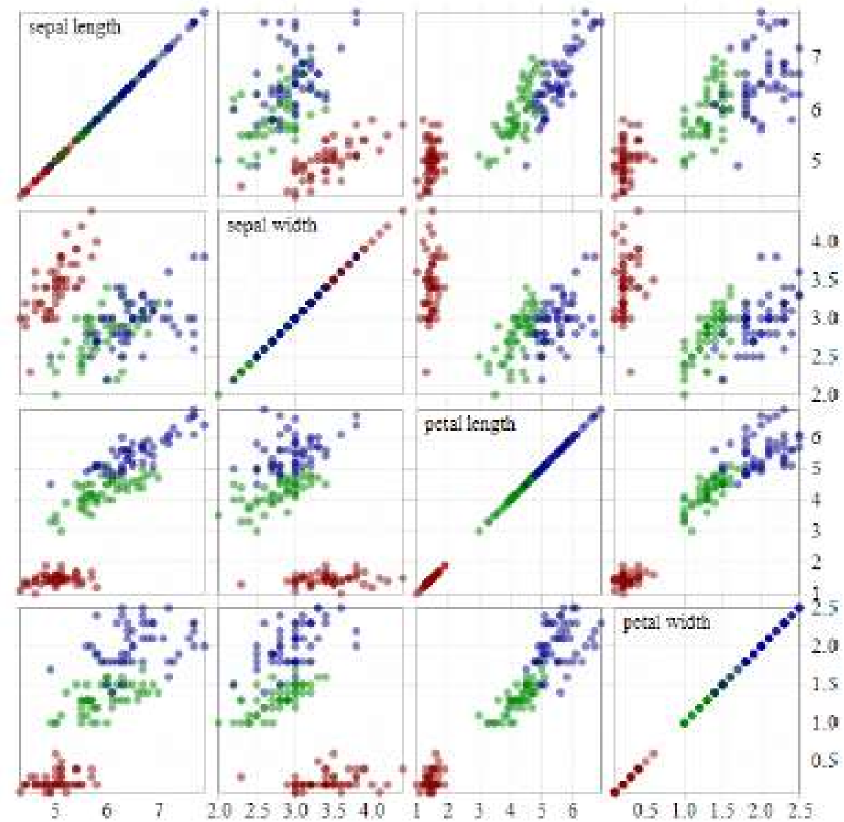


Multiple Line Charts



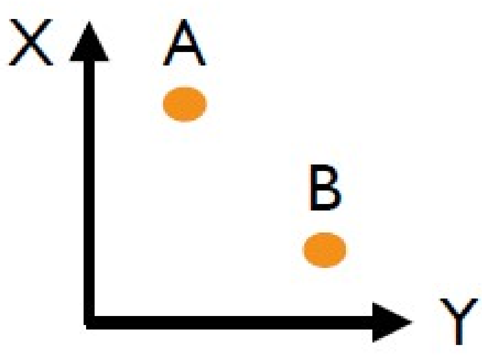
Scatterplot Matrices (SPLOM)

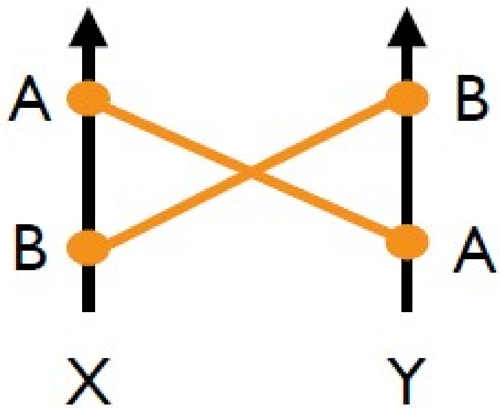
- Matrix of size d*d
- Each row/column is one dimension
- Each cell plots a scatterplot of two dimensions
- Scalability: ~20 dimensions, ~500-1krecords



