



Data Mining – Chapter 2

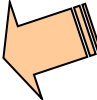
Kyuseok Shim

Seoul National University

<http://kdd.snu.ac.kr/~shim>

Extended from the slides of the book "Data Mining:
Concepts and Techniques (3rd ed.)" provided by Jiawei
Han, Micheline Kamber, and Jian Pei

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

Types of Data Sets

- Record

- Relational records
- Data matrix, e.g., numerical matrix, crosstabs
- Document data: text documents: term-frequency vector
- Transaction 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

- Graph and network

- World Wide Web
- Social or information networks
- Molecular Structures

- Ordered

- Video data: sequence of images
- Temporal data: time-series
- Sequential Data: transaction sequences
- Genetic sequence data

- Spatial, image and multimedia:

- Spatial data: maps
- Image data:
- Video data:

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Important Characteristics of Structured Data

- 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
 - Ordinal
 - Numeric: quantitative
 - Interval-scaled
 - Ratio-scaled

Attributes

- The term **dimension** is used in data warehousing.
- Machine learning literature tends to use the term **feature**, while statisticians prefer the term **variable**.
- Data mining and database professionals use the term **attribute**.
- Observed values for a given attribute are known as **observations**.
- A set of attributes used to describe a given object is called an **attribute vector** (or **feature vector**).
- The distribution of data involving one attribute (or variable) is called **univariate**.
- A **bivariate** distribution involves **two attributes**.
- The type of an attribute is determined by the set of possible values—**nominal**, **binary**, **ordinal**, or **numeric**—the attribute can have.

Nominal Attributes

- Nominal means “relating to names.”
- The values are symbols or names of things.
- Each value represents some kind of category, code, or state, and so nominal attributes are also referred to as **categorical**.
- Values do not have any meaningful order.
- It is possible to represent such symbols or “names” with numbers.
 - With hair color, for instance, we can assign a code of 0 for black, 1 for brown, and so on.
- However, in such cases, the numbers are not intended to be used quantitatively.
 - It makes no sense to find the mean (average) value or median (middle) value for such an attribute, given a set of objects.
- **Mode** is the attribute’s most commonly occurring value.

Binary Attributes

- A binary attribute is a **nominal** attribute with only two categories
 - 0 means that the attribute is absent, and 1 means that it is present.
- Given the attribute smoker describing a patient object, 1 indicates that the patient smokes, while 0 indicates that the patient does not.
- A binary attribute is **symmetric** if both of its states are equally valuable and carry the same weight.
 - e.g., the gender attribute – male and female.
- A binary attribute is **asymmetric** if the outcomes of the states are not equally important
 - e.g., the positive and negative outcomes of a medical test for HIV.

Ordinal Attributes

- An ordinal attribute is an attribute with possible values that have a meaningful **order** or **ranking** among them, but the magnitude between successive values is not known.
- Ordinal attributes may also be obtained from the discretization of numeric quantities by splitting the value range into a finite number of ordered categories.
- The central tendency of an ordinal attribute can be represented by its **mode** and its **median** (the middle value in an ordered sequence), but the mean cannot be defined.
- Note that nominal, binary, and ordinal attributes are qualitative.
 - They describe a feature of an object without giving an actual size or quantity.
 - The values of such qualitative attributes are typically words representing categories.

Ordinal Attributes

- e.g., size of drinks available at a fast-food restaurant.
 - This nominal attribute has three possible values: small, medium, and large.
 - Values have a meaningful sequence (which corresponds to increasing drink size)
 - However, we cannot tell from the values how much bigger, say, a medium is than a large.
- Useful for registering subjective assessments of qualities that cannot be measured objectively – thus used in surveys for ratings.
 - 0: very dissatisfied
 - 1: somewhat dissatisfied
 - 2: neutral
 - 3: satisfied
 - 4: very satisfied

Attribute Types (Recap.)

- **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 Attributes

- A numeric attribute is quantitative
 - A measurable quantity, represented in integer or real values.
- Numeric attributes can be interval-scaled or ratio-scaled.

Numeric Attribute Types

■ Interval-Scaled Attributes

- Measured on a scale of equal-size units.
- Values have order and can be positive, 0, or negative.
- In addition to providing a ranking of values, such attributes allow us to compare and quantify the difference between values.
- Values have order
 - e.g., temperature in C° or F°, calendar dates
- No true zero-point
- We can compute their mean value, in addition to the median and mode measures of central tendency.

Numeric Attribute Types

■ Ratio-Scaled Attributes

- A numeric attribute with an inherent zero-point.
- If a measurement is ratio-scaled, we can represent a value as being a multiple (or ratio) of another value.
- In addition, the values are ordered, and we can also compute the difference between values, as well as the mean, median, and mode.
- Inherent zero-point
- Examples - Temperature in Kelvin, length, counts, monetary quantities.
 - 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°).
 - Count attributes such as years of experience (e.g., the objects are employees) and number of words (e.g., the objects are documents).
 - Weight, height, latitude and longitude coordinates (e.g., when clustering houses), and monetary quantities (e.g., you are 100 times richer with \$100 than with \$1).

Discrete vs. Continuous Attributes

■ Discrete Attribute

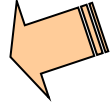
- Has only a finite or countably infinite set of values
 - e.g., zip codes, profession, or the set of words in a collection of documents
- An attribute is countably infinite if the set of possible values is infinite but the values can be put in a one-to-one correspondence with natural numbers.
- The attribute customer ID is countably infinite.
 - Number of customers can grow to infinity, but in reality, the actual set of values is countable (where the values can be put in one-to-one correspondence with the set of integers).
- Note: Binary attributes are a special case of discrete attributes

Discrete vs. Continuous 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
- The terms **numeric** attribute and **continuous** attribute are often used interchangeably in the literature.

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

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.

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

- Weighted arithmetic mean:
- Trimmed mean: chopping extreme values

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

- Median:

- Middle value if odd number of values, or average of the middle two values otherwise

- Estimated by interpolation (for *grouped data*):

