Data Mining 2025

Data Preprocessing

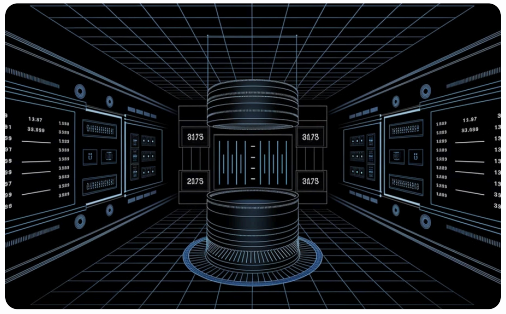
Dept. of Computer Science and Information Engineering

National Cheng Kung University

Kun-Ta Chuang

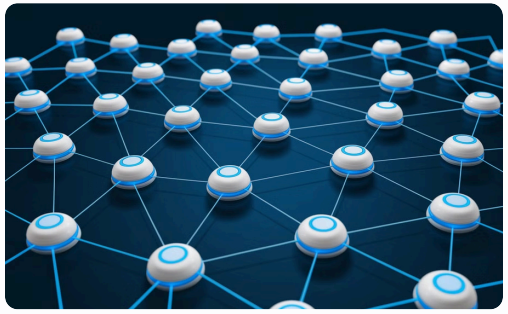
ktchuang@mail.ncku.edu.tw

Types of Data Sets



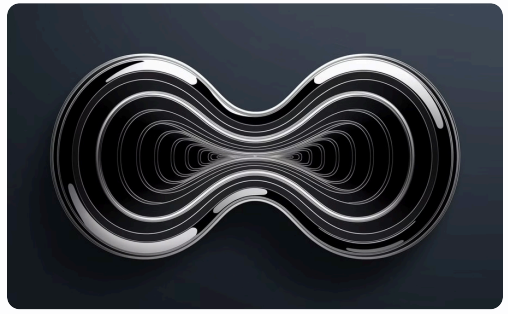
Record Data

- relational records
- data matrices
- document term-frequency vectors
- transaction data records



Graph and Network Data

Encompasses World Wide Web structures, social networks, information networks, and molecular structures



Ordered Data

Contains video sequences, temporal data, transaction sequences, and genetic sequence data

Spatial and Multimedia Data

Covers spatial data maps, image data, and video data with geographic components

Important Characteristics of Structured Data

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

- Dimensionality
  - Curse of dimensionality
- Sparsity
  - Only presence counts
- Resolution
  - Patterns depend on the scale
- Distribution
  - Centrality and dispersion

Data Objects

- Data sets are made up of data objects.
- A **data object** represents an entity.
- Examples:
  - sales database: customers, store items, sales
  - medical database: patients, treatments
  - university database: students, professors, courses
- Also called *samples, examples, instances, data points, objects, tuples*.
- Data objects are described by **attributes**.
- Database rows -> data objects; columns ->attributes.

Attributes

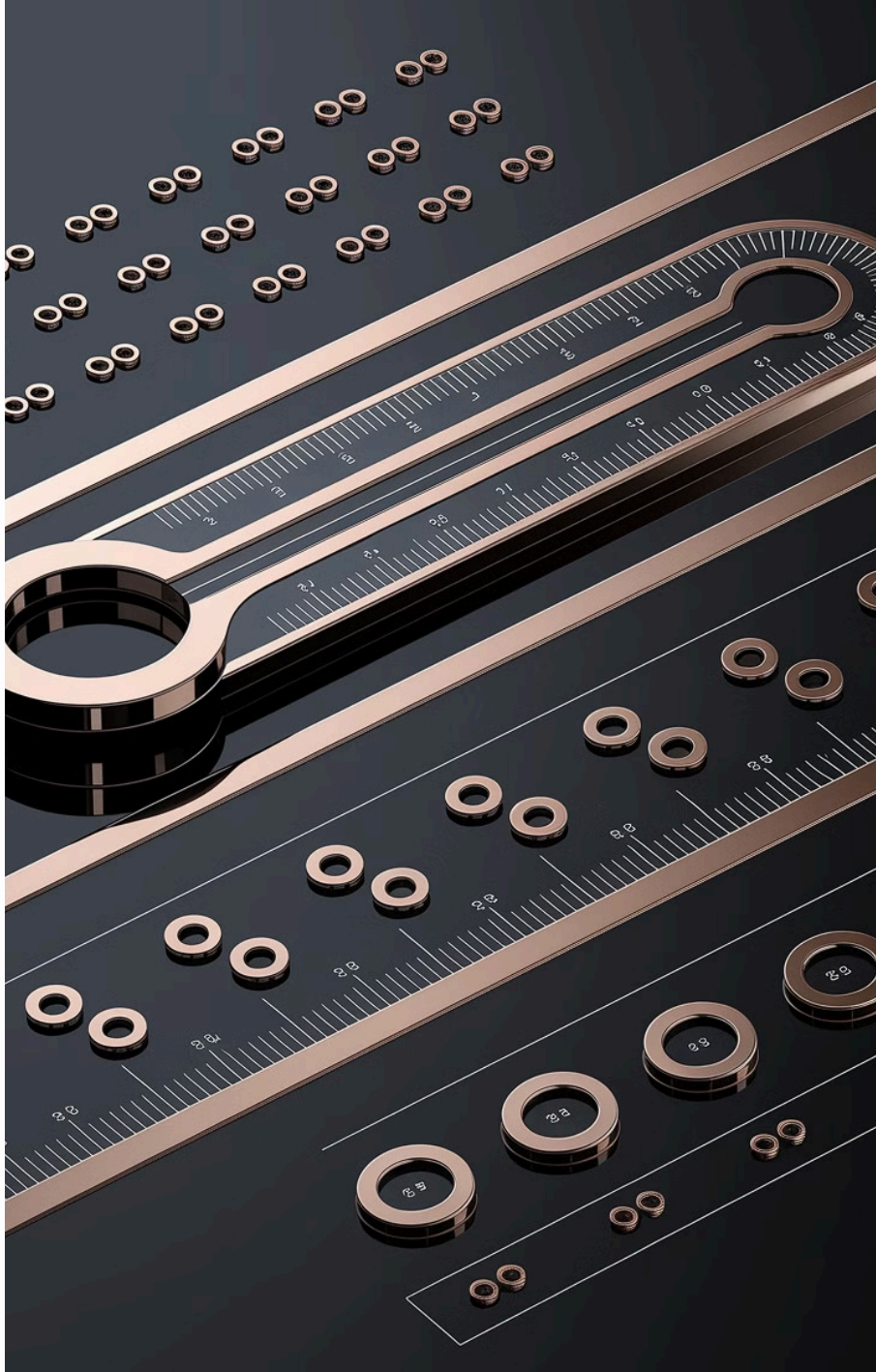
- **Attribute** (or **dimensions, features, variables**): a data field, representing a characteristic or feature of a data object.
  - E.g., customer _ID, name, address
- Types:
  - Nominal
  - Binary
  - Numeric: quantitative
    - Interval-scaled
    - Ratio-scaled

Attribute Types

- **Nominal:** categories, states, or "names of things"
  - *Hair_color = {auburn, black, blond, brown, grey, red, white}*
  - marital status, occupation, ID numbers, zip codes
- Binary
  - Nominal attribute with only 2 states (0 and 1)
  - Symmetric binary: both outcomes equally important
    - e.g., gender
  - Asymmetric binary: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
    - Convention: assign **1 to most important outcome** (e.g., HIV positive)
- Ordinal
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size = {small, medium, large}*, grades, army rankings

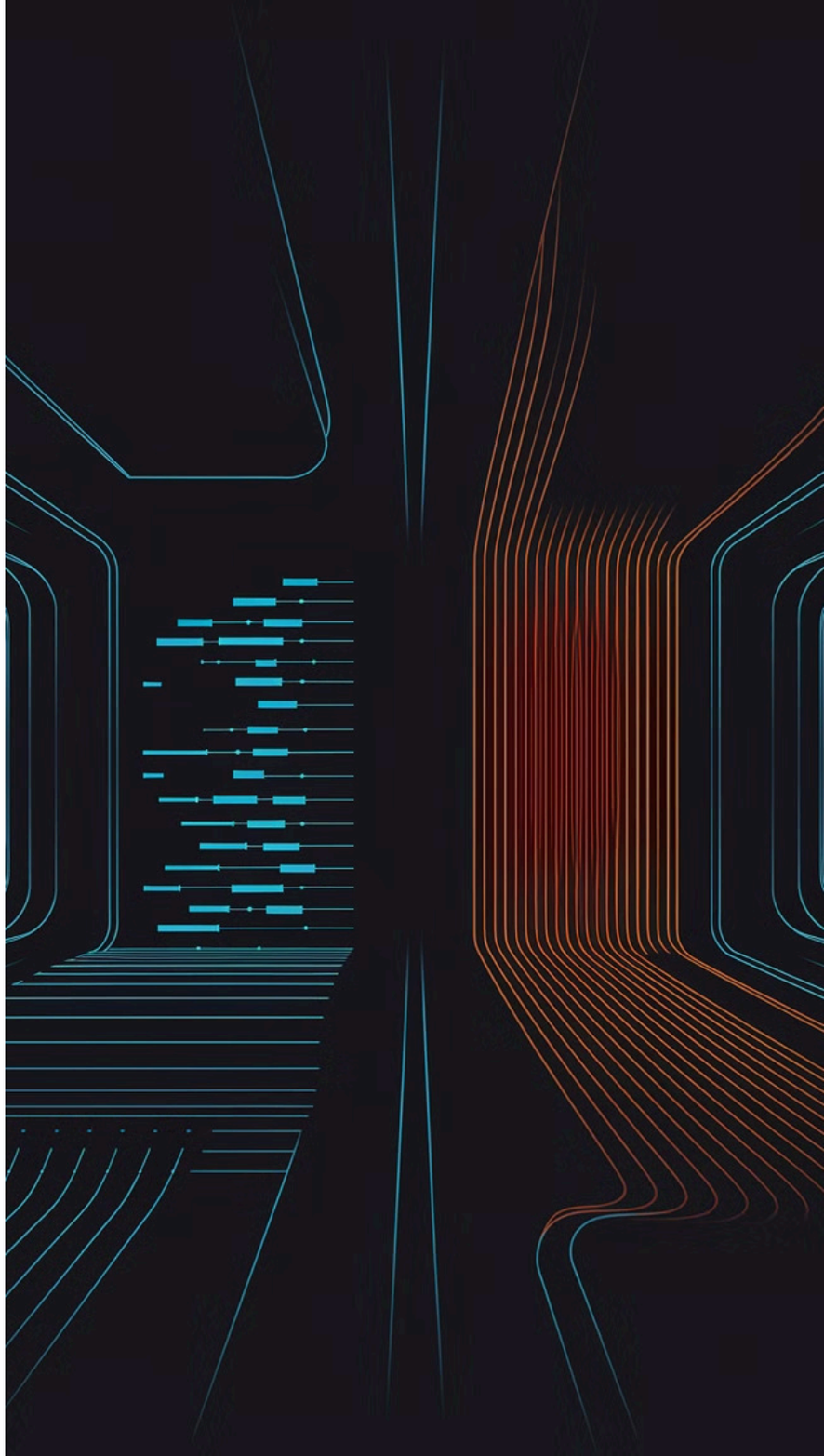
Numeric Attribute Types

- Quantity (integer or real-valued)
- Interval
  - Measured on a scale of **equal-sized units**
  - Values have order
    - E.g., *temperature in C° or F°, calendar dates*
  - No true zero-point
- Ratio
  - Inherent **zero-point**
  - We can speak of values as being an order of magnitude larger than the unit of measurement (10 K° is twice as high as 5 K°).
    - e.g., *temperature in Kelvin, length, counts, monetary quantities*



Discrete vs. Continuous Attributes

- **Discrete Attribute**
  - Has only a finite or countably infinite set of values
    - E.g., zip codes, profession, or the set of words in a collection of documents
  - Sometimes, represented as integer variables
  - Note: Binary attributes are a special case of discrete attributes
- **Continuous Attribute**
  - Has real numbers as attribute values
    - E.g., temperature, height, or weight
  - Practically, real values can only be measured and represented using a finite number of digits
  - Continuous attributes are typically represented as floating-point variables



Basic Statistical Descriptions of Data

- **Motivation**
  - To better understand the data: central tendency, variation and spread
- **Data dispersion characteristics**
  - median, max, min, quantiles, outliers, variance, etc.
- **Numerical dimensions** correspond to sorted intervals
  - Data dispersion: analyzed with multiple granularities of precision
  - Boxplot or quantile analysis on sorted intervals
- **Dispersion analysis on computed measures**
  - Folding measures into numerical dimensions
  - Boxplot or quantile analysis on the transformed cube

Measuring the Central Tendency

- **Mean (algebraic measure) (sample vs. population):**

Note: *n* is sample size and *N* is population size.