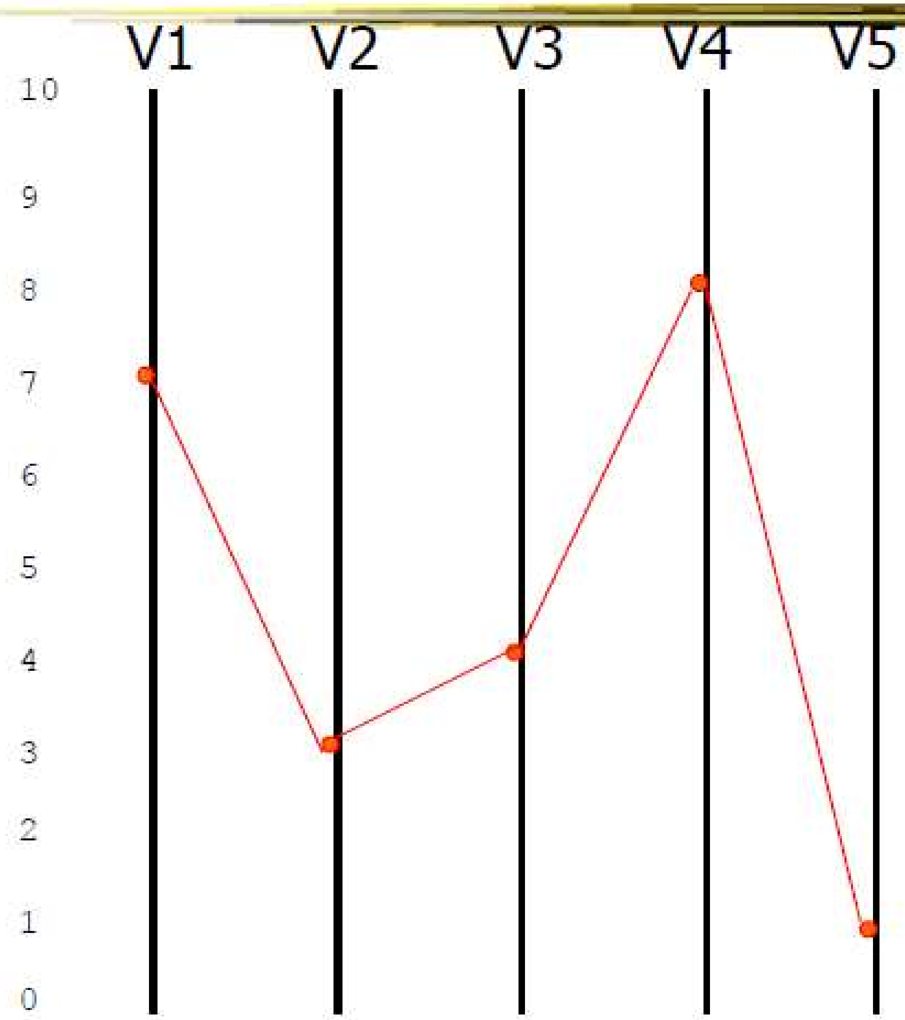
Parallel Coordinates

- Axes represent attributes
- Lines connecting axes represent items
- Suitable for
  - All tabular data types
  - Hypervariate data