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

- Mode

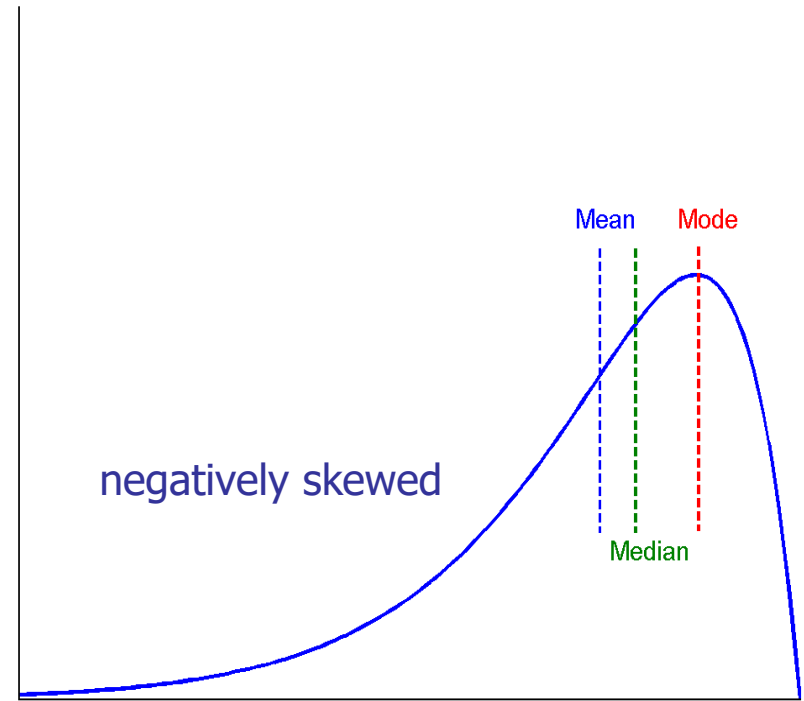
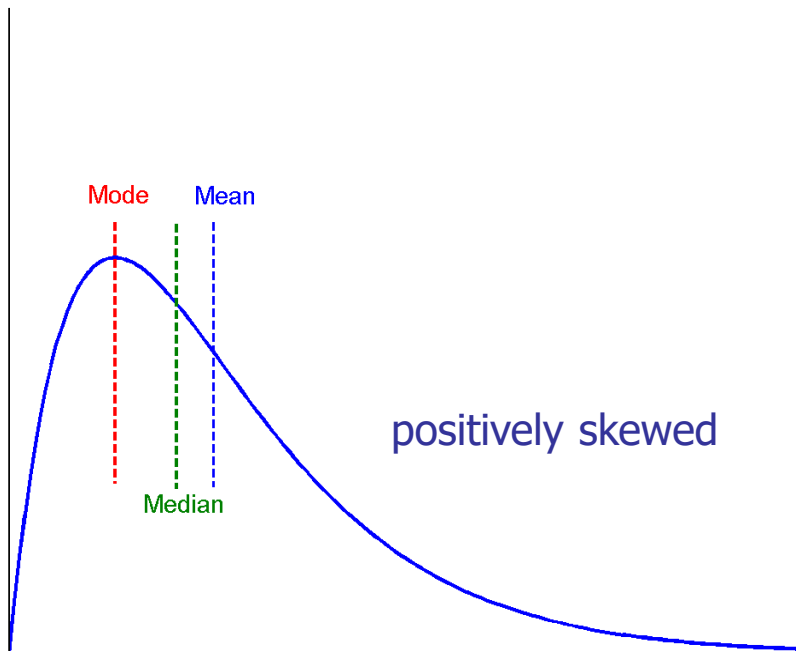
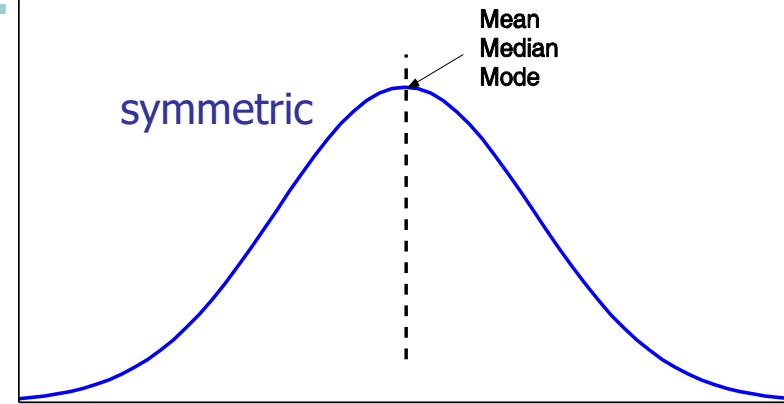
- Value that occurs most frequently in the data
- Unimodal, bimodal, trimodal

- Empirical formula: $mean - mode = 3 \times (mean - median)$

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

Symmetric vs. Skewed Data

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



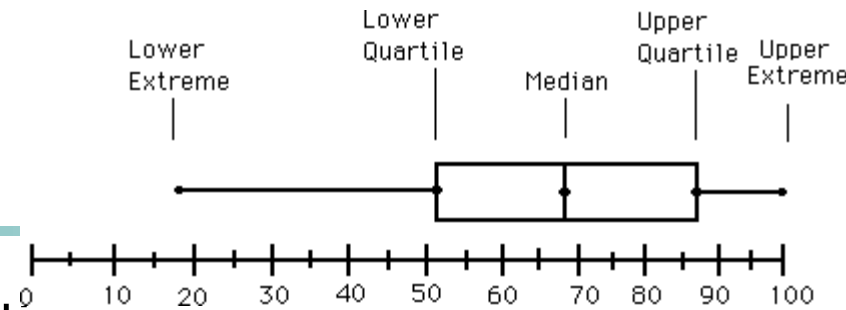
Quantiles

- Points taken at regular intervals of a data distribution, dividing it into equal-size consecutive sets.
- The k -th q -quantile, where k is an integer such that $0 < k < q$, is the value x such that
 - at most k/q of the data values are less than x
 - at most $(q-k)/q$ of the data values are more than x
- The median, quartiles, and percentiles are widely used quantiles.
 - The 2-quantile (i.e., the median) – data point dividing the lower and upper halves of the data distribution.
 - The 4-quantiles (referred to as quartiles) – three data points that split the data distribution into four equal parts
 - The 100-quantiles (referred to as percentiles) - 99 data points that split the data distribution into 100 equal-sized parts.

Interquartile Range (IQR)

- The distance between the first and third quartiles
 - i.e., $IQR = Q_3 - Q_1$
- A simple measure of spread that gives the range covered by the middle half of the data.
- Consider data with 12 observations, already sorted in increasing order.
- The quartiles for this data are the third, sixth, and ninth values, respectively, in the sorted list.