- Weighted arithmetic mean:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \mu = \frac{\sum x}{N}$$

- Trimmed mean: chopping extreme values

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

age	*frequency*
1–5	200
6–15	450
16–20	300
21–50	1500
51–80	700
81–110	44

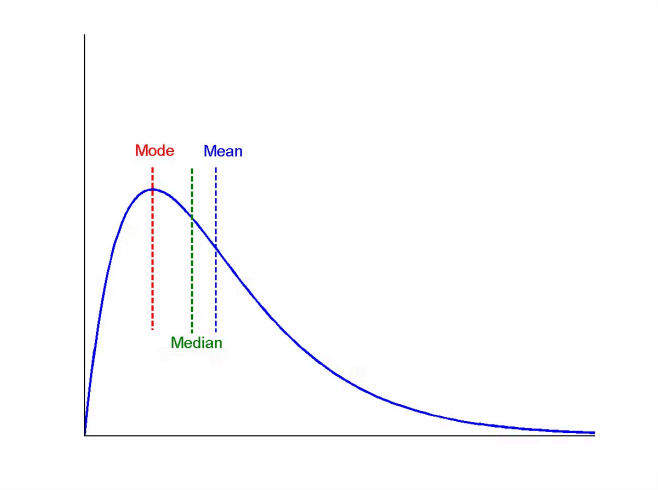
- **Median:**
- Middle value if odd number of values, or average of the middle two values otherwise
- Estimated by interpolation (for *grouped data*):

$$median = L_1 + (\frac{n/2 - (\sum freq)l}{freq_{median}})width$$

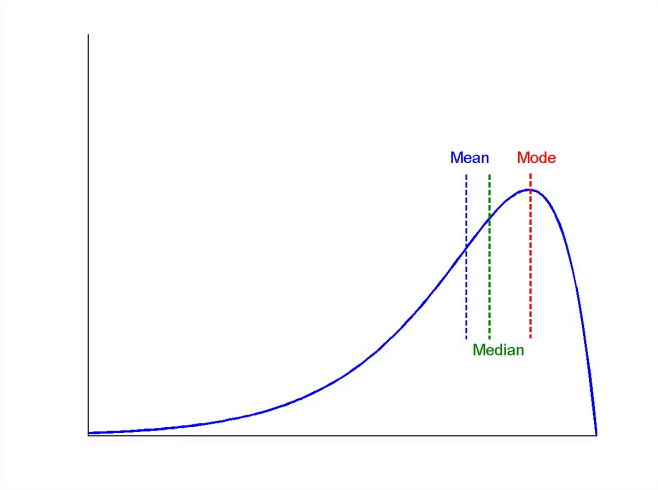
- **Mode**
- Value that occurs most frequently in the data
- Unimodal, bimodal, trimodal
- Empirical formula:

Symmetric vs. Skewed Data

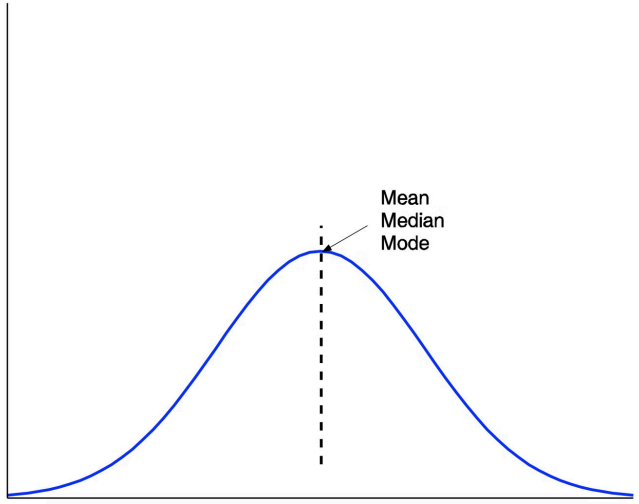
Median, mean and mode of symmetric, positively and negatively skewed data



positively skewed



negatively skewed



Measuring the Dispersion of Data

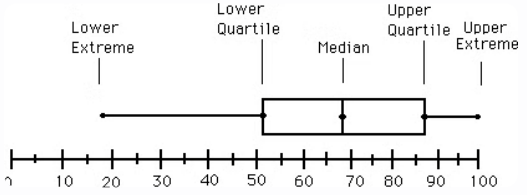
- Quartiles, outliers and boxplots
  - **Quartiles**: Q_1 (25th percentile), Q_3 (75th percentile)
  - **Inter-quartile range**: IQR = $Q_3 - Q_1$
  - **Five number summary**: min, Q_1, median, Q_3, max
  - **Boxplot**: ends of the box are the quartiles; median is marked; add whiskers, and plot outliers individually
  - **Outlier**: usually, a value higher/lower than 1.5 x IQR
- Variance and standard deviation (*sample: s, population: σ*)
  - **Variance**: (algebraic, scalable computation)
  - **Standard deviation** *s (or σ)* is the square root of variance *s² (or σ²)*

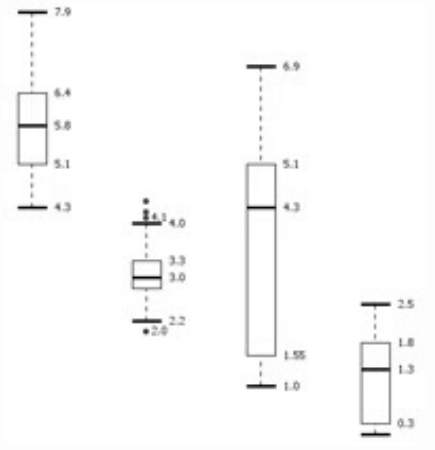
$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}[\sum_{i=1}^{n}x_i^2 - \frac{1}{n}(\sum_{i=1}^{n}x_i)^2] \qquad \sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2 = \frac{1}{N}\sum_{i=1}^{n}x_i^2 - \mu^2$$

  -

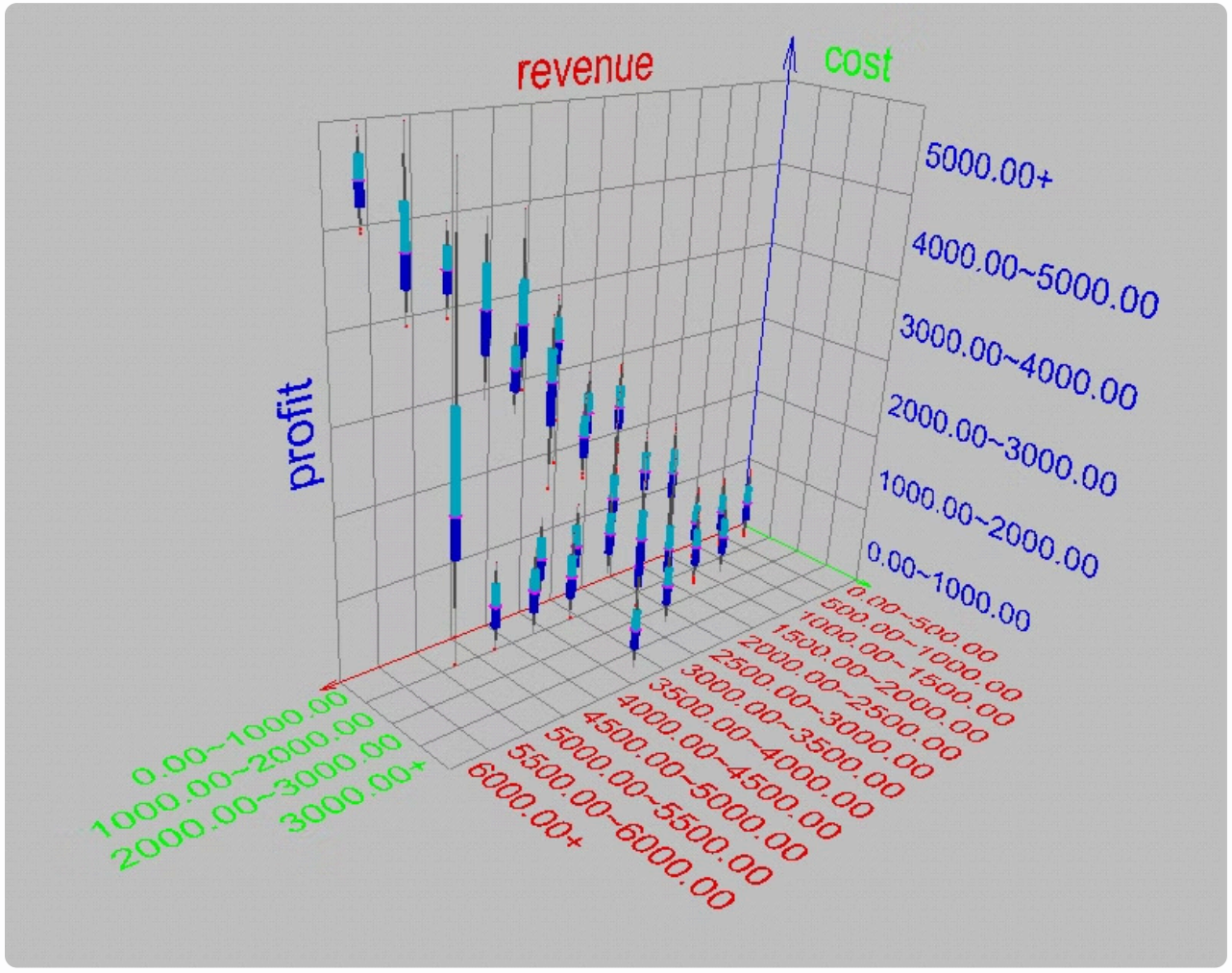
Boxplot Analysis

- **Five-number summary** of a distribution
  - Minimum, Q1, Median, Q3, Maximum
- Boxplot
  - Data is represented with a box
  - The ends of the box are at the first and third quartiles, i.e., the height of the box is IQR
  - The median is marked by a line within the box
  - Whiskers: two lines outside the box extended to Minimum and Maximum
  - Outliers: points beyond a specified outlier threshold, plotted individually



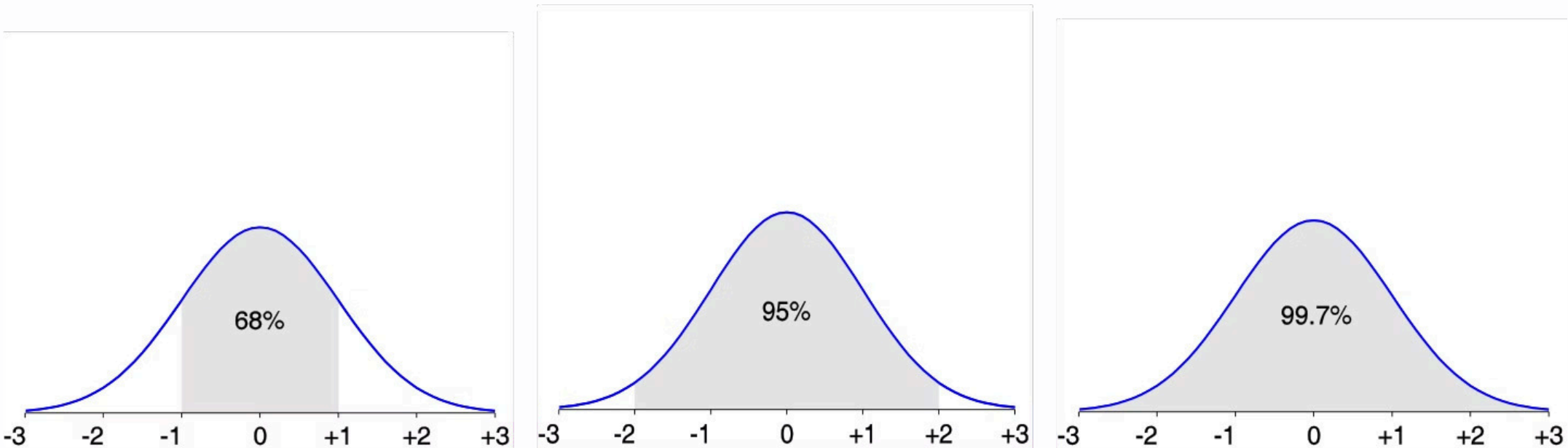


Visualization of Data Dispersion: 3-D Boxplots



Properties of Normal Distribution Curve

- The normal (distribution) curve
  - From μ−σ to μ+σ: contains about 68% of the measurements (μ: mean, σ: standard deviation)
  - From μ−2σ to μ+2σ: contains about 95% of it
  - From μ−3σ to μ+3σ: contains about 99.7% of it