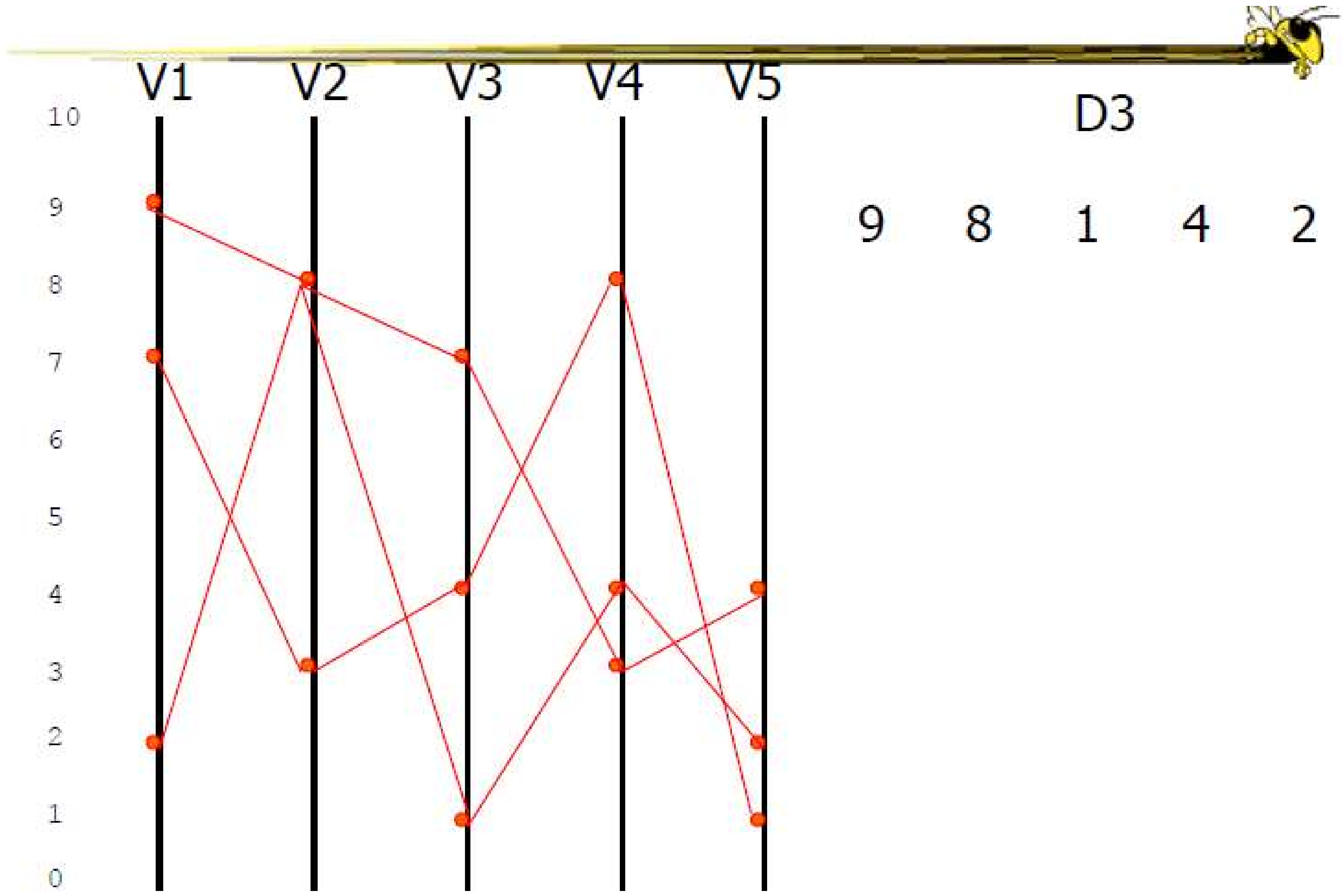
Parallel Coordinates



D1

7 3 4 8 1

Parallel Coordinates

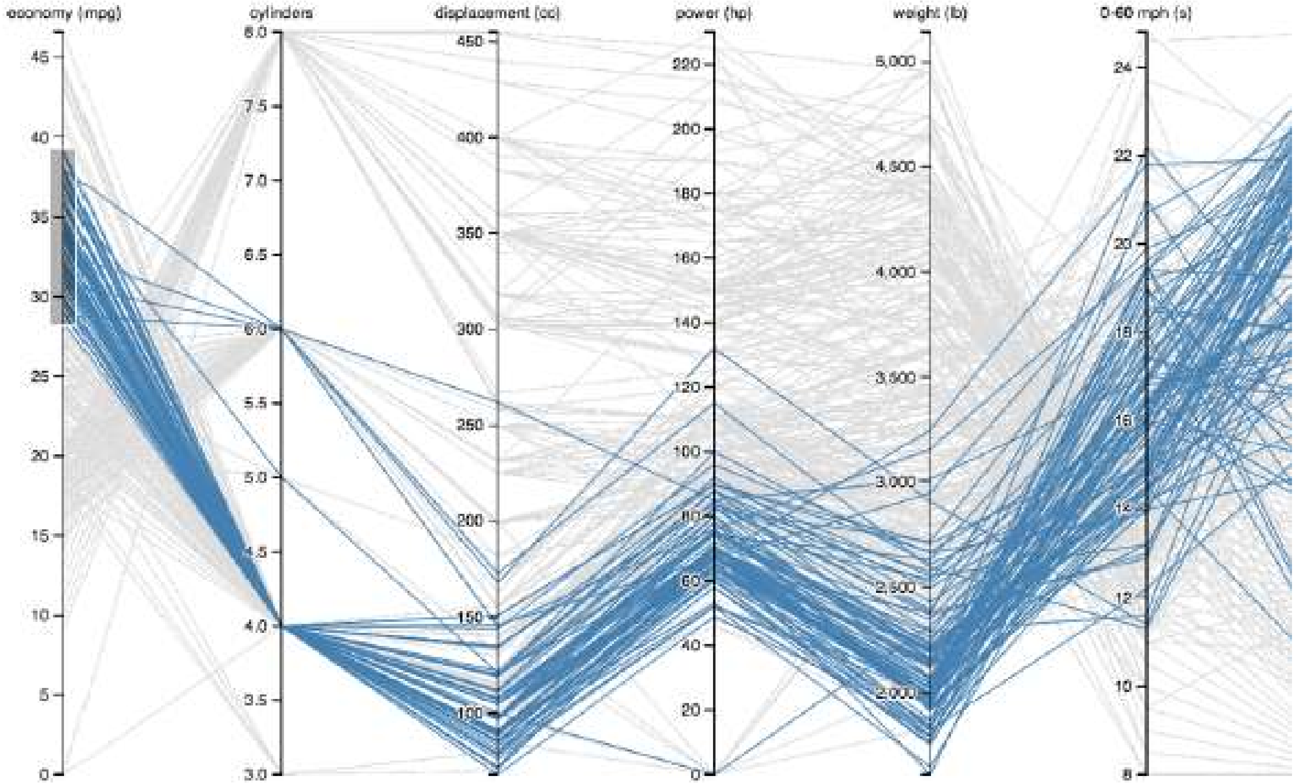


Exercise

- Examine the given table.
- Draw a Parallel Coordinate view of the data

Address	Price	Beds	Baths
1301 Robinson Court	$ 355,000	3	2
2479 North Bend Road	$ 109,900	1	1
897 Wiseman Street	$ 448,000	5	3
4960 Rosewood Lane	$ 849,900	3	2.5
4883 Hartland Avenue	$ 129,900	1	1