Boxplot Analysis

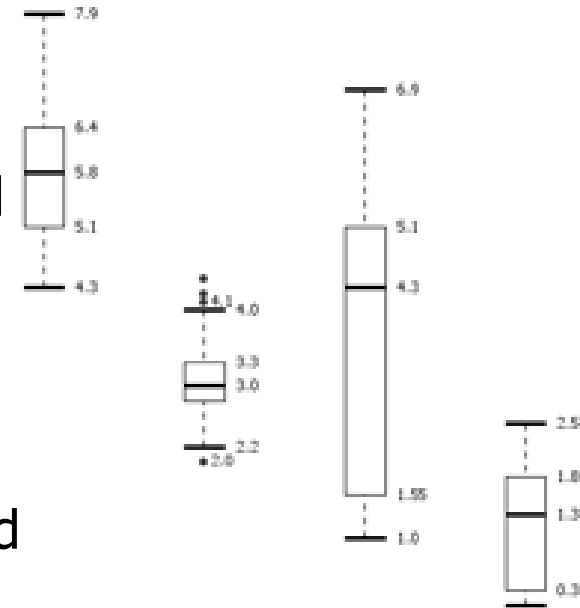


- **Five-number summary** of a distribution

- 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



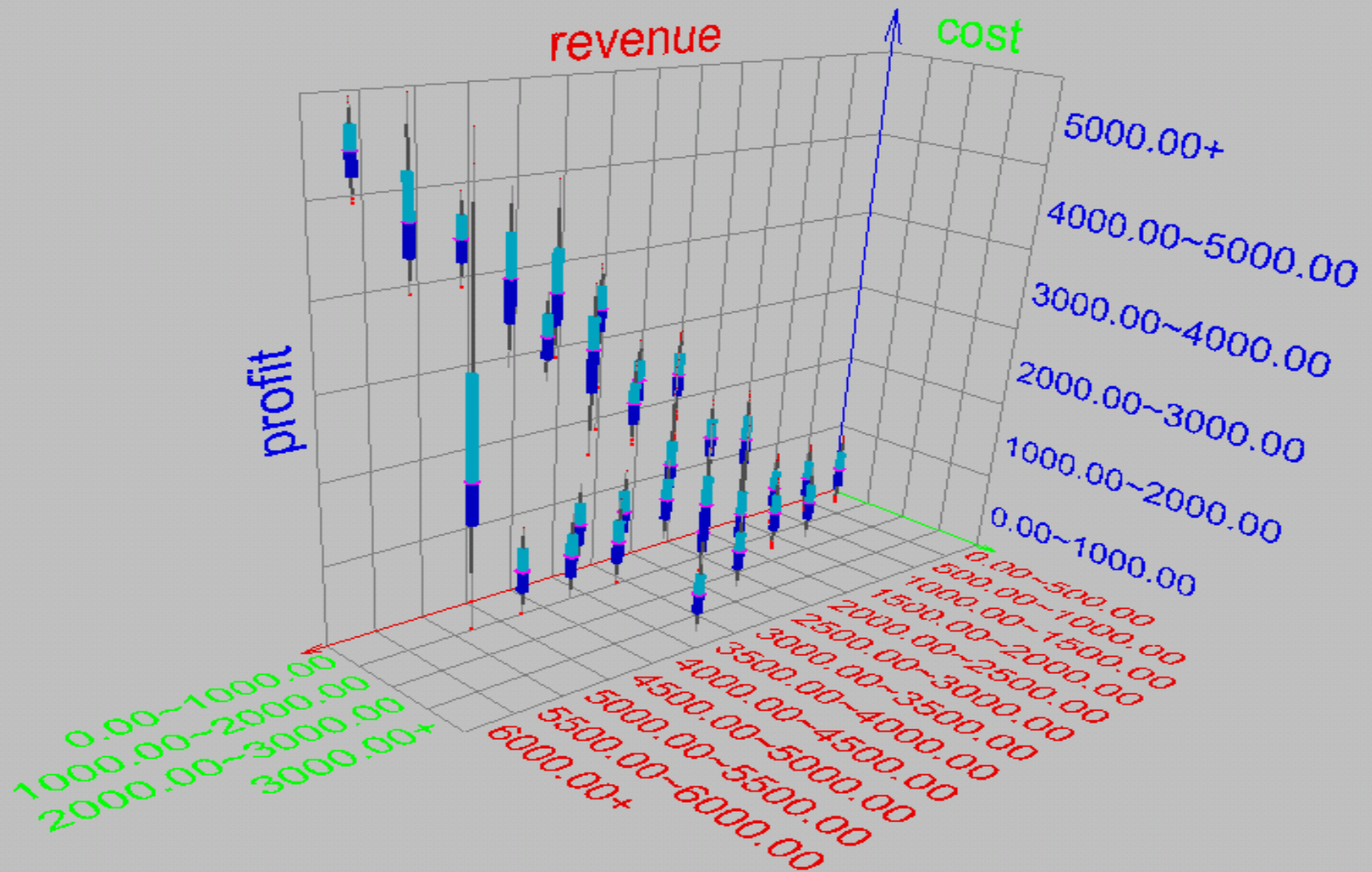
Measuring the Dispersion of Data

- 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)

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^n (x_i - \mu)^2 = \frac{1}{N} \sum_{i=1}^n x_i^2 - \mu^2$$

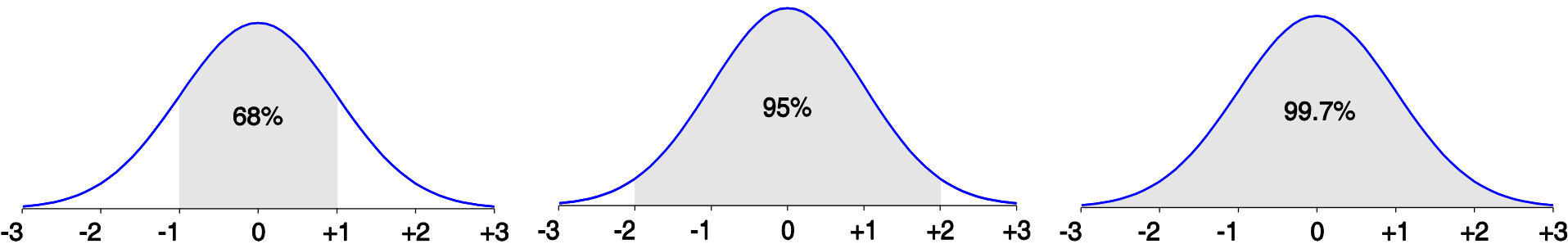
- **Standard deviation** σ is the square root of variance σ^2