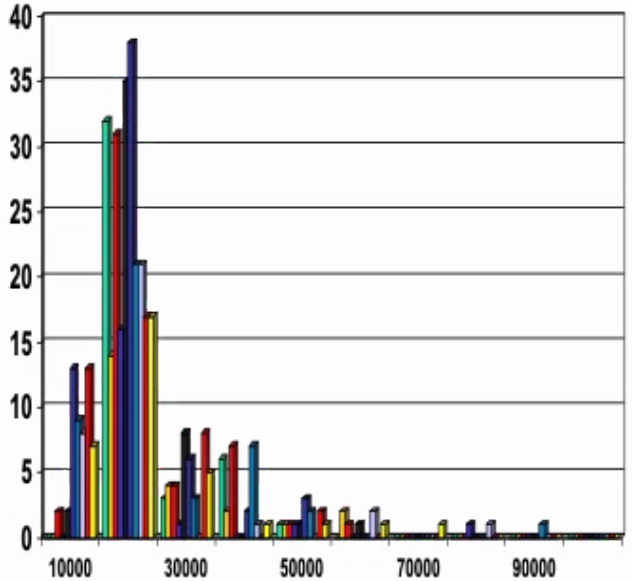
  -

Graphic Displays of Basic Statistical Descriptions

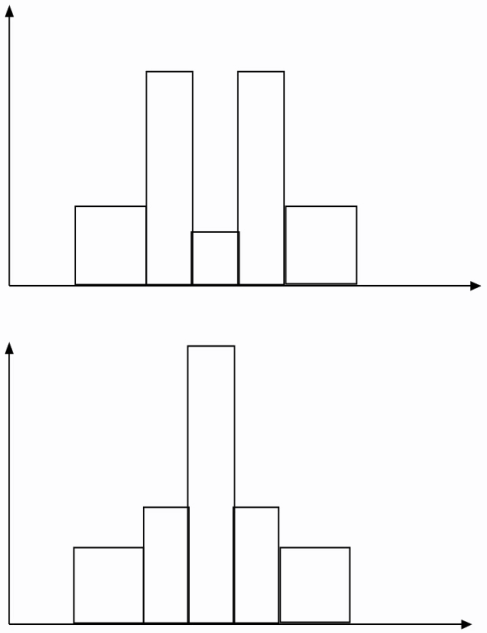
- **Boxplot**: graphic display of five-number summary
- **Histogram**: x-axis are values, y-axis repres. frequencies
- **Quantile plot**: each value x*i* is paired with f*i* indicating that approximately 100 f_*i*% of data are ≤ x_*i*
- **Quantile-quantile (q-q) plot**: graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- **Scatter plot**: each pair of values is a pair of coordinates and plotted as points in the plane

Histogram Analysis

- Histogram: Graph display of tabulated frequencies, shown as bars
- It shows what proportion of cases fall into each of several categories
- Differs from a bar chart in that it is the *area* of the bar that denotes the value, not the height as in bar charts, a crucial distinction when the categories are not of uniform width
- The categories are usually specified as non-overlapping intervals of some variable. The categories (bars) must be adjacent



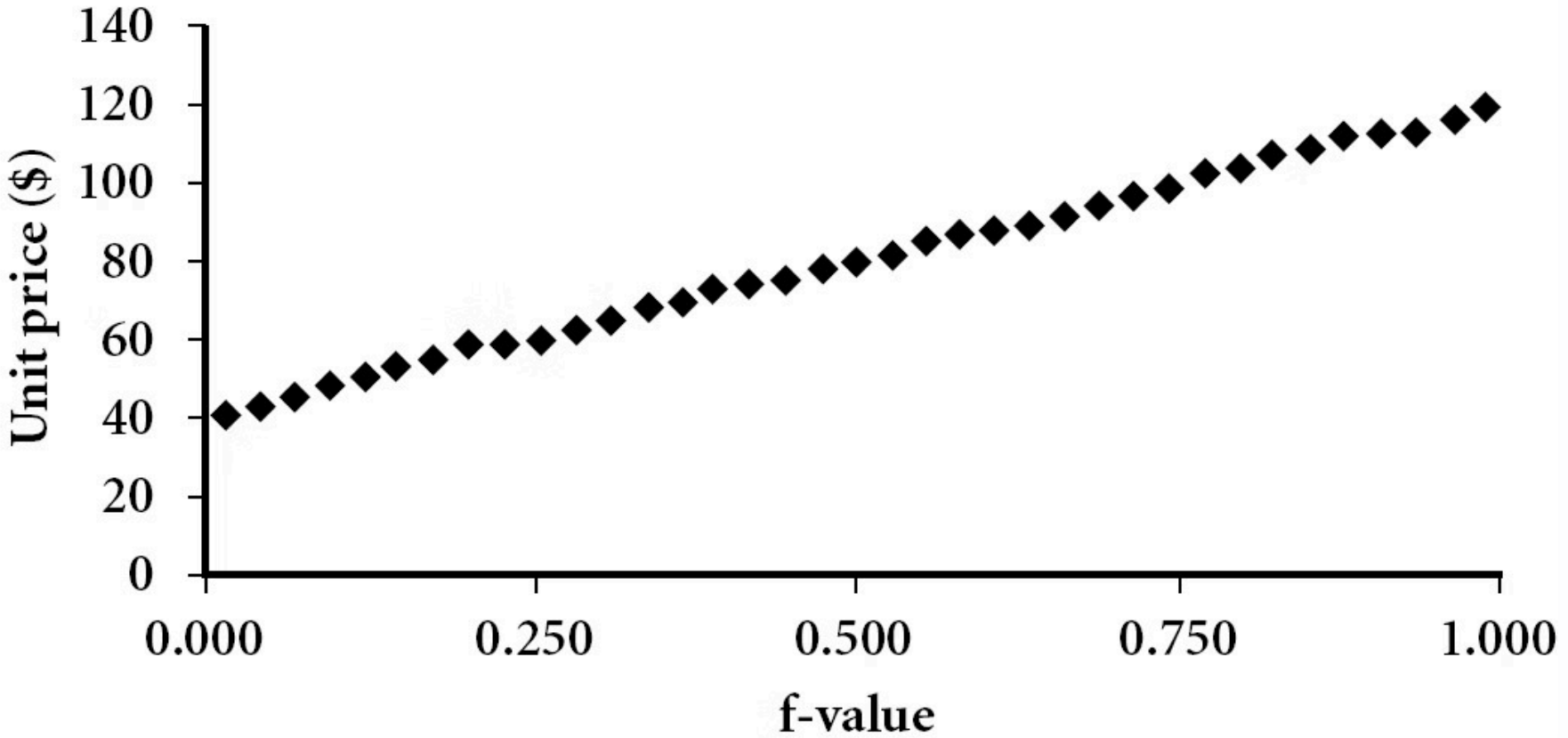
Histograms Often Tell More than Boxplots



- The two histograms shown in the left may have the same boxplot representation
  - The same values for: min, Q1, median, Q3, max
- But they have rather different data distributions

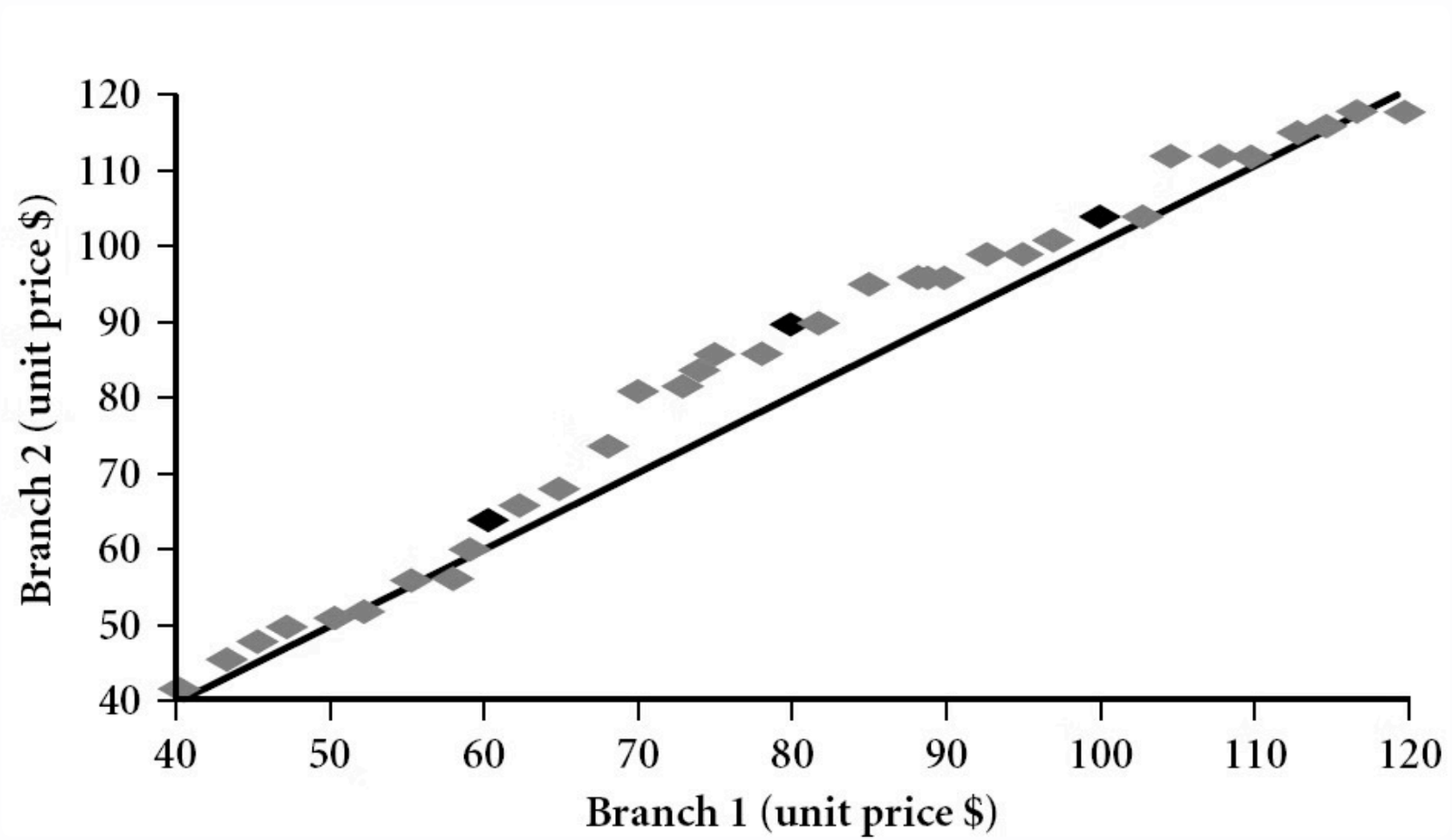
Quantile Plot

- Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)
- Plots **quantile** information
  - For a data *xi* data sorted in increasing order, *fi* indicates that approximately 100 *fi*% of the data are below or equal to the value *xi*

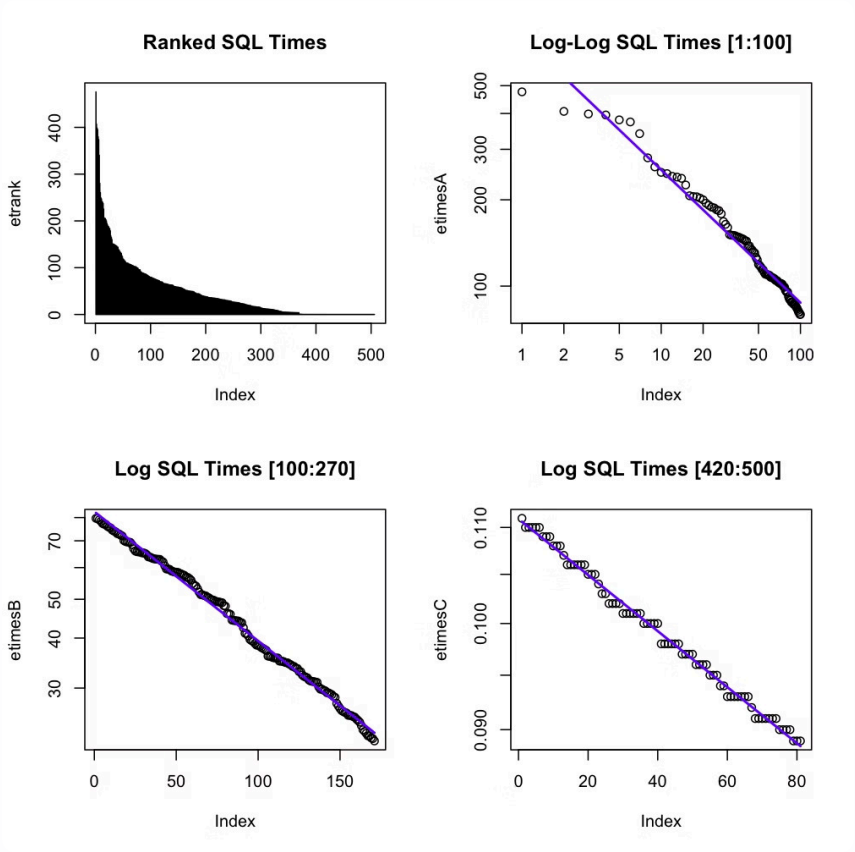


Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- View: Is there is a shift in going from one distribution to another?
- Example shows unit price of items sold at Branch 1 vs. Branch 2 for each quantile. Unit prices of items sold at Branch 1 tend to be lower than those at Branch 2.