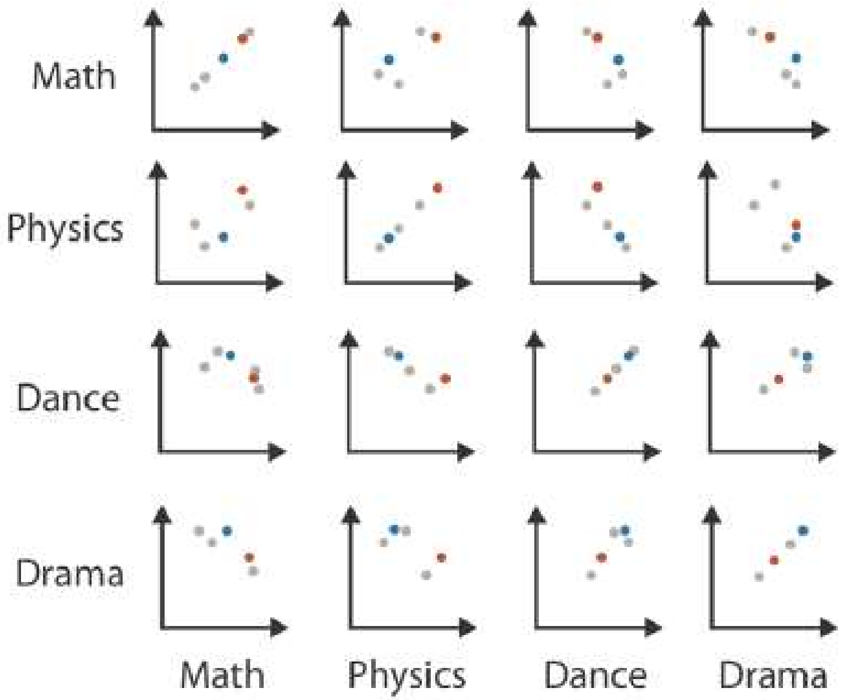
Parallel Coordinates



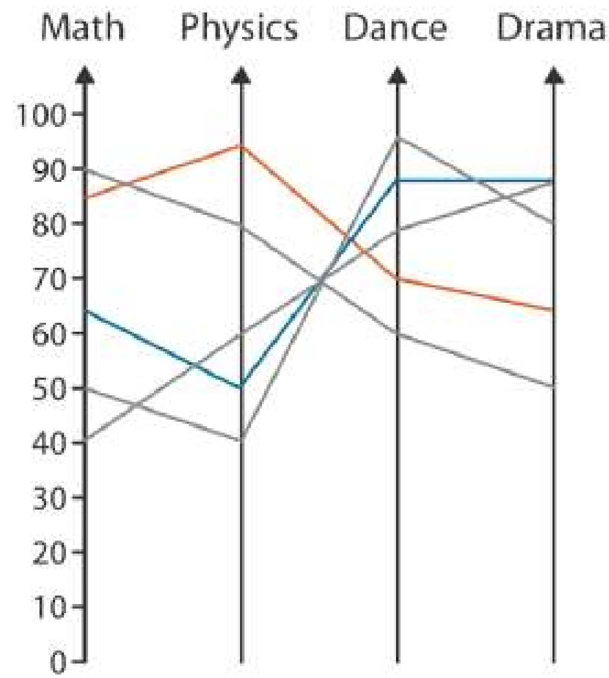
Table

Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90

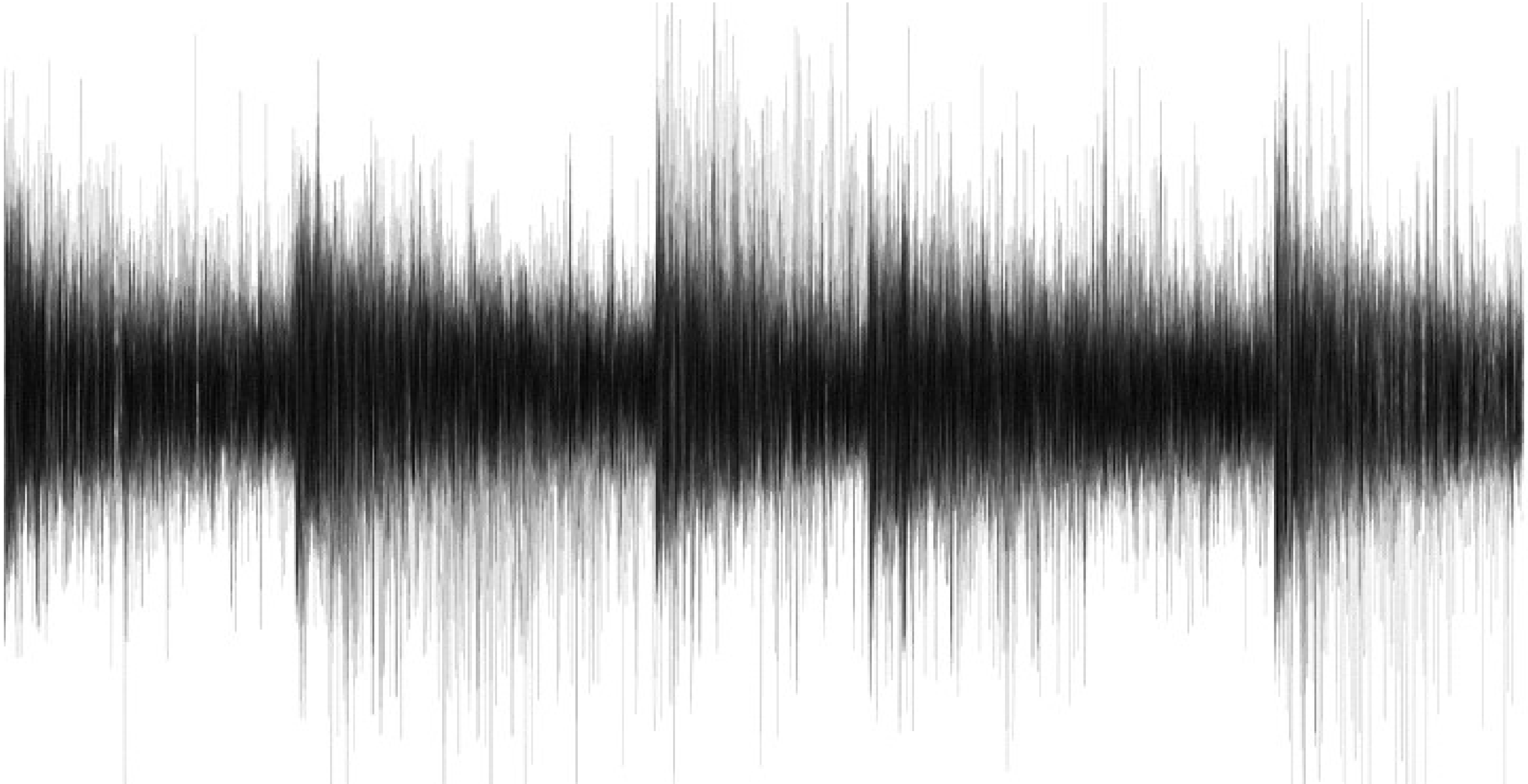
Scatterplot Matrix



Parallel Coordinates



Limitation – Scalability to many dimensions



Dimensional Reordering

Can you reduce clutter and highlight other interesting features in data by changing order of dimensions?