Visualization of Data Dispersion: 3-D Boxplots



Properties of Normal Distribution Curve

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



Graphic Displays of Basic Statistical Descriptions

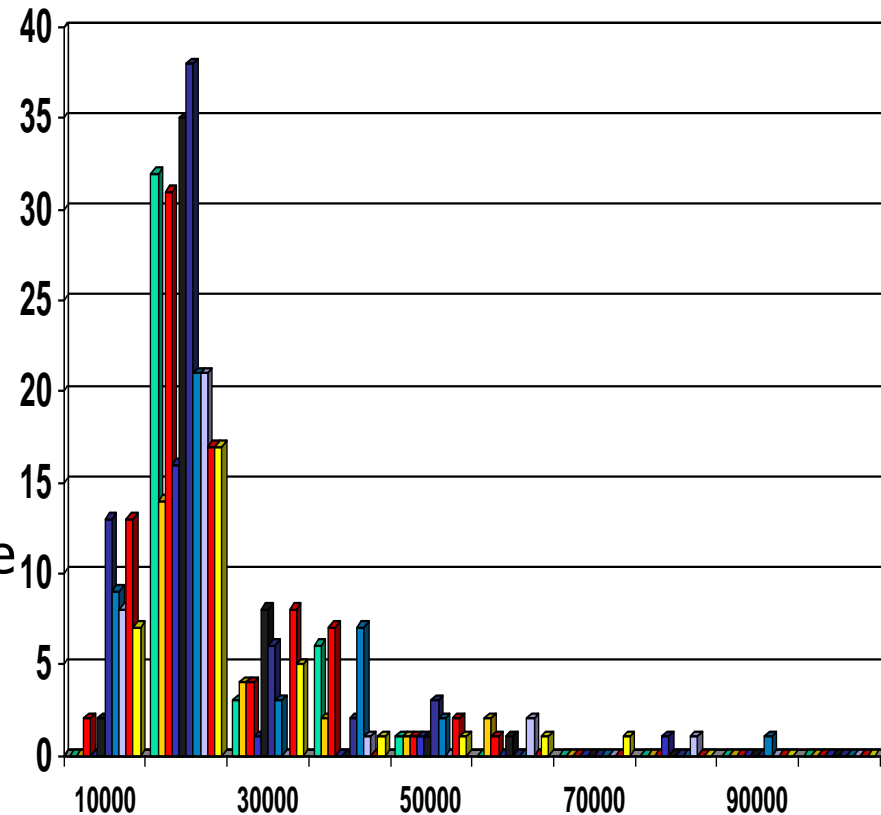
- **Boxplot:** graphic display of five-number summary
- **Histogram:** x-axis are values, y-axis repres. frequencies
- **Quantile plot:** each value x_i is paired with f_i indicating that approximately 100 f_i % of data are $\leq x_i$
- **Quantile-quantile (q-q) plot:** graphs the quantiles of one univariant 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

Histograms

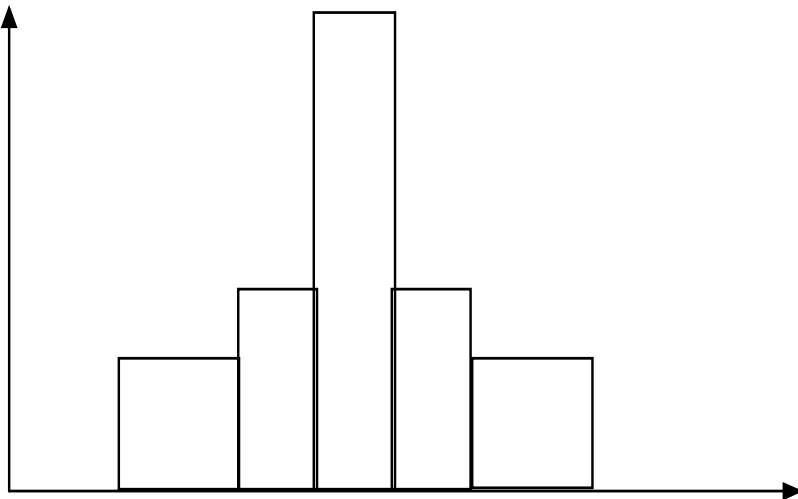
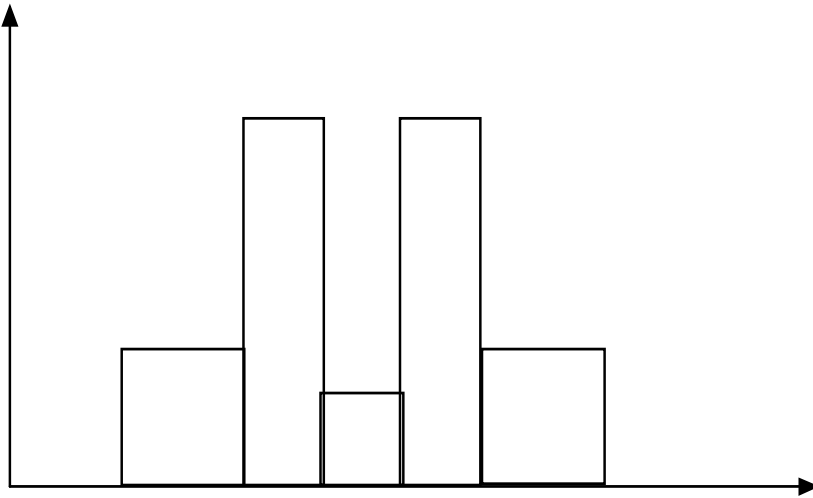
- Graph display of tabulated frequencies for a given attribute, X .
- If X is nominal, such as automobile model or item type, then a pole or vertical bar is drawn for each known value of X .
- The height of the bar indicates the frequency (i.e., count) of that X value.
- The range of values for X is partitioned into disjoint consecutive subranges (referred to as **buckets** or **bins**).
- The range of a bucket is known as the **width**.
- Example
 - A price attribute with a value range of \$1 to \$200 (rounded up to the nearest dollar) can be partitioned into subranges 1 to 20, 21 to 40, 41 to 60, and so on.
 - For each subrange, a bar is drawn with a height that represents the total count of items observed within the subrange.

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



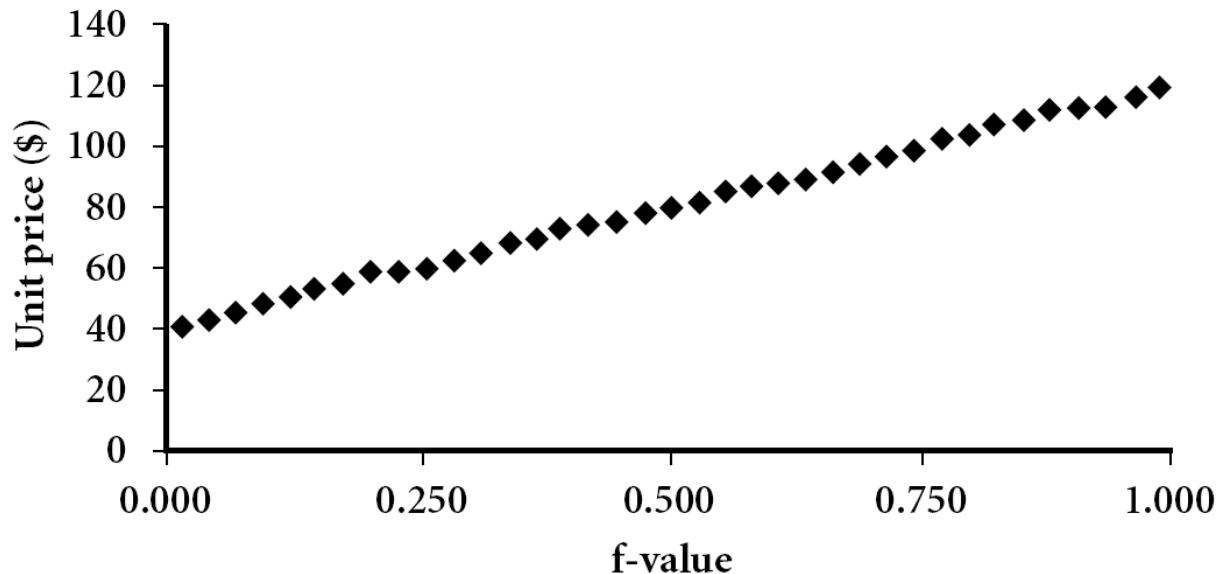
Histograms Often Tell More than Boxplots