Q-Q Plot and Power Laws

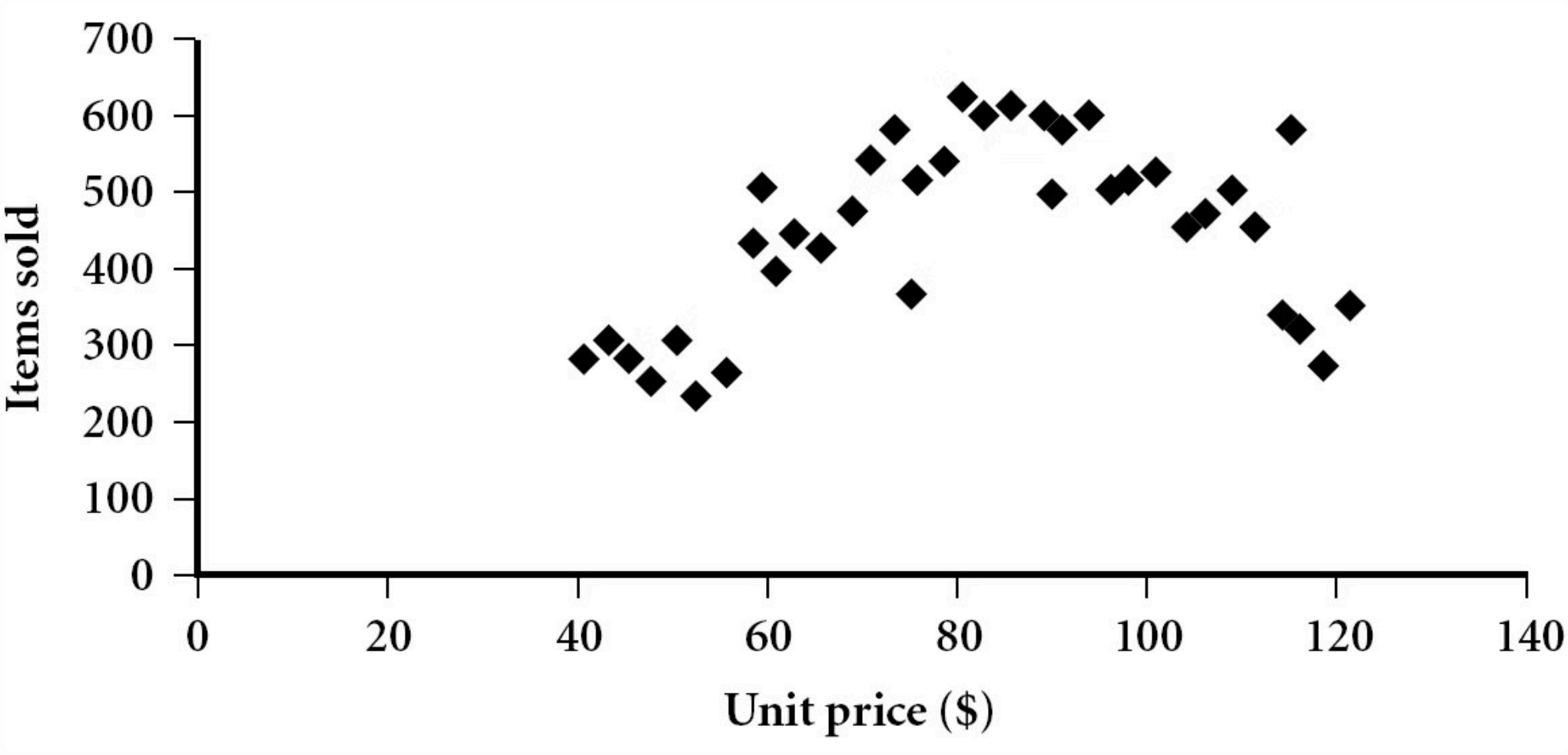


In the Q-Q plot, the data points should fall approximately along a straight line if the two distributions come from the same family. Deviations from the straight line indicate differences between the distributions being compared. Power laws describe relationships where a quantity varies as a power of another.

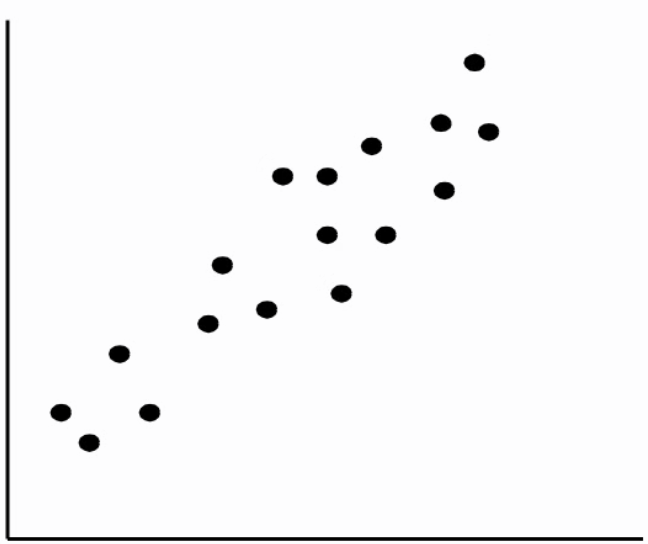
good ref to see the power-law insight for database performance: **https://perfdynamics.blogspot.com/2011/08/q-q-plots-for-multi-modal-performance.html**

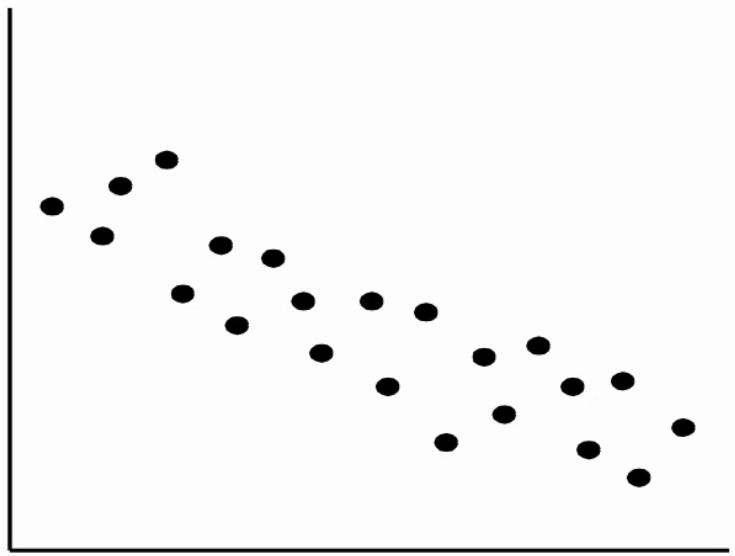
Scatter plot

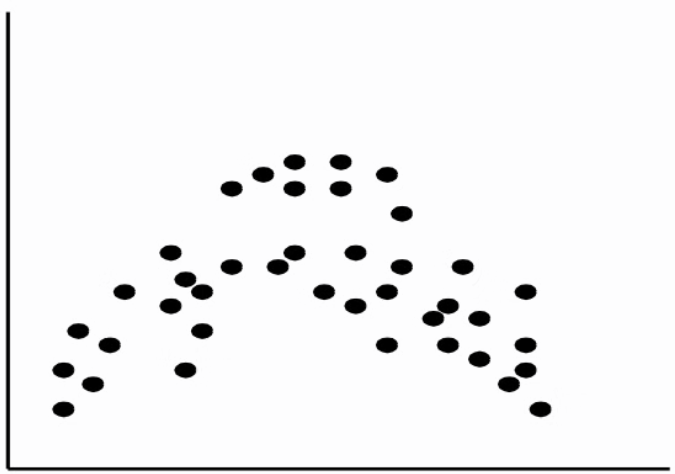
- Provides a first look at bivariate data to see clusters of points, outliers, etc
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane



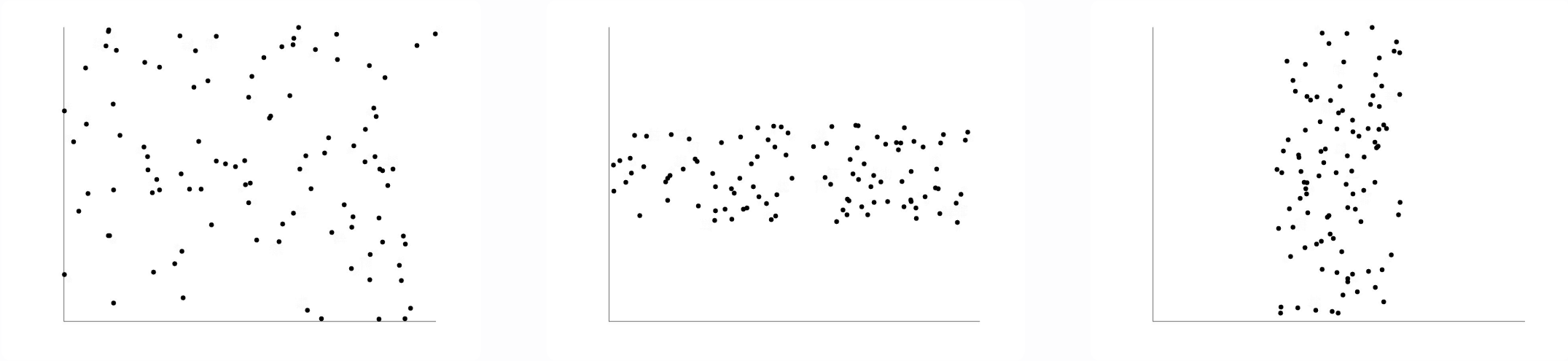
Positively and Negatively Correlated Data







Uncorrelated Data



Chapter 2: Getting to Know Your Data

- Data Objects and Attribute Types
- Basic Statistical Descriptions of Data
- Data Visualization
- Measuring Data Similarity and Dissimilarity
- Summary

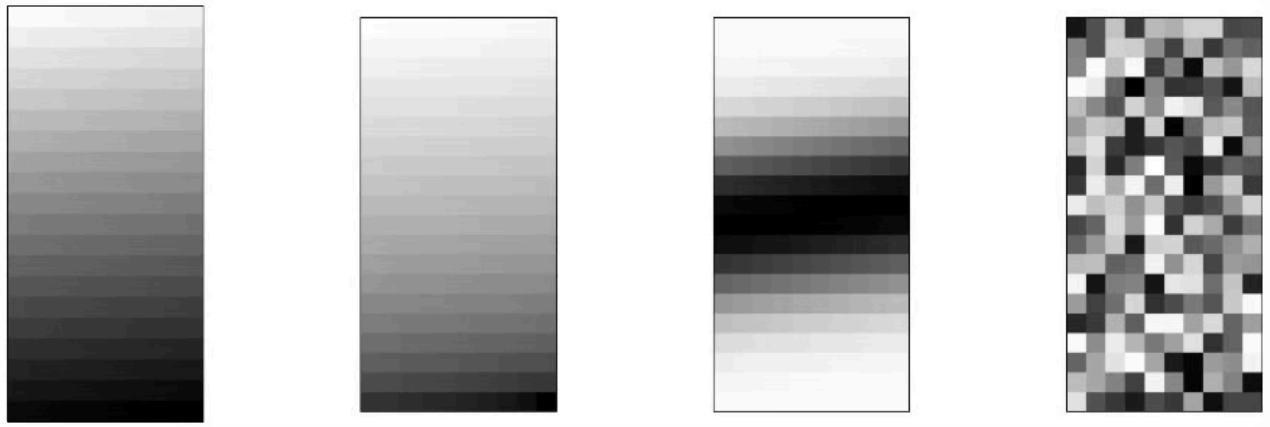
Data Visualization

- Why data visualization?
  - **Gain insight** into an information space by mapping data onto graphical primitives
  - **Provide qualitative overview** of large data sets
  - **Search** for patterns, trends, structure, irregularities, relationships among data
  - **Help find interesting regions and suitable parameters** for further quantitative analysis
  - **Provide a visual proof** of computer representations derived
- Categorization of visualization methods:
  - Pixel-oriented visualization techniques
  - Geometric projection visualization techniques
  - Icon-based visualization techniques
  - Hierarchical visualization techniques
  - Visualizing complex data and relations

Pixel-Oriented Visualization Techniques

- For a data set of m dimensions, create m windows on the screen, one for each dimension
- The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows
- The colors of the pixels reflect the corresponding values

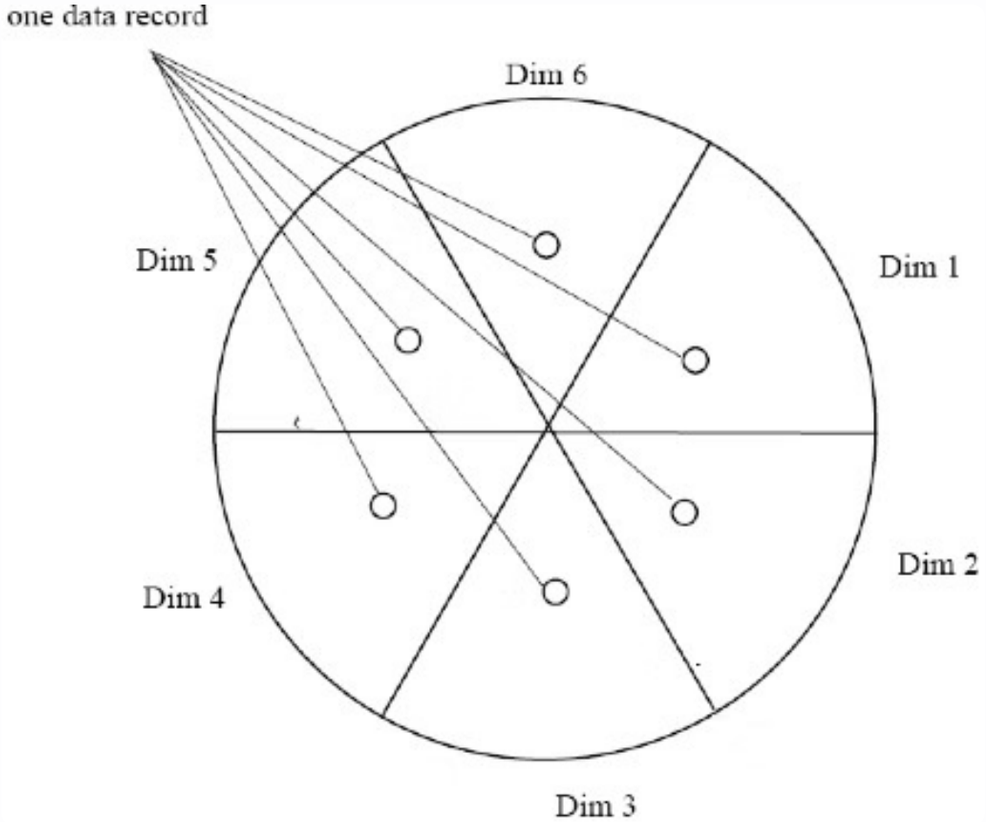
(a) Income (b) Credit Limit (c) transaction volume (d) age