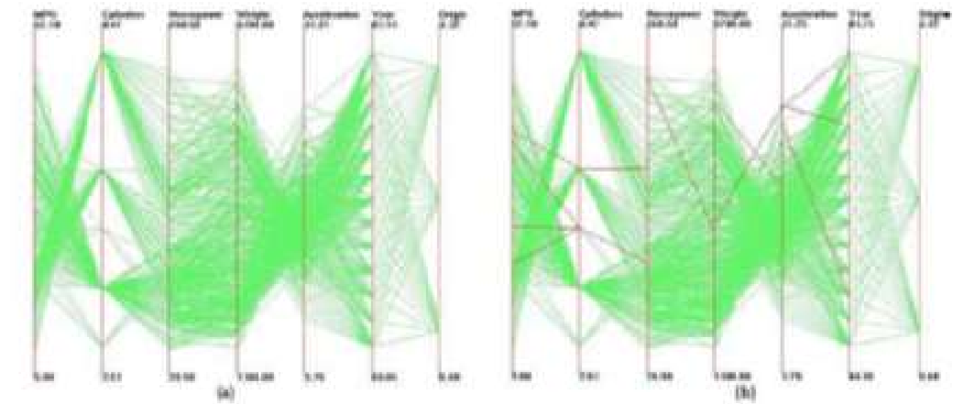
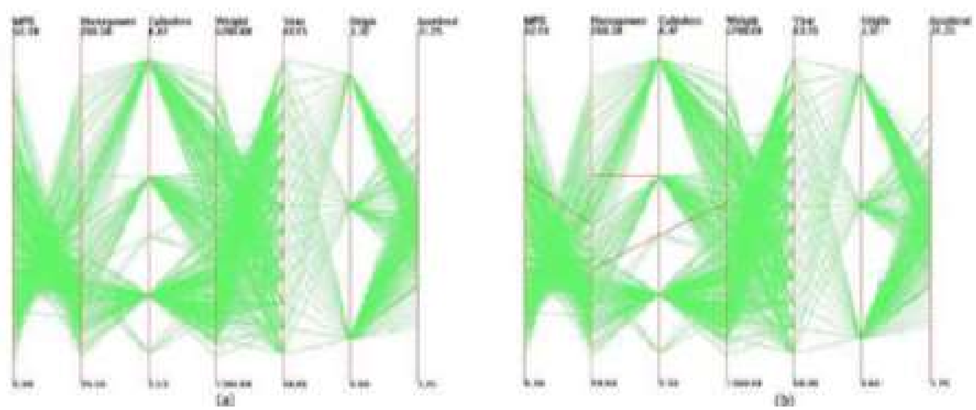
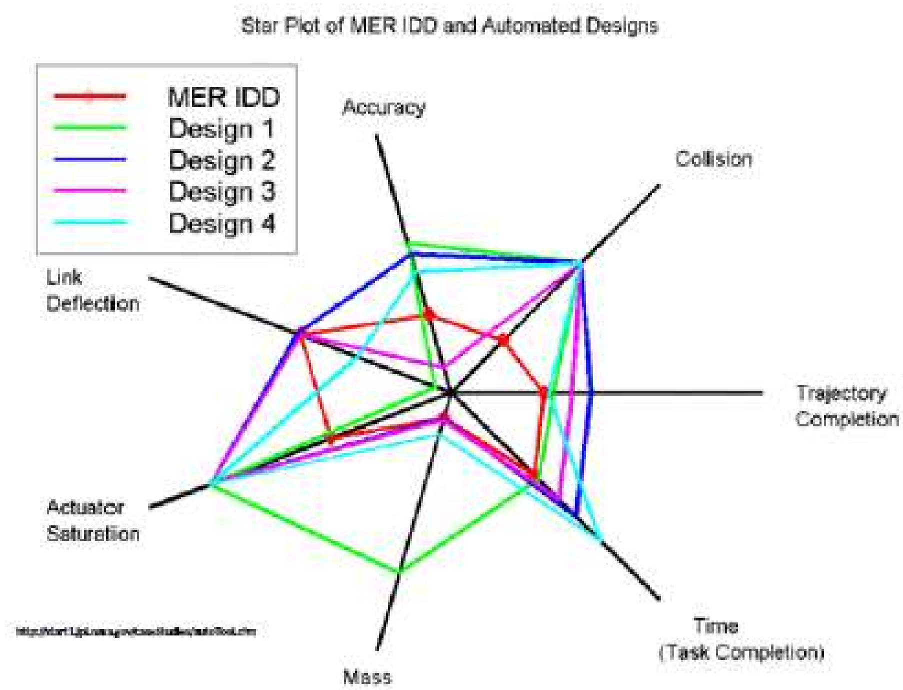


Figure 1: Parallel coordinates visualization of Cars dataset. Outliers are highlighted with red in (b).