- 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

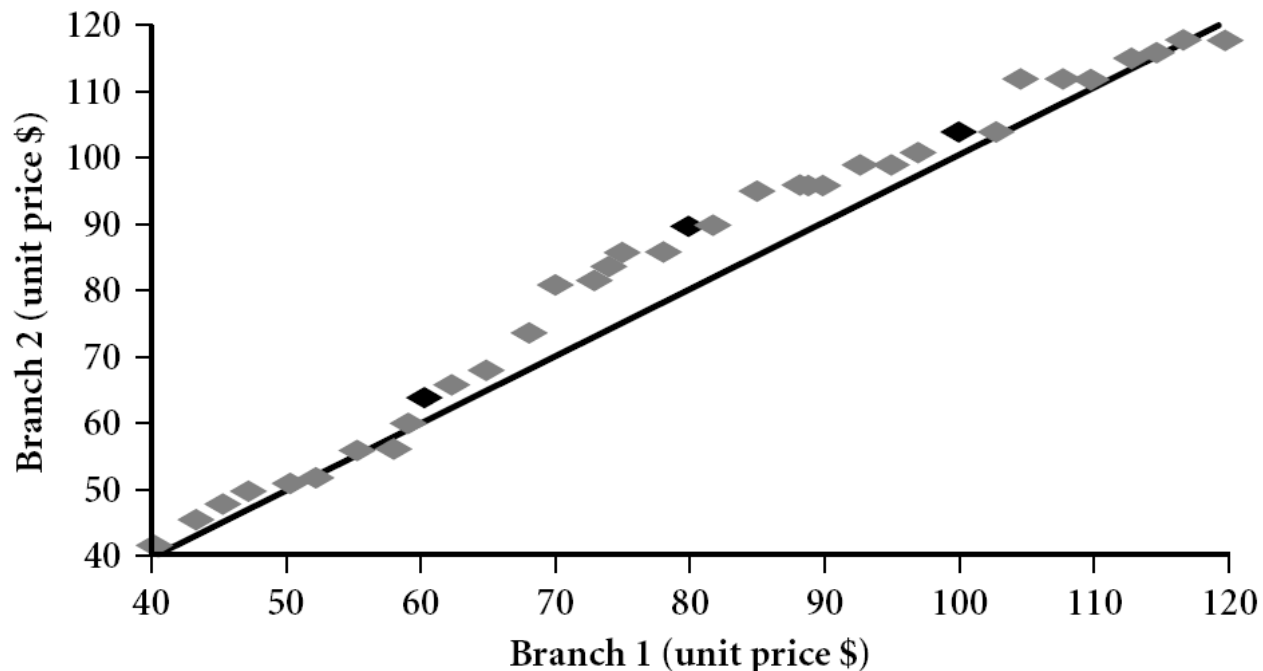
Quantile Plot

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



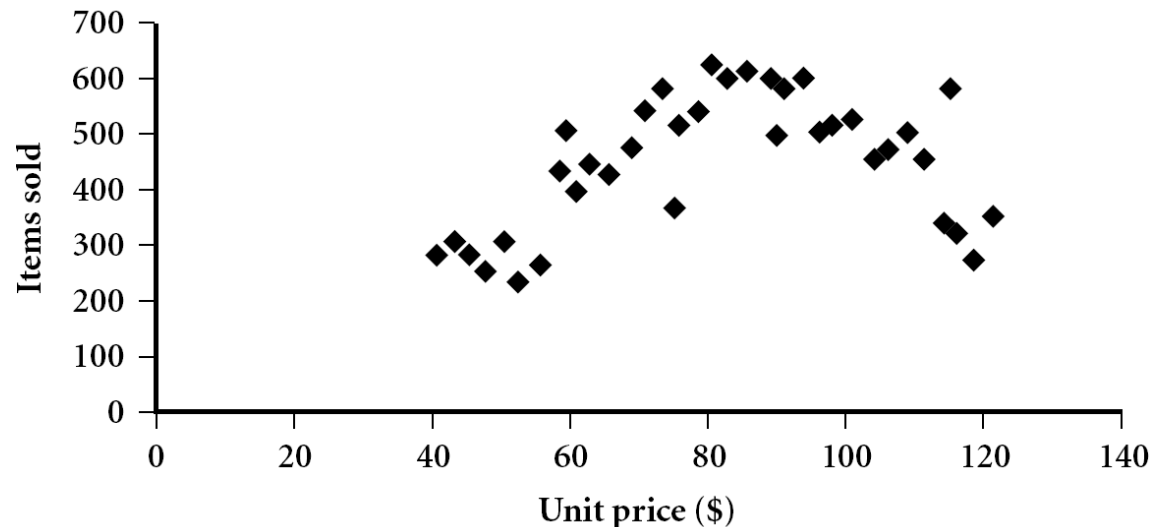
Quantile-Quantile (Q-Q) Plot

- 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 B



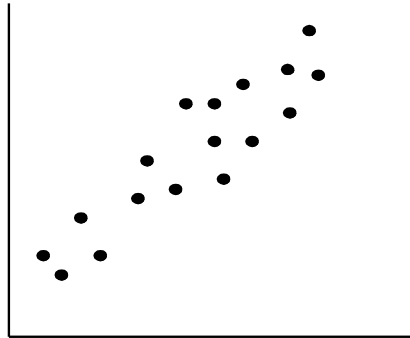
Scatter plot

- One of the most effective graphical methods for determining if there appears to be a relationship, pattern, or trend between **two numeric attributes**.
- Provides a first look at bivariate data to see clusters of points, outliers, etc. or to explore the possibility of correlation relationships.
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane.



Positively and Negatively Correlated Data

- Two attributes, X , and Y , are correlated if one attribute implies the other.
- Correlations can be positive, negative, or null (uncorrelated).
- If the plotted points pattern slopes from lower left to upper right, this means that the values of X increase as the values of Y increase, suggesting a **positive correlation**.