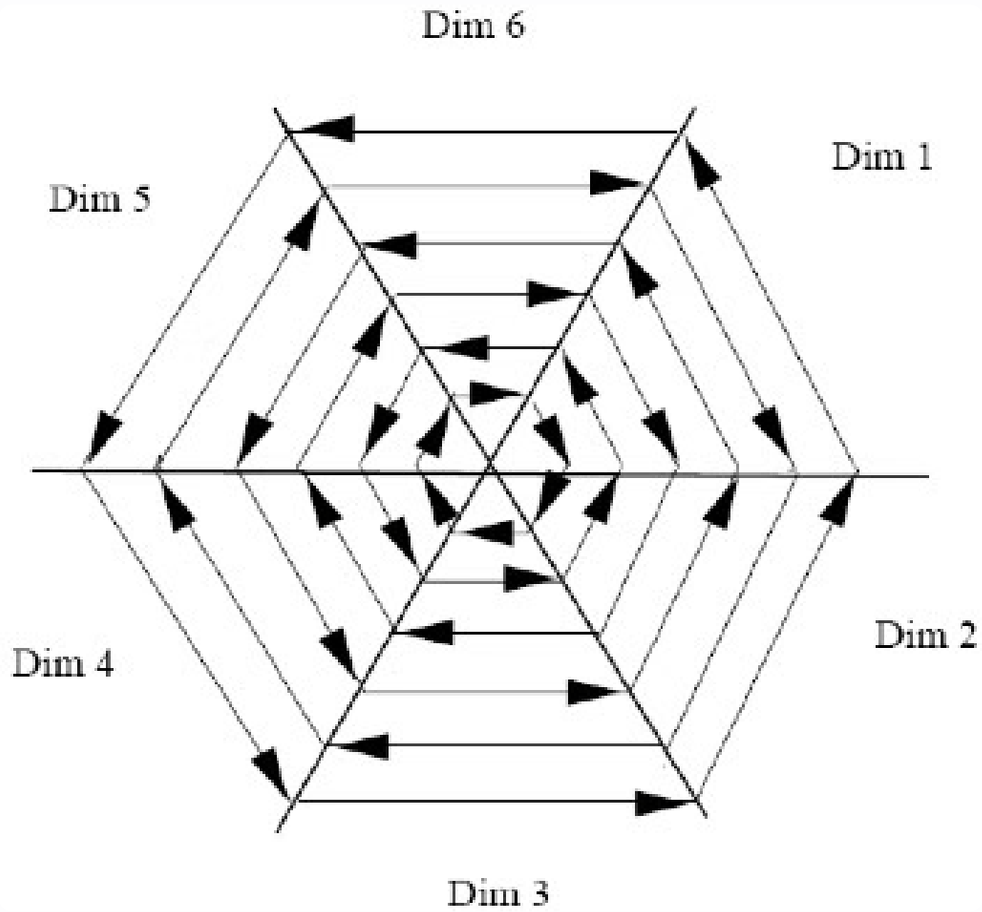
Laying Out Pixels in Circle Segments

- To save space and show the connections among multiple dimensions, space filling is often done in a circle segment

(a) Representing a data record in circle segment



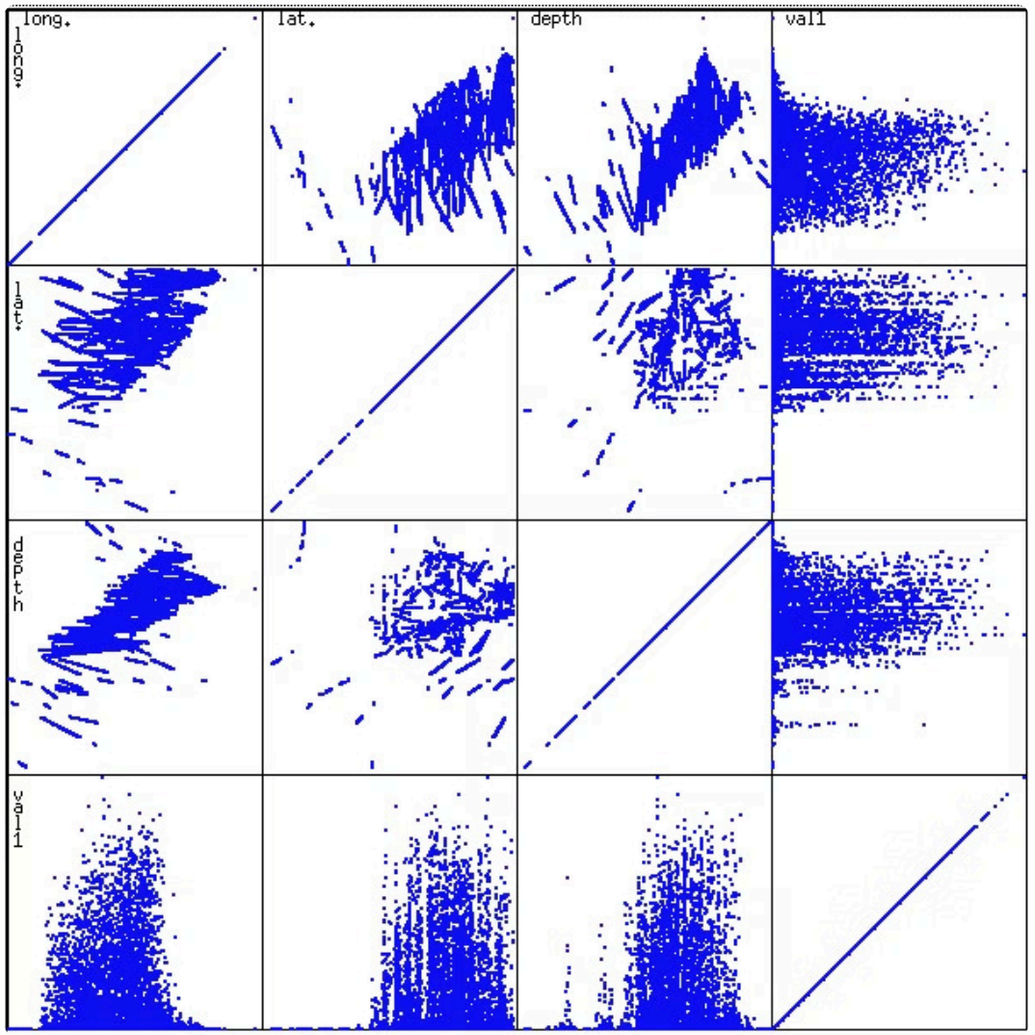
(b) Laying out pixels in circle segment



Geometric Projection Visualization Techniques

- Visualization of geometric transformations and projections of the data
- Methods
  - Direct visualization
  - Scatterplot and scatterplot matrices
  - Landscapes
  - Projection pursuit technique: Help users find meaningful projections of multidimensional data
  - Prosection views
  - Hyperslice
  - Parallel coordinates

Scatterplot Matrices



Matrix of scatterplots (x-y-diagrams) of the k-dim. data [total of (k2/2-k) scatterplots]

Parallel Coordinates

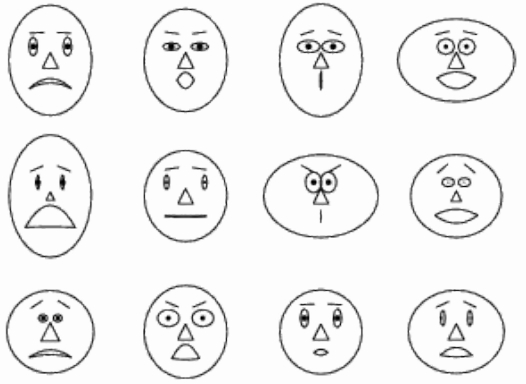
- n equidistant axes which are parallel to one of the screen axes and correspond to the attributes
- The axes are scaled to the [minimum, maximum]: range of the corresponding attribute
- Every data item corresponds to a polygonal line which intersects each of the axes at the point which corresponds to the value for the attribute

Icon-Based Visualization Techniques

- Visualization of the data values as features of icons
- Typical visualization methods
  - Chernoff Faces
  - Stick Figures
- General techniques
  - Shape coding: Use shape to represent certain information encoding
  - Color icons: Use color icons to encode more information
  - Tile bars: Use small icons to represent the relevant feature vectors in document retrieval

Chernoff Faces

- A way to display variables on a two-dimensional surface, e.g., let x be eyebrow slant, y be eye size, z be nose length, etc.
- The figure shows faces produced using 10 characteristics--head eccentricity, eye size, eye spacing, eye eccentricity, pupil size, eyebrow slant, nose size, mouth shape, mouth size, and mouth opening): Each assigned one of 10 possible values, generated using *Mathematica* (S. Dickson)

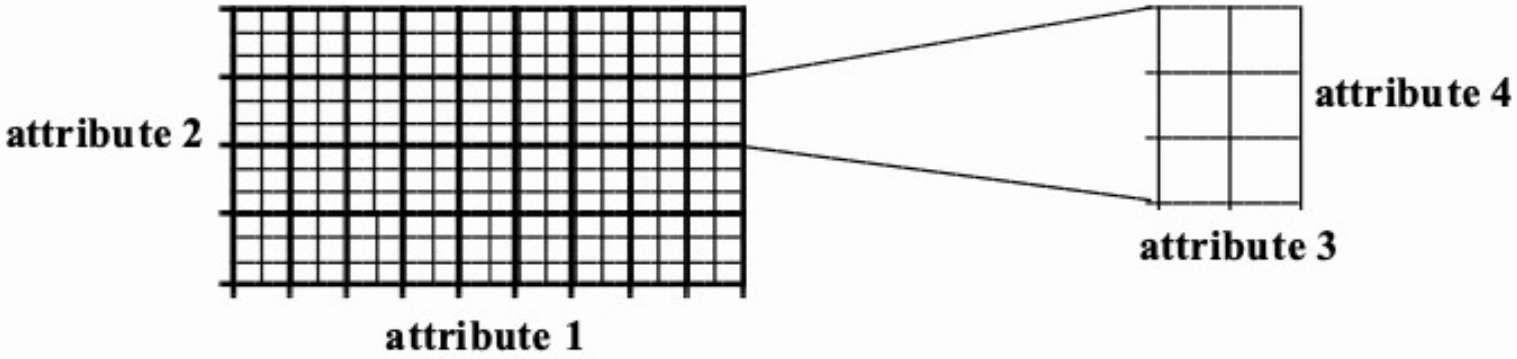


- REFERENCE: Gonick, L. and Smith, W. *The Cartoon Guide to Statistics.* New York: Harper Perennial, p. 212, 1993
- Weisstein, Eric W. "Chernoff Face." From *MathWorld*--A Wolfram Web Resource. **mathworld.wolfram.com/ChernoffFace.html**