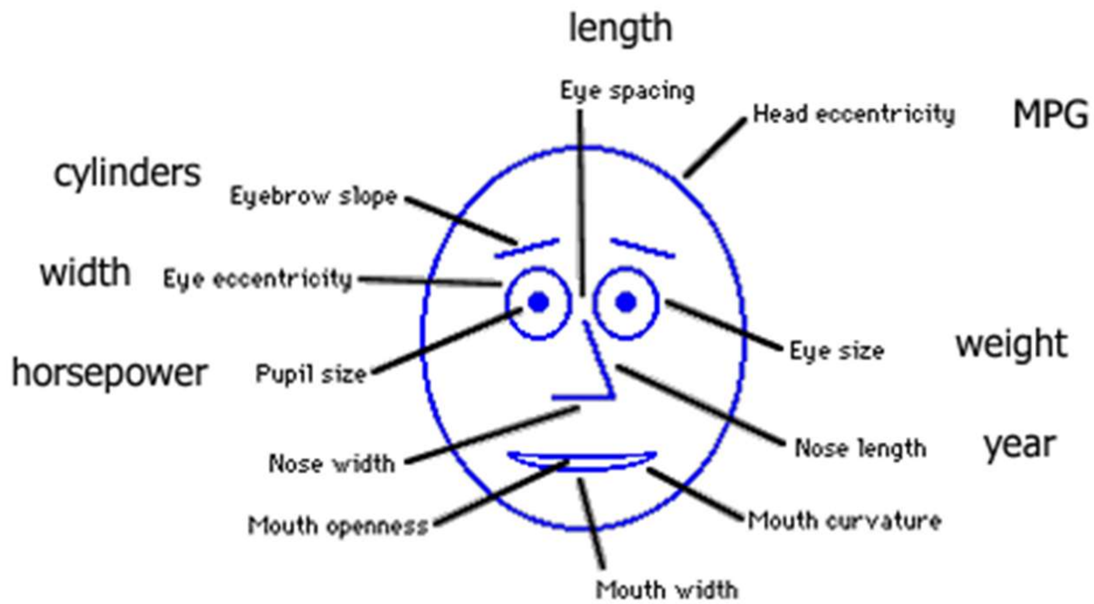
Star Plots

- Space out the n variables at equal angles around a circle
- Each “spoke” encodes a variable’s value
- Data point is now a “shape”