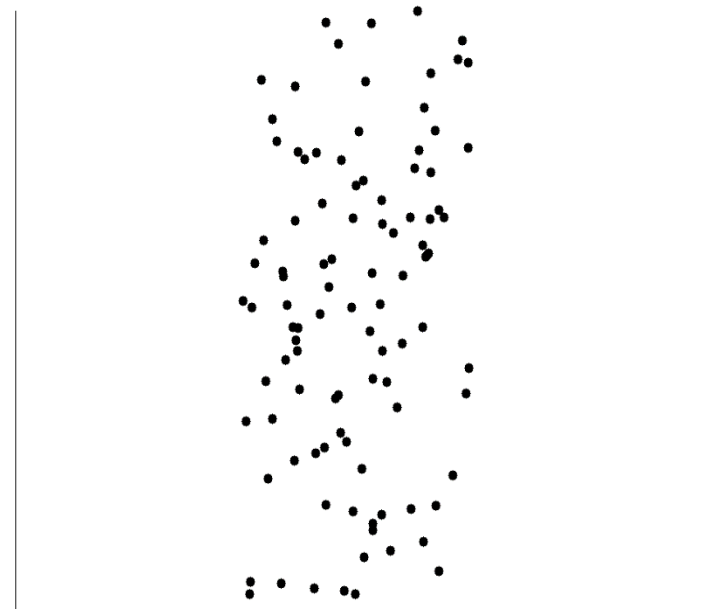
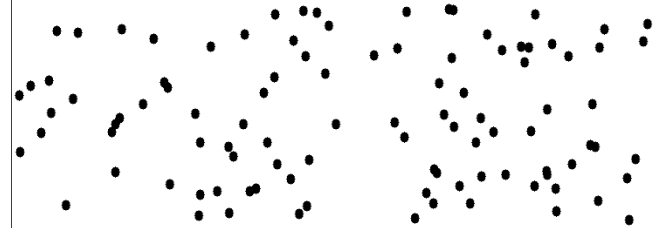
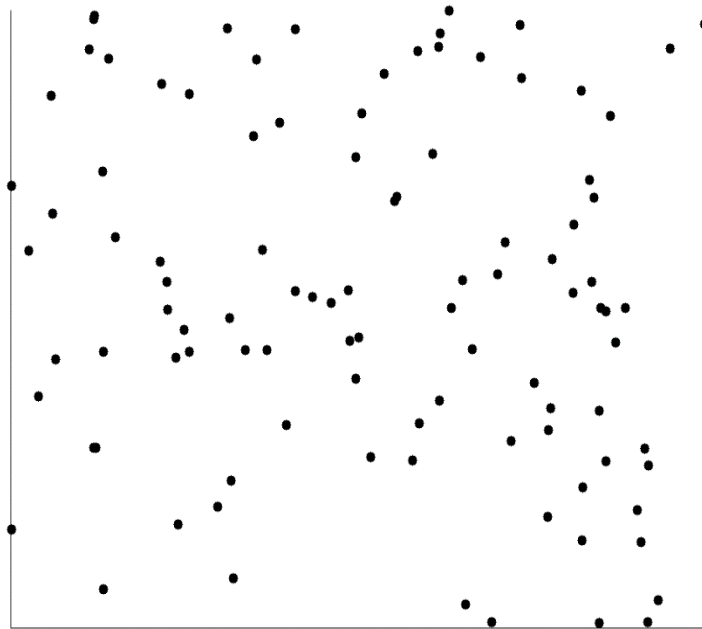


- If the pattern of plotted points slopes from upper left to lower right, the values of X increase as the values of Y decrease, suggesting a **negative correlation**.




- A line of best fit can be drawn to study the correlation between the variables.

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

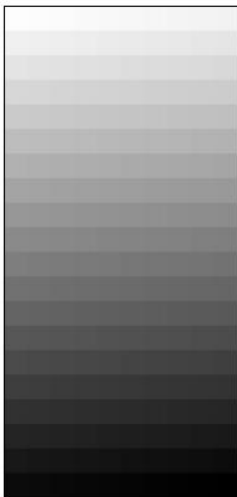
- Why data visualization?
 - Gain insight into an information space by mapping data onto graphical primitives
 - Provide qualitative overview of large data sets
 - Search for patterns, trends, structure, irregularities, relationships among data that are otherwise not easily observable by looking at the raw 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

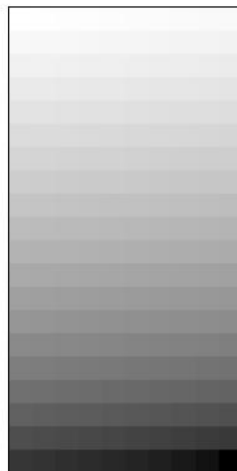
- A simple way to visualize the value of a dimension is to use a pixel where the color of the pixel reflects the dimension's value.
- For a data set of m dimensions, create m windows on the screen, one for each dimension.
- The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows.
- The colors of the pixels reflect the corresponding values.
- Inside a window, the data values are arranged in some global order shared by all windows.
- The global order may be obtained by sorting all data records in a way that's meaningful for the task at hand.

Pixel-Oriented Visualization

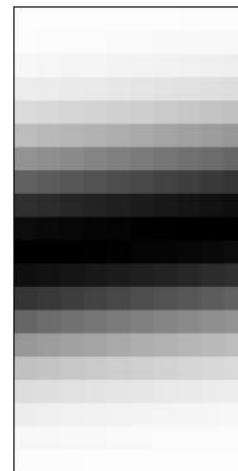
- A customer table consisting of four dimensions: income, credit limit, transaction volume, and age.
- We want to analyze the correlation between income and the other attributes by visualization.
- Sort all customers in income-ascending order, and use this order to lay out the customer data in the four visualization windows.
- The pixel colors are chosen so that the smaller the value, the lighter the shading.
- Using pixel-based visualization, we can easily observe the following.
 - Credit limit increases as income increases
 - Customers whose income is in the middle range are more likely to purchase more.
 - There is no clear correlation between income and age.