Hierarchical Visualization Techniques

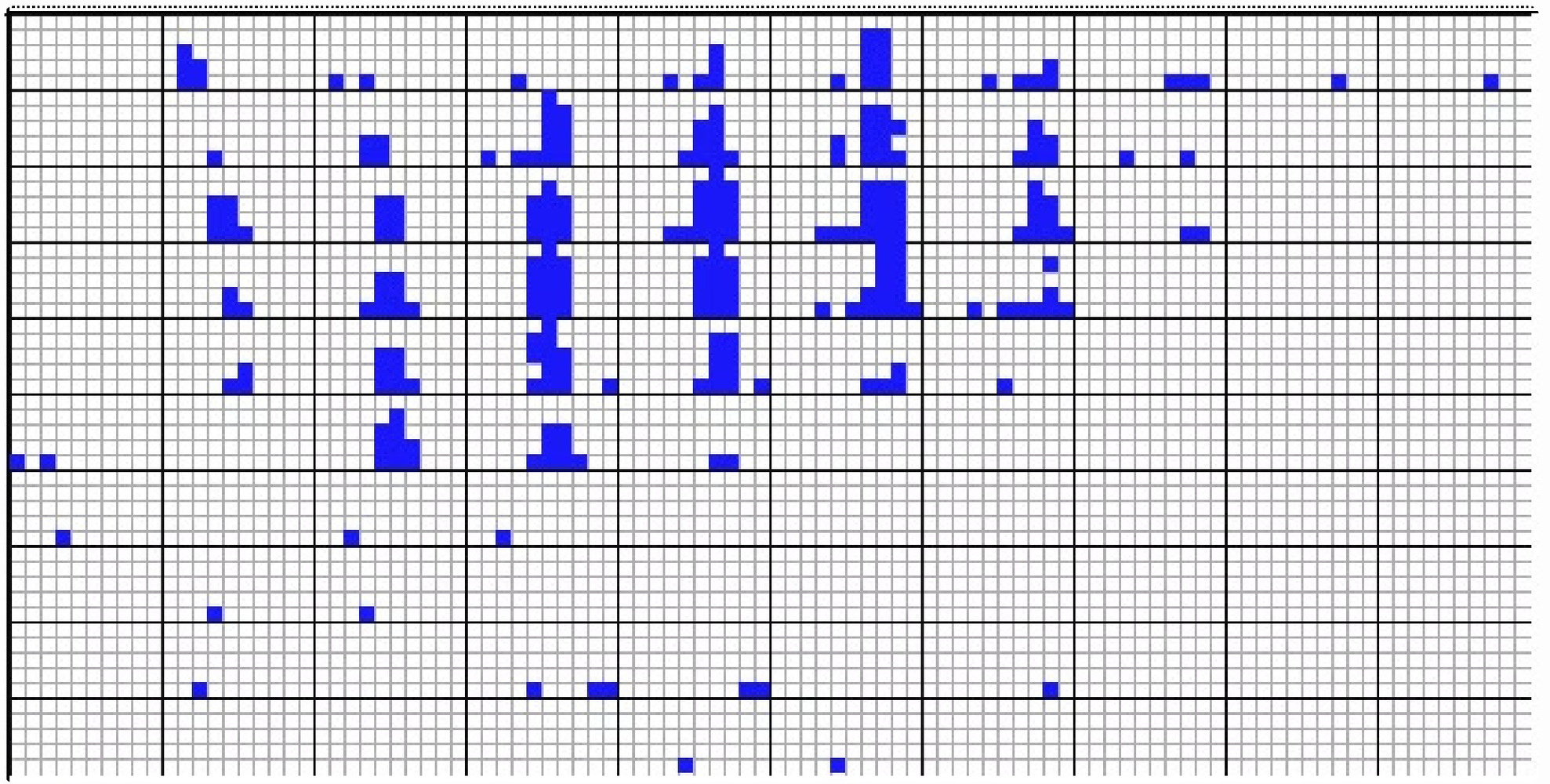
- Visualization of the data using a hierarchical partitioning into subspaces
- Methods
  - Dimensional Stacking
  - Worlds-within-Worlds
  - Tree-Map
  - Cone Trees
  - InfoCube

Dimensional Stacking



- Partitioning of the n-dimensional attribute space in 2-D subspaces, which are 'stacked' into each other
- Partitioning of the attribute value ranges into classes. The important attributes should be used on the outer levels.
- Adequate for data with ordinal attributes of low cardinality
- But, difficult to display more than nine dimensions
- Important to map dimensions appropriately

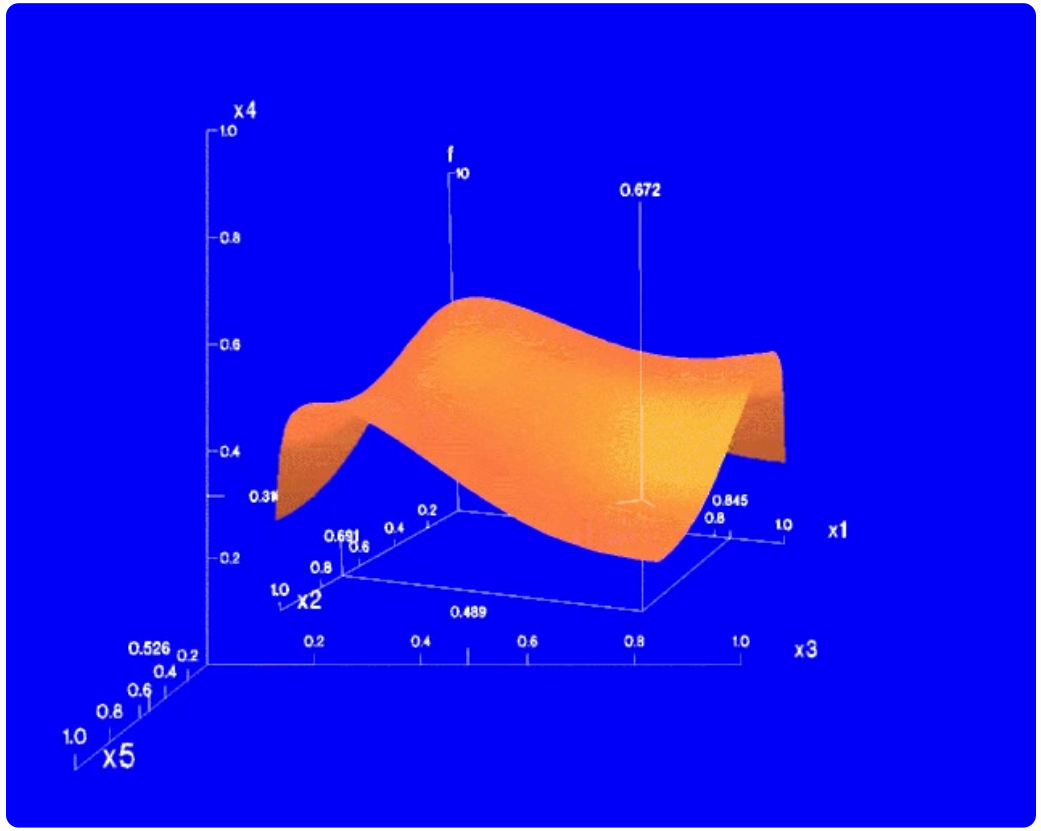
Dimensional Stacking



Visualization of oil mining data with longitude and latitude mapped to the outer x-, y-axes and ore grade and depth mapped to the inner x-, y-axes

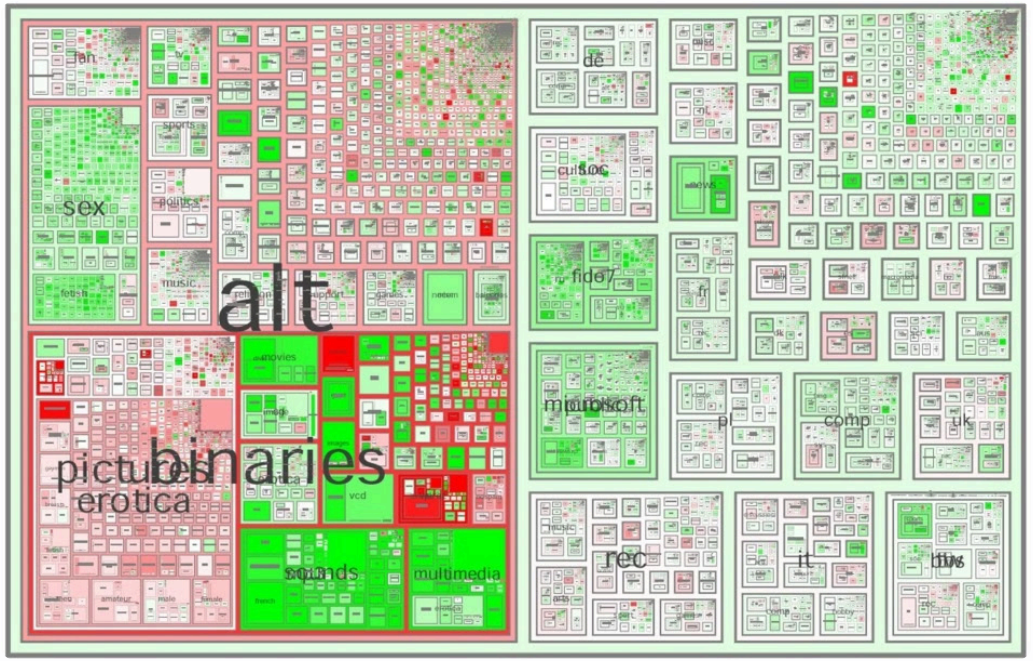
Worlds-within-Worlds

- Assign the function and two most important parameters to innermost world
- Fix all other parameters at constant values - draw other (1 or 2 or 3 dimensional worlds choosing these as the axes)
- Software that uses this paradigm
- N–vision: Dynamic interaction through data glove and stereo displays, including rotation, scaling (inner) and translation (inner/outer)
- Auto Visual: Static interaction by means of queries



Tree-Map

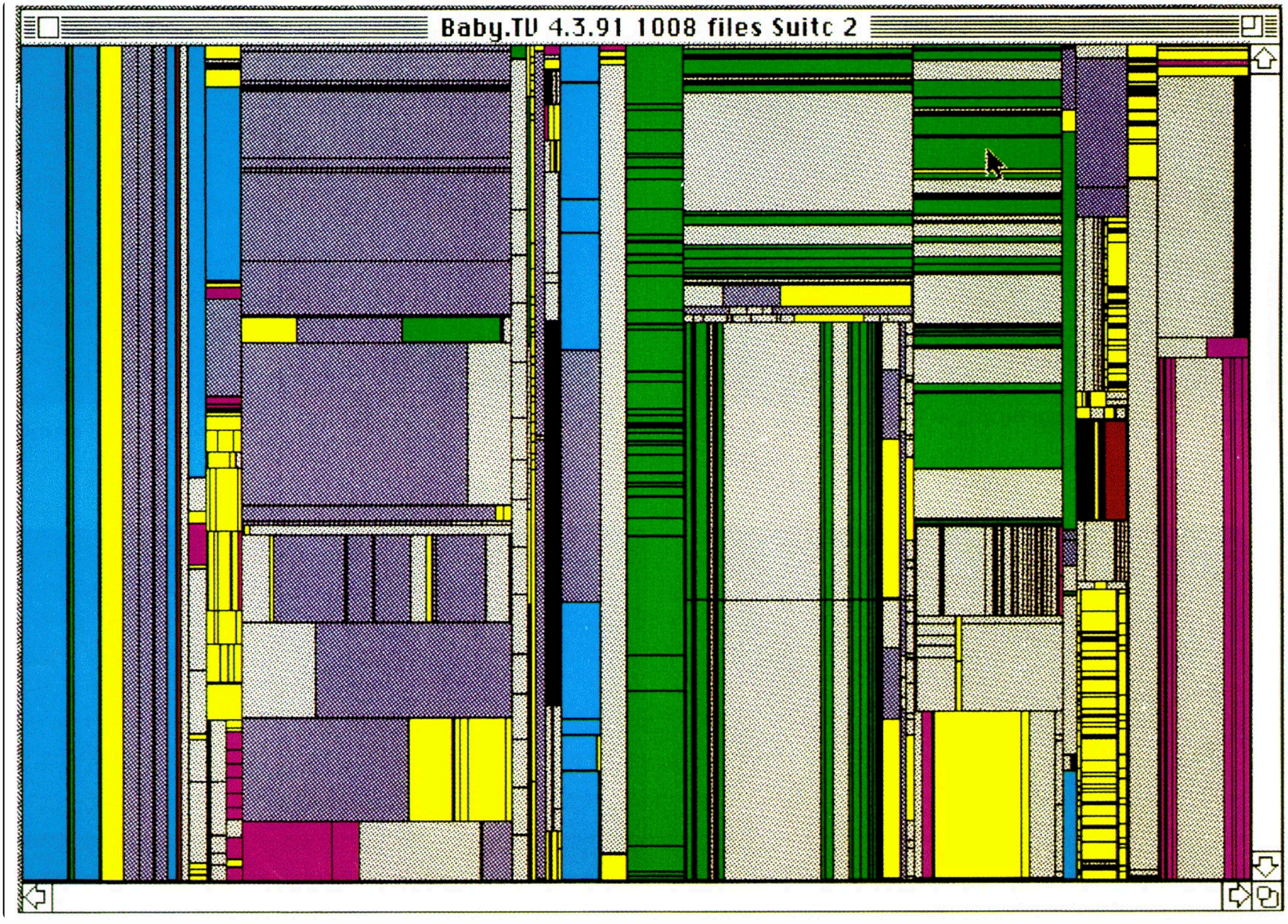
- Screen-filling method which uses a hierarchical partitioning of the screen into regions depending on the attribute values
- The x- and y-dimension of the screen are partitioned alternately according to the attribute values (classes)