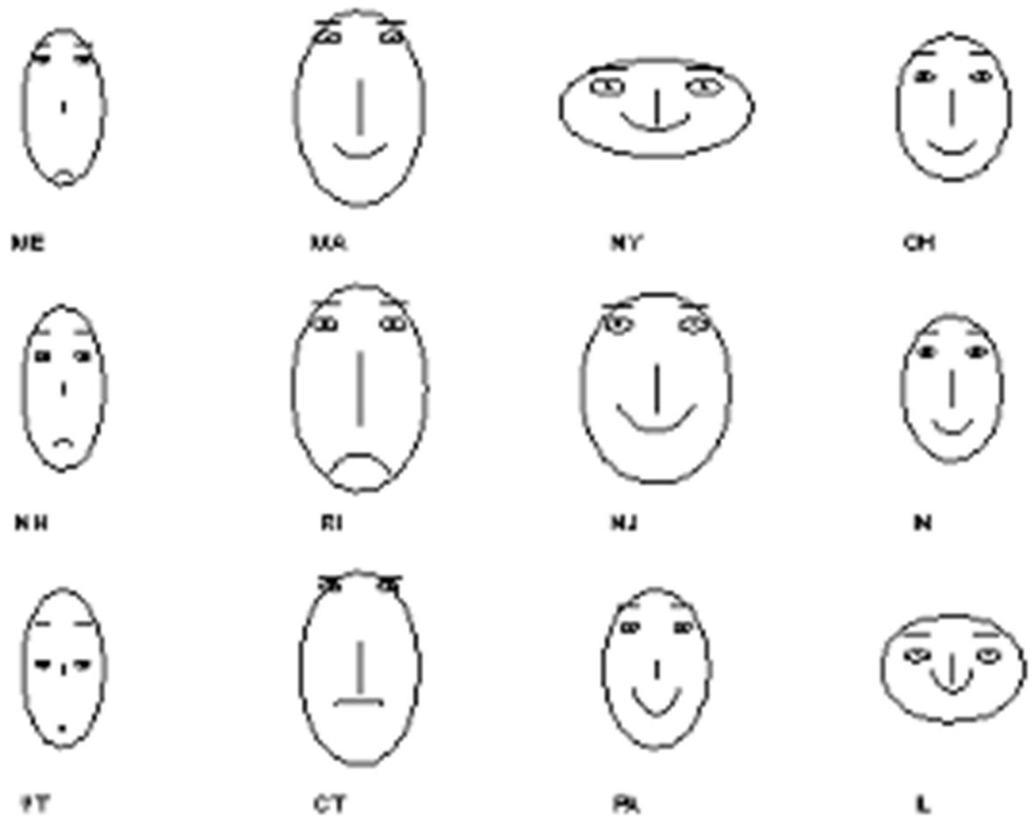


Chernoff Faces

Encode different variables' values in characteristics of human face

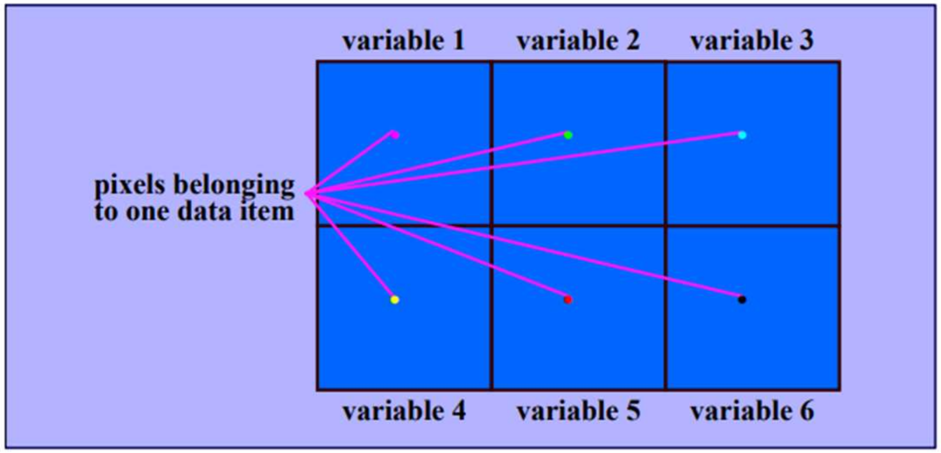


Chernoff Faces - Examples



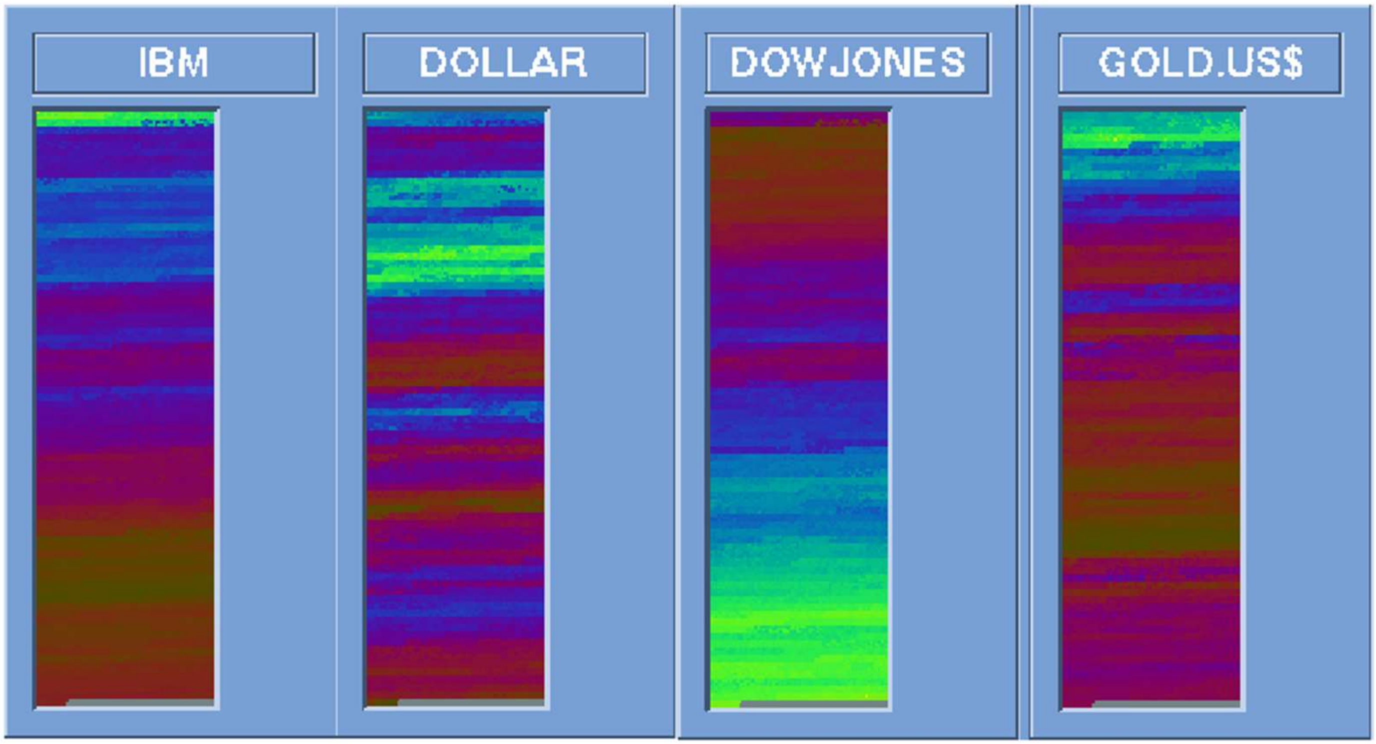
Pixel-based Methods

- Pixel-oriented visualization techniques map each attribute value of the data to a single colored pixel, yielding the display of the most possible information at a time
- Maintain the global view of large amounts of data while still preserving the perception of small regions of interest
- Meaning derived from ordering



Pixel-Oriented Visualization Techniques for Exploring Very Large Data Bases Daniel A. Keim
Journal of Computational and Graphical Statistics
Vol. 5, No. 1 March, 1996