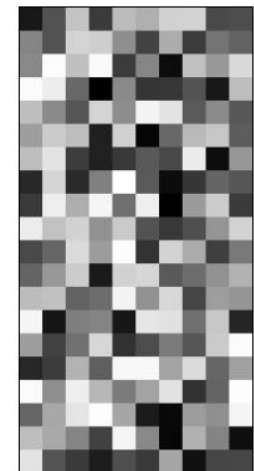
(a) Income



(b) Credit Limit



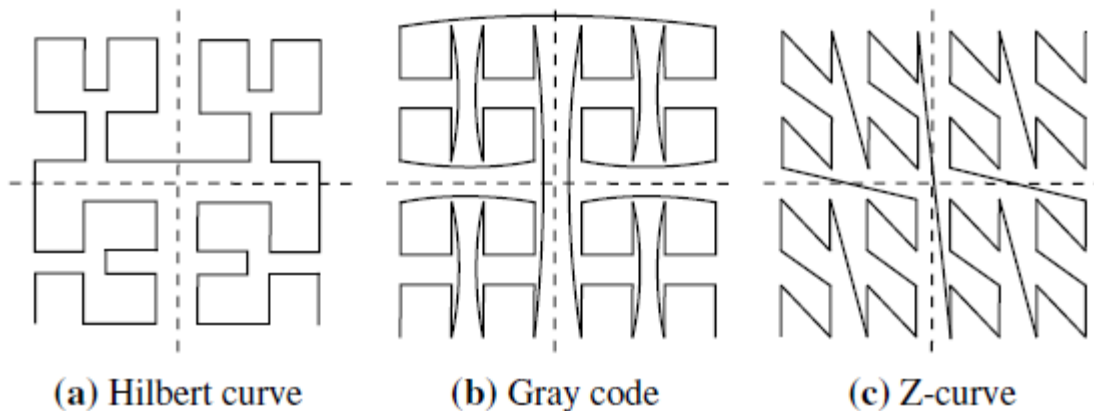
(c) transaction volume



(d) age

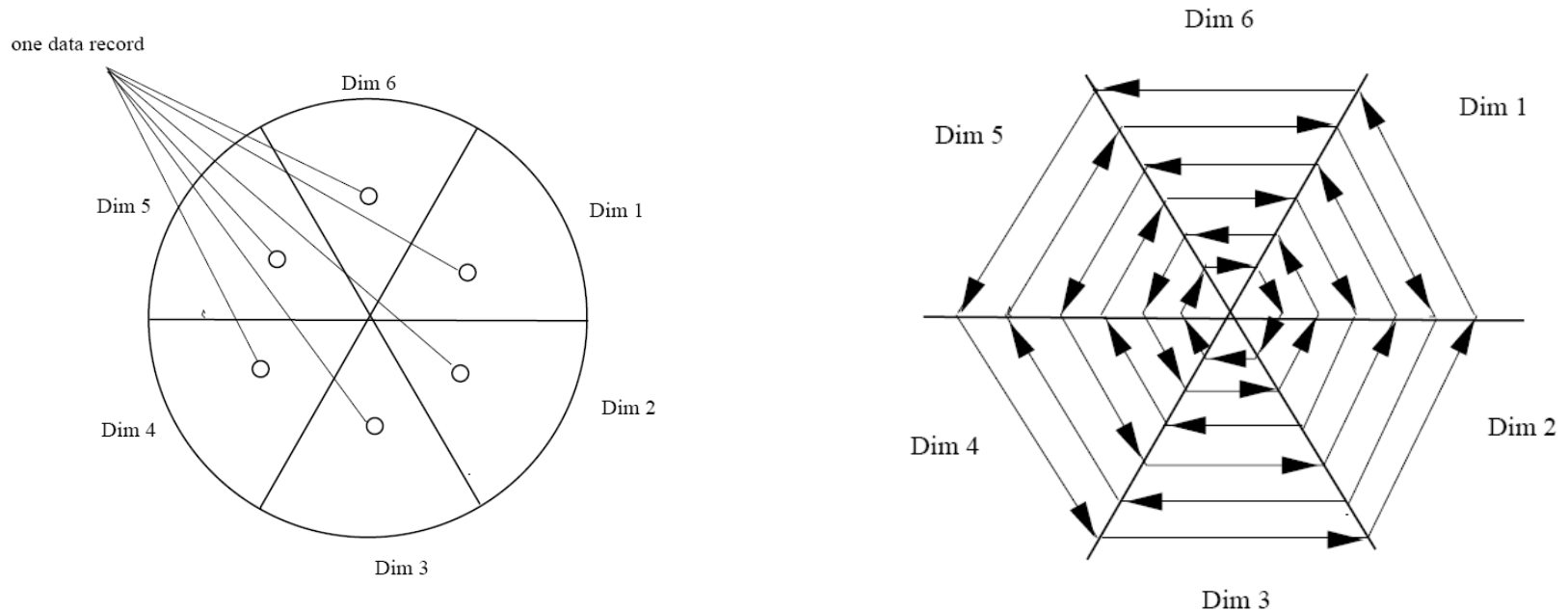
Pixel-Oriented Visualization

- Data records can also be ordered in a query-dependent way.
 - Given a point query, we can sort all records in descending order of similarity to the point query.
- Filling a window by laying out the data records in a linear way may not work well.
 - The first pixel in a row is far away from the last pixel in the previous row, though they are next to each other in the global order.
- To solve the problem, lay out the data records in a space-filling curve to fill the windows.
- A space-filling curve is a curve with a range that covers the entire n-dimensional unit hypercube.
- Since the visualization windows are 2-D, we can use any 2-D space-filling curve.



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
- This technique can ease the comparison of dimensions because the dimension windows are located side by side and form a circle.



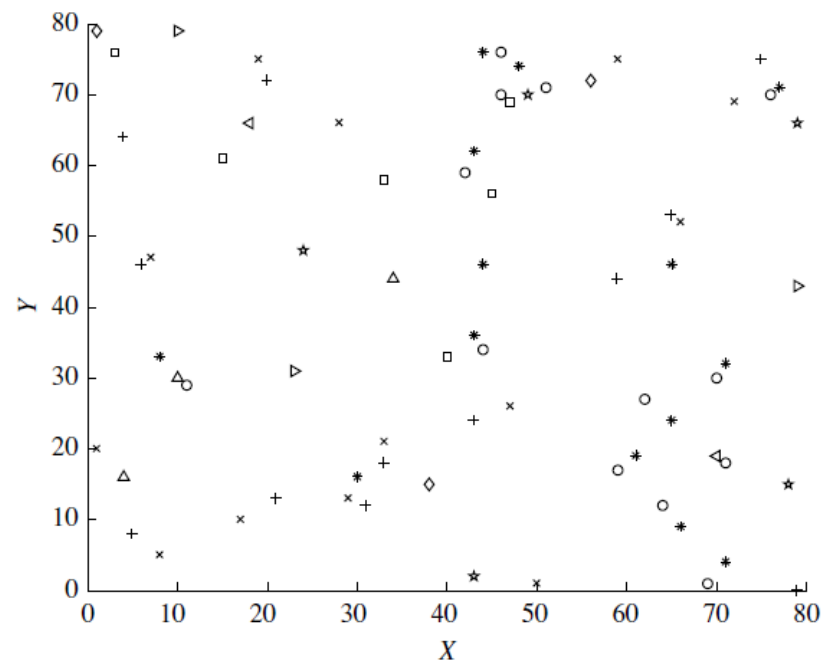
(a) Representing a data record in circle segment (b) Laying out pixels in circle segment

Geometric Projection Visualization Techniques

- A drawback of pixel-oriented visualization techniques
 - They cannot help us much in understanding the distribution of data in a multi-dimensional space.
- Geometric projection techniques help users find interesting projections of multidimensional data sets.
- Visualize a high-dimensional space on a 2-D display.
- Methods
 - Direct visualization
 - Scatterplot and scatterplot matrices
 - Landscapes
 - Projection pursuit technique: Help users find meaningful projections of multidimensional data
 - Projection views
 - Hyperslice
 - Parallel coordinates