MSR Netscan Image

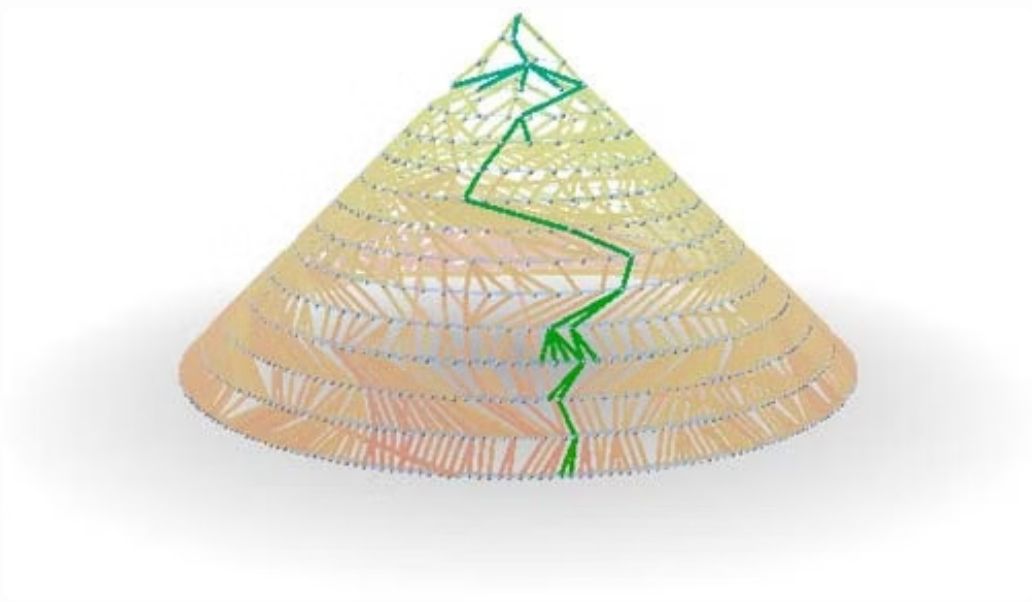
Ack.: http://www.cs.umd.edu/hcil/treemap-history/all102001.jpg

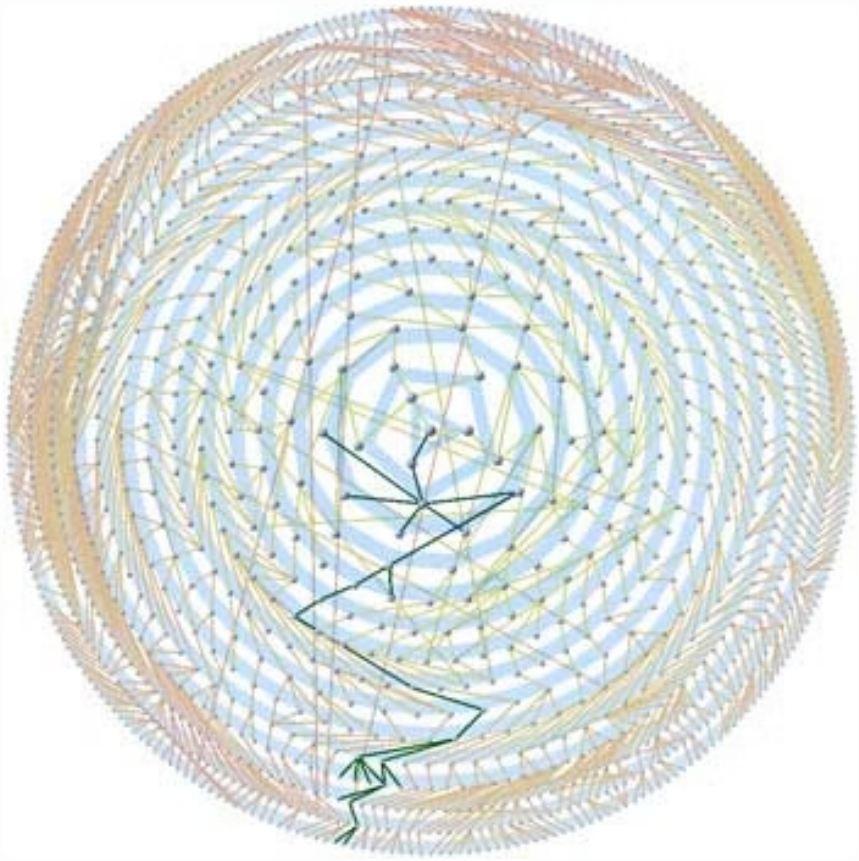
Tree-Map of a File System (Schneiderman)



Three-D Cone Trees

- *3D cone tree* visualization technique works well for up to a thousand nodes or so
- First build a *2D circle tree* that arranges its nodes in concentric circles centered on the root node
- Cannot avoid overlaps when projected to 2D
- G. Robertson, J. Mackinlay, S. Card. "Cone Trees: Animated 3D Visualizations of Hierarchical Information", *ACM SIGCHI'91*
- Graph from Nadeau Software Consulting website: Visualize a social network data set that models the way an infection spreads from one person to the next

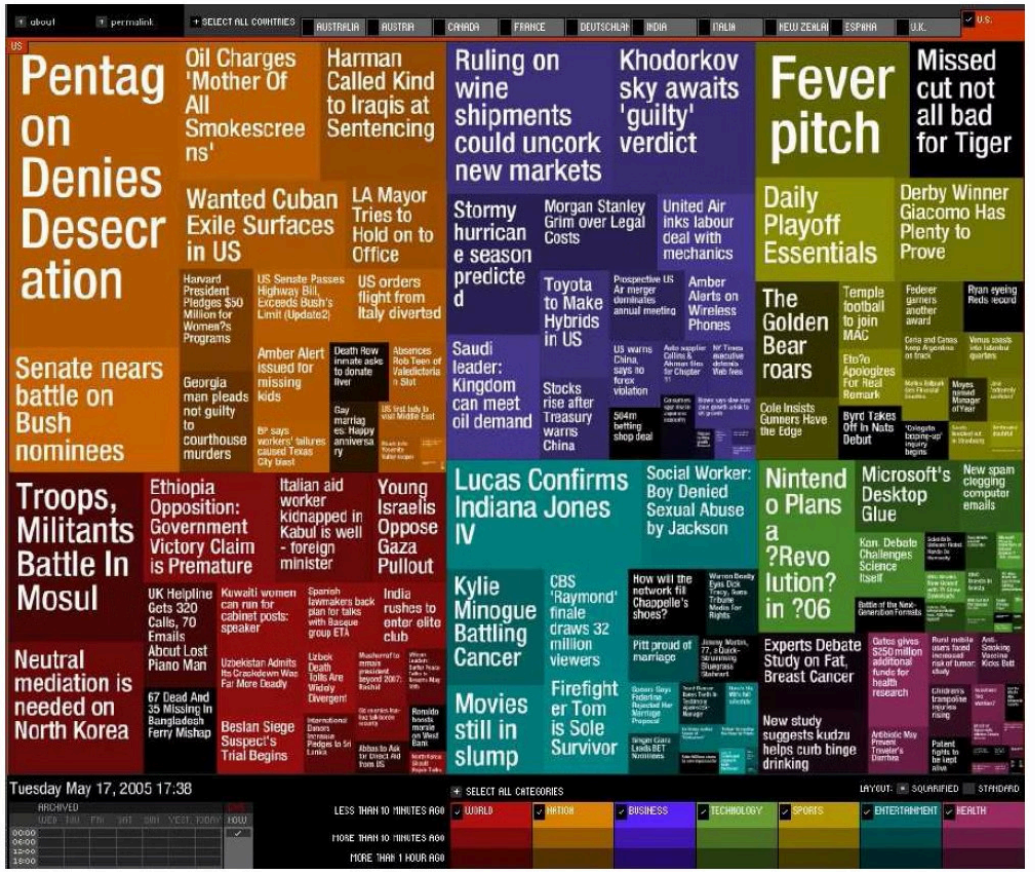




Ack.: http://nadeausoftware.com/articles/visualization

Visualizing Complex Data and Relations

- Visualizing non-numerical data: text and social networks
- Tag cloud: visualizing user-generated tags
  - The importance of tag is represented by font size/color
- Besides text data, there are also methods to visualize relationships, such as visualizing social networks



Newsmap: Google News Stories in 2005

Similarity and Dissimilarity

Similarity

- Numerical measure of how alike two data objects are
- Value is higher when objects are more alike
- Often falls in the range [0,1]

Dissimilarity (e.g., distance)

- Numerical measure of how different two data objects are
- Lower when objects are more alike
- Minimum dissimilarity is often 0
- Upper limit varies

Proximity

refers to a similarity or dissimilarity

Data Matrix and Dissimilarity Matrix

Data matrix

- n data points with p dimensions
- Two modes

Dissimilarity matrix

- n data points, but registers only the distance
- A triangular matrix
- Single mode

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Proximity Measure for Nominal Attributes

- Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i,j) = \frac{p-m}{p}$$

- Method 2: Use a large number of binary attributes
  - creating a new binary attribute for each of the *M* nominal states

Proximity Measure for Binary Attributes

Object *j*

- A contingency table for binary data
- Distance measure for symmetric binary variables:
- Distance measure for asymmetric binary variables:
- **Jaccard** coefficient (*similarity* measure for ***asymmetric*** binary variables):

	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

$$d(i,j) = \frac{r+s}{q+r+s+t} \qquad d(i,j) = \frac{r+s}{q+r+s}$$

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

- Note: Jaccard coefficient is the same as "coherence":

$$coherence(i,j) = \frac{sup(i,j)}{sup(i)+sup(j)-sup(i,j)} = \frac{q}{(q+r)+(q+s)-q}$$

Dissimilarity between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

  - Gender is a symmetric attribute
  - The remaining attributes are asymmetric binary
  - Let the values Y and P be 1, and the value N 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Standardizing Numeric Data

- Z-score:

$$z = \frac{x - \mu}{\sigma}$$

  - X: raw score to be standardized, μ: mean of the population, σ: standard deviation
  - the distance between the raw score and the population mean in units of the standard deviation
  - negative when the raw score is below the mean, "+" when above
- An alternative way: Calculate the mean absolute deviation
  - standardized measure (*z-score*)

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf})$$

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation

Distance on Numeric Data: Minkowski Distance

- *Minkowski distance*: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

- Properties
  - d(i, j) > 0 if i ≠ j, and d(i, i) = 0 (Positive definiteness)
  - d(i, j) = d(j, i)(Symmetry)
  - d(i, j) ≤ d(i, k) + d(k, j)(Triangle Inequality)
- A distance that satisfies these properties is a **metric**

Special Cases of Minkowski Distance

- *h* = 1:Manhattan (city block, L1 norm) distance
  - E.g., the Hamming distance: the number of bits that are different between two binary vectors
- *h* = 2: (L2 norm) Euclidean distance
- *h* → infinite."supremum" (Lmax norm, L_inf norm) distance.
  - This is the maximum difference between any component (attribute) of the vectors

$$d(i, j) = \lim_{h \to \infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_{f}^{p} |x_{if} - x_{jf}|$$

Cosine Similarity

- A **document** can be represented by thousands of attributes, each recording the *frequency* of a particular word (such as keywords) or phrase in the document.

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- Other vector objects: gene features in micro-arrays, …
- Applications: information retrieval, biologic taxonomy, gene feature mapping, ...
- Cosine measure: If *d1* and *d2* are two vectors (e.g., term-frequency vectors), then

$$\cos(d1, d2) = (d1 \times d2) / ||d1|| \, ||d2||$$

(x: indicates vector dot product, $||d||$: the length of vector d)

Example: Cosine Similarity

- Ex: Find the **similarity** between documents 1 and 2.

d1 **=**(5, 0, 3, 0, 2, 0, 0, 2, 0, 0)

d2 **=**(3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

*d1*x*d2* = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25

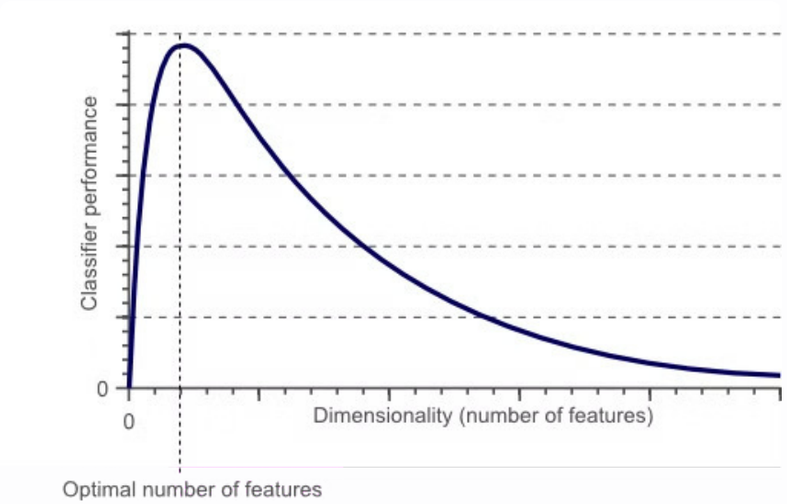
||*d1*||= (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)**0.5**=(42)**0.5**= 6.481