Pixel-based Display



Prices for 7 years
January ‘87 to March ’93
16,350 data items

Sand Dance

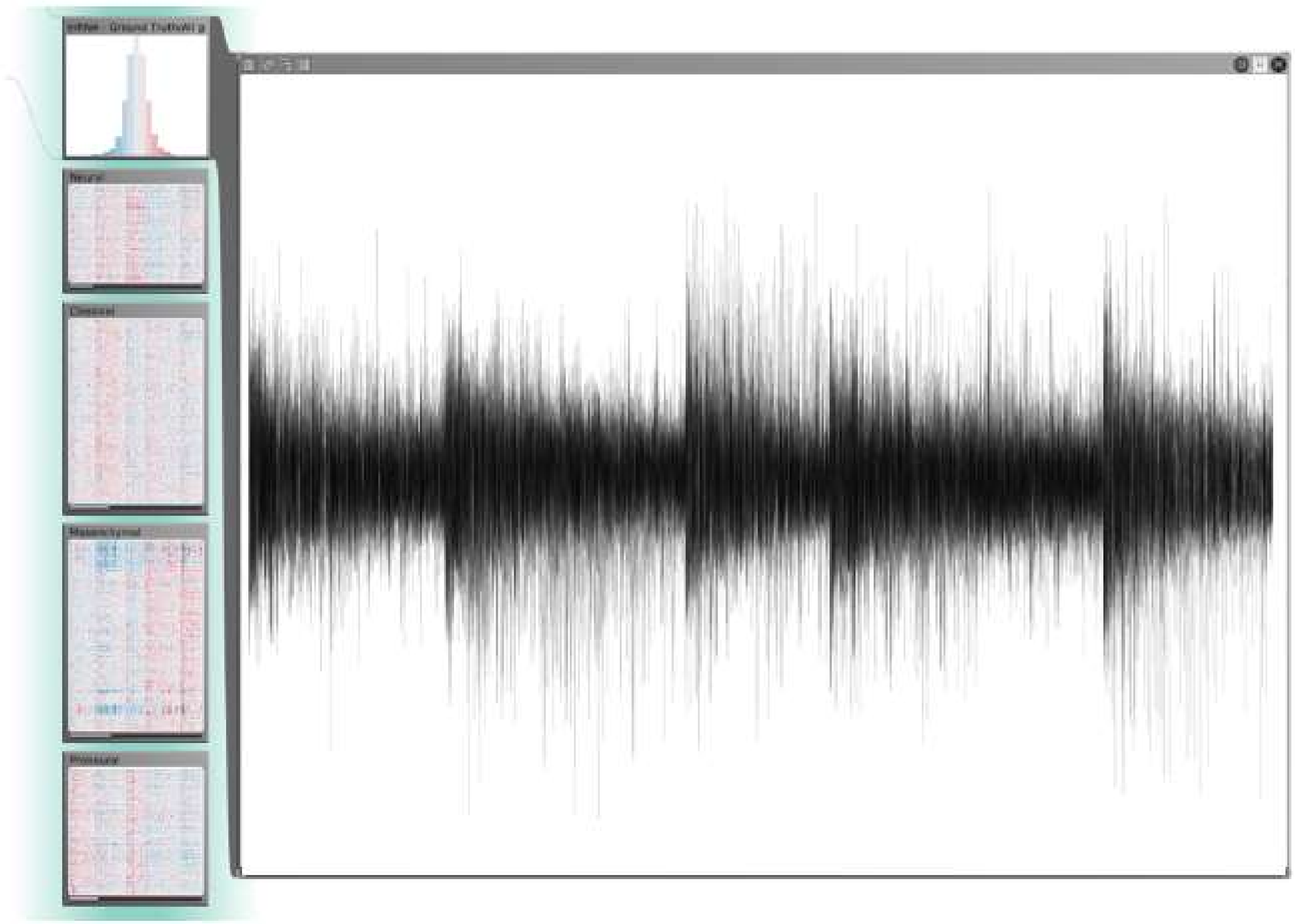
- Data items as small squares
- Can position and color based on different attributes
- Multiple layouts provided
- Slick animated transitions

https://sanddance.azurewebsites.net/BeachPartyApp/BeachPartyApp.html

Data Reduction

- **Sampling**
  - Don’t show every element, show a (random) subset
  - Efficient for large dataset
  - Apply only for display purposes
  - Outlier-preserving approaches
- **Filtering**
  - Define criteria to remove data, e.g., minimum variability
  - > / < / = specific value for one dimension
  - Can be interactive, combined with sampling
- **Clustering**
  - Classification of items into “similar” bins
  - Based on similarity measures

Clustered Heat Map



Dimensionality Reduction

- Reduce high dimensional to lower dimensional space
- Preserve as much of variation as possible
- Plot lower dimensional space
- Techniques:
  - Principal Component Analysis (PCA)
  - Multidimensional Scaling
  - tSNE

Principal Component Analysis (PCA)

- Principal Component Analysis (PCA) is a **linear dimensionality reduction** technique that can be utilized for extracting information from a high-dimensional space by projecting it into a lower-dimensional sub-space.
- It tries to preserve the essential parts that have more variation of the data and remove the non-essential parts with fewer variation.
- When the data is projected into a lower dimension (assume three dimensions) from a higher space, the lower dimensions are nothing but the Principal Components that captures most of the variance of the data.
- Principal components have both direction and magnitude. The direction represents across which *principal axes* the data is mostly spread out or has most variance and the magnitude signifies the amount of variance that Principal Component captures of the data when projected onto that axis.
- The principal components are a straight line, and the first principal component holds the most variance in the data.
- Each subsequent principal component is orthogonal to the last and has a lesser variance.

Multidimensional Scaling

- MDS is a non-linear technique for embedding data in a lower-dimensional space
- It maps points residing in a higher-dimensional space to a lower-dimensional space while preserving the distances between those points as much as possible.
- Because of this, the pairwise distances between points in the lower-dimensional space are matched closely to their actual distances.

t-SNE

- **t-distributed stochastic neighbor embedding** (**t-SNE**) is a statistical method for visualizing high-dimensional data by giving each datapoint a location in a two or three-dimensional map.
- It is a nonlinear dimensionality reduction technique for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions.
- Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modeled by nearby points and dissimilar objects are modeled by distant points with high probability.

t-SNE Algorithm

The t-SNE algorithm comprises the following stages:

- t-SNE models a point being selected as a neighbor of another point in both higher and lower dimensions.

- It starts by calculating a pairwise similarity between all data points in the high-dimensional space using a Gaussian kernel. The points that are far apart have a lower probability of being picked than the points that are close together.

- Then, the algorithm tries to map higher dimensional data points onto lower dimensional space while preserving the pairwise similarities.

- It is achieved by minimizing the divergence between the probability distribution of the original high-dimensional and lower-dimensional. The algorithm uses gradient descent to minimize the divergence. The lower-dimensional embedding is optimized to a stable state.

Reading

- R. Rao and S.K. Card, **The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabluar Information**, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ACM CHI 1994.* https://www.researchgate.net/publication/2541647_The_Table_Lens_Merging_Graphical_and_Symbolic_Representations_in_an_Interactive_FocusContext_Visualization_for_Tabular_Information

- L. Van der Maaten and G. Hinton, **Visualizing data using t-SNE**, Journal of machine learning research, vol. 9, no. 11, 2008. https://www.cs.toronto.edu/~hinton/absps/tsne.pdf