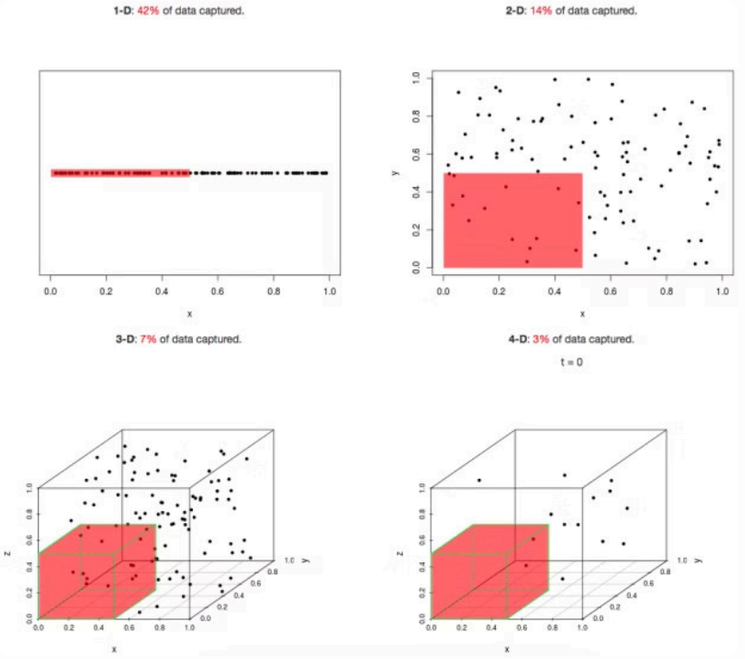
||*d2*||= (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)**0.5**=(17)**0.5**= 4.12

cos(*d1, d2*) = 0.94

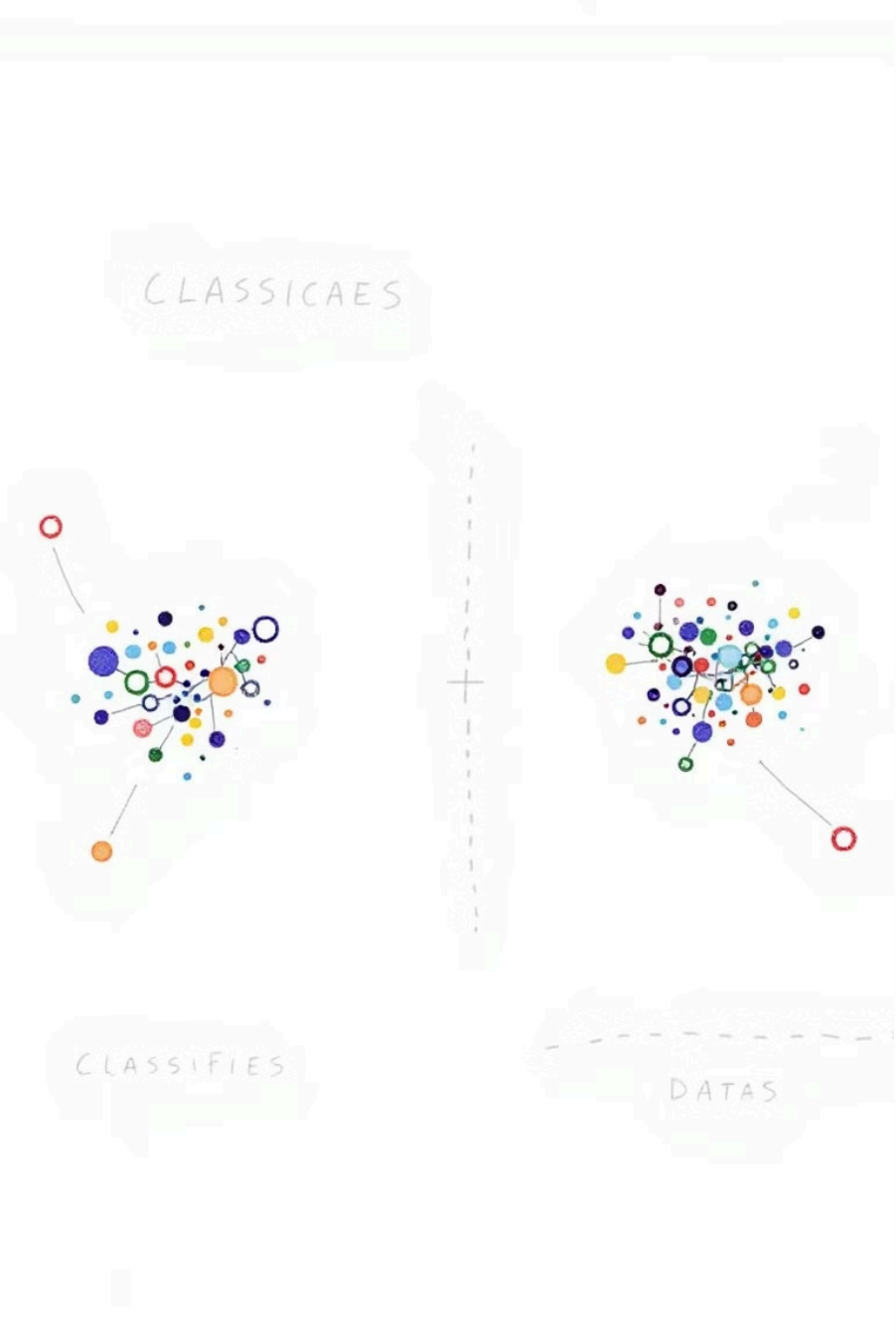
Curse of Dimensionality

- As the number of dimensions increases, data points become sparse and distant in high-dimensional space.
- This leads to challenges in:
  - **Machine Learning**: Harder to find meaningful patterns due to sparsity.
  - **Optimization**: Increased computational complexity.
  - **Distance Metrics**: Euclidean distance loses effectiveness.
- Solutions: **Dimensionality reduction (PCA, t-SNE), feature selection, domain knowledge**.





ref: **https://en.wikipedia.org/wiki/Curse_of_dimensionality**



Mining Tasks and Evaluation

Supervised vs. Unsupervised Learning

Supervised Learning

Learning with **labels + data**.

Goal: predict values

Examples: classification, regression, ranking.

Unsupervised Learning

Learning with **data only**.

Goal: find patterns/groupings.

Examples: clustering, frequent patterns, dimension reduction.

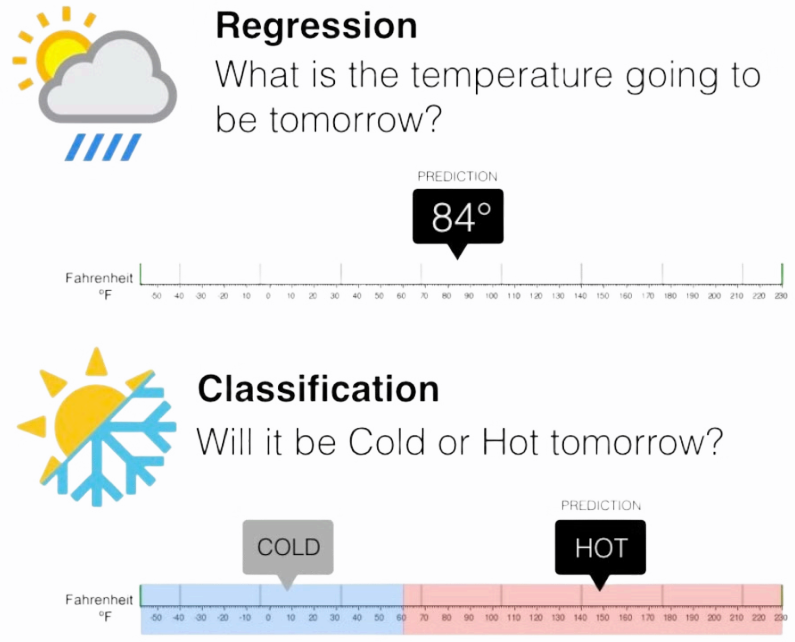
Classification vs. Regression

Classification

Supervised learning with **discrete** target variables. Predicts categories or classes.

Regression

Supervised learning with **continuous** target variables. Predicts numerical values.



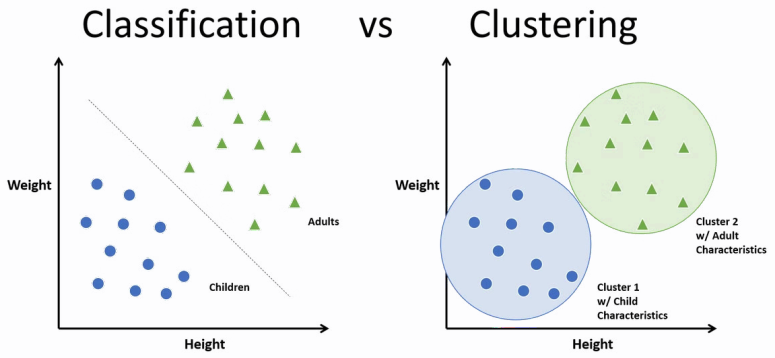
Classification vs. Clustering

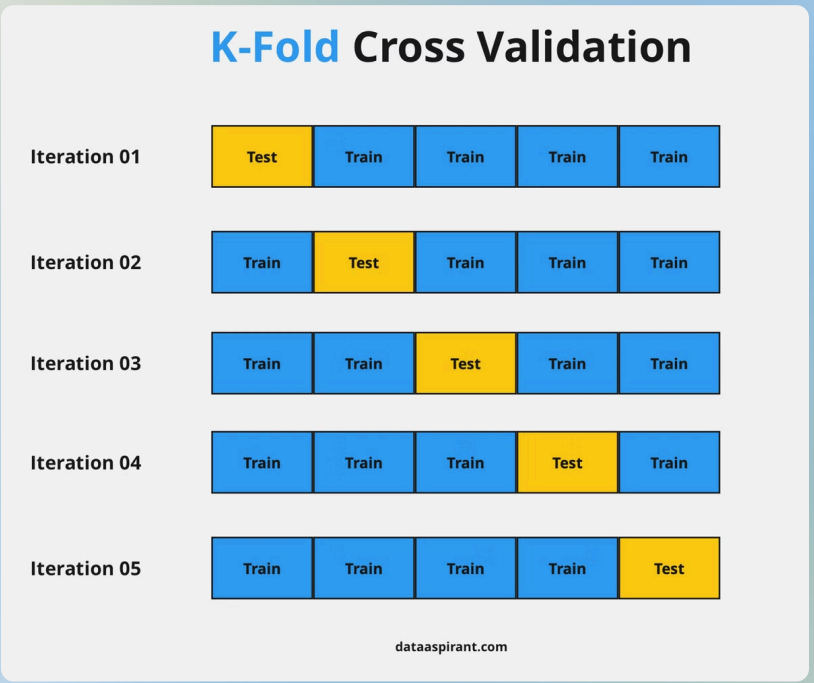
Classification Order

Classes have specific meaning.
Class A is different from Class B.

Clustering Exchangeability

Clusters are exchangeable.
Swapping Cluster A and B doesn't change meaning.





Cross Validation - k folds

Split Data

Divide dataset into k equal parts or folds.

Train and Test

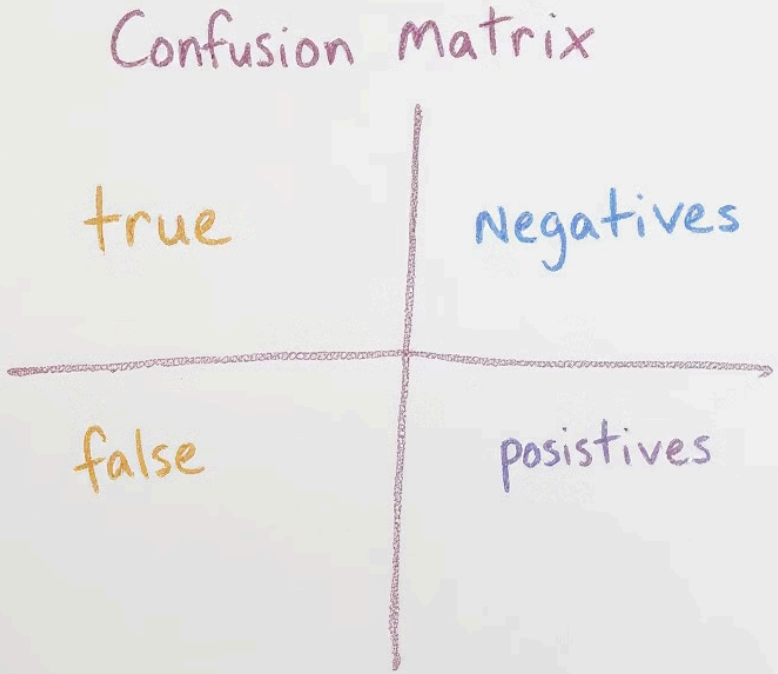
Use k-1 folds for training, 1 fold for testing.

Rotate and Repeat

Repeat process k times, using each fold as test set once.

Average Results

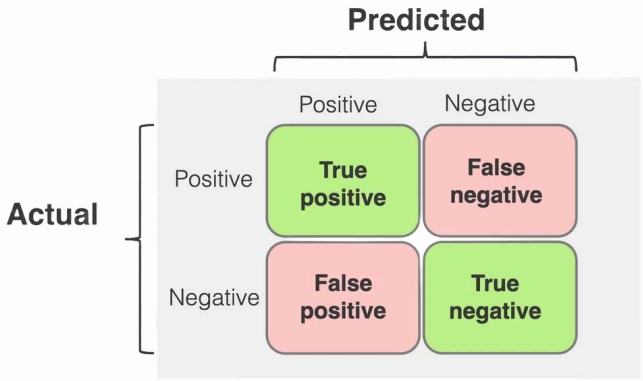
Calculate average performance across all k iterations.



Confusion Matrix: TP, TN, FP, FN

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Classification Metrics



1. Accuracy

   (TP + TN) / (TP + FP + FN + TN)

2. Precision

   TP / (TP + FP)

3. Recall

   TP / (TP + FN)

4. F1 Score

   Harmonic Mean between Precision and Recall