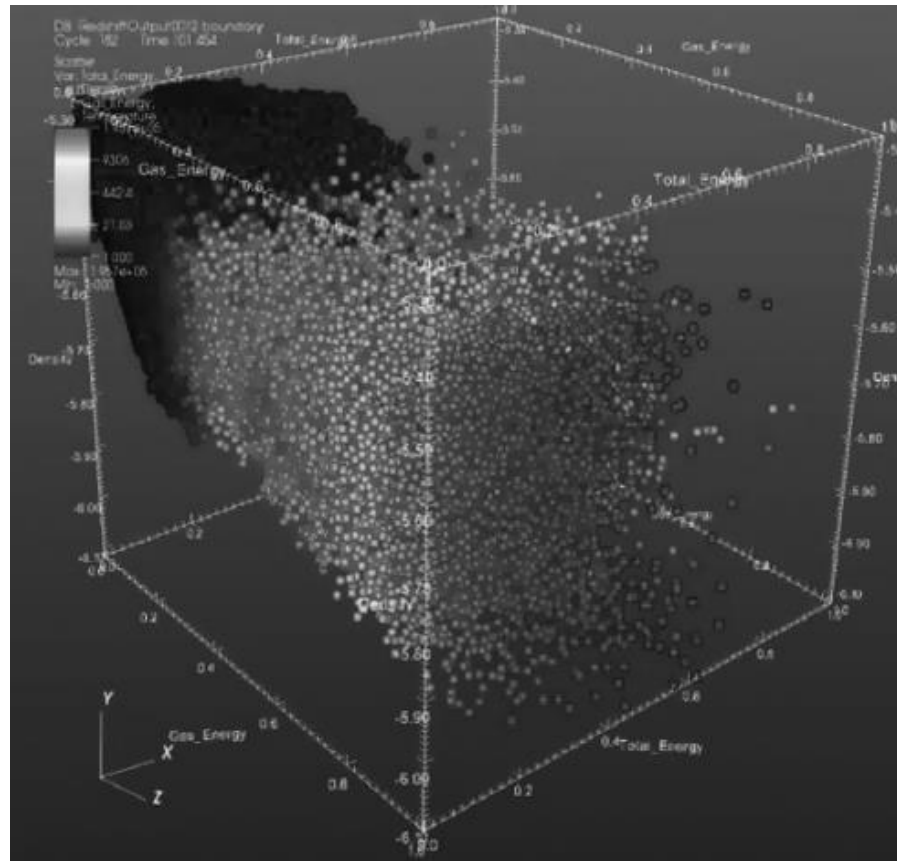
Visualization of a 2-D data set using a scatter plot

- A scatter plot displays 2-D data points using Cartesian coordinates.
- A third dimension can be added using different colors or shapes to represent different data points.
- Figure shows an example, where X and Y are two spatial attributes and the third dimension is represented by different shapes.
- Through this visualization, we can see that points of types “+” and “x” tend to be colocated.



Visualization of a 3-D data set using a scatter plot

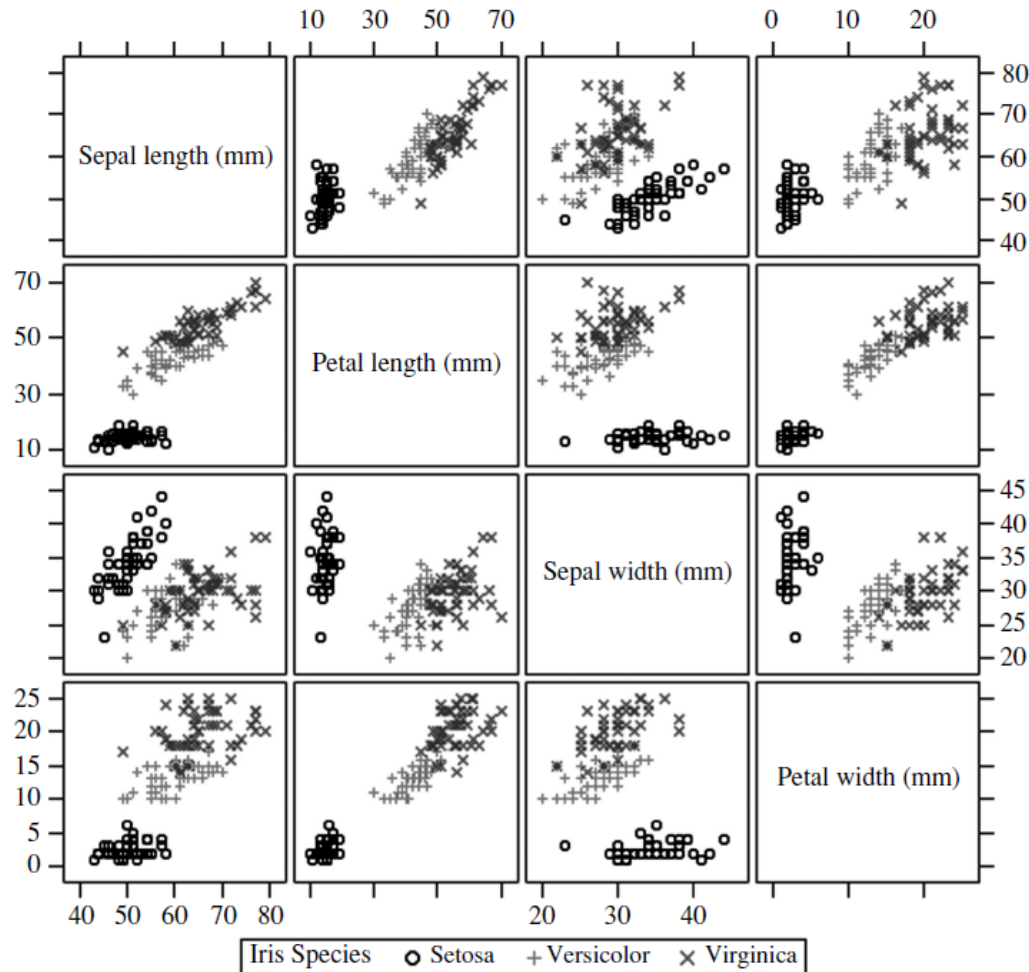
- Source: http://upload.wikimedia.org/wikipedia/commons/c/c4/Scatter_plot.jpg



Scatterplot Matrices

- For data sets with more than four dimensions, scatter plots are usually ineffective.
- The scatter-plot matrix technique is a useful extension to the scatter plot.
- For an n -dimensional data set, a scatter-plot matrix is an $n \times n$ grid of 2-D scatter plots that provides a visualization of each dimension with every other dimension.
- Figure shows an example, which visualizes the Iris data set.
 - The data set consists of 450 samples from each of three species of Iris flowers.
 - There are five dimensions in the data set: length and width of sepal and petal, and species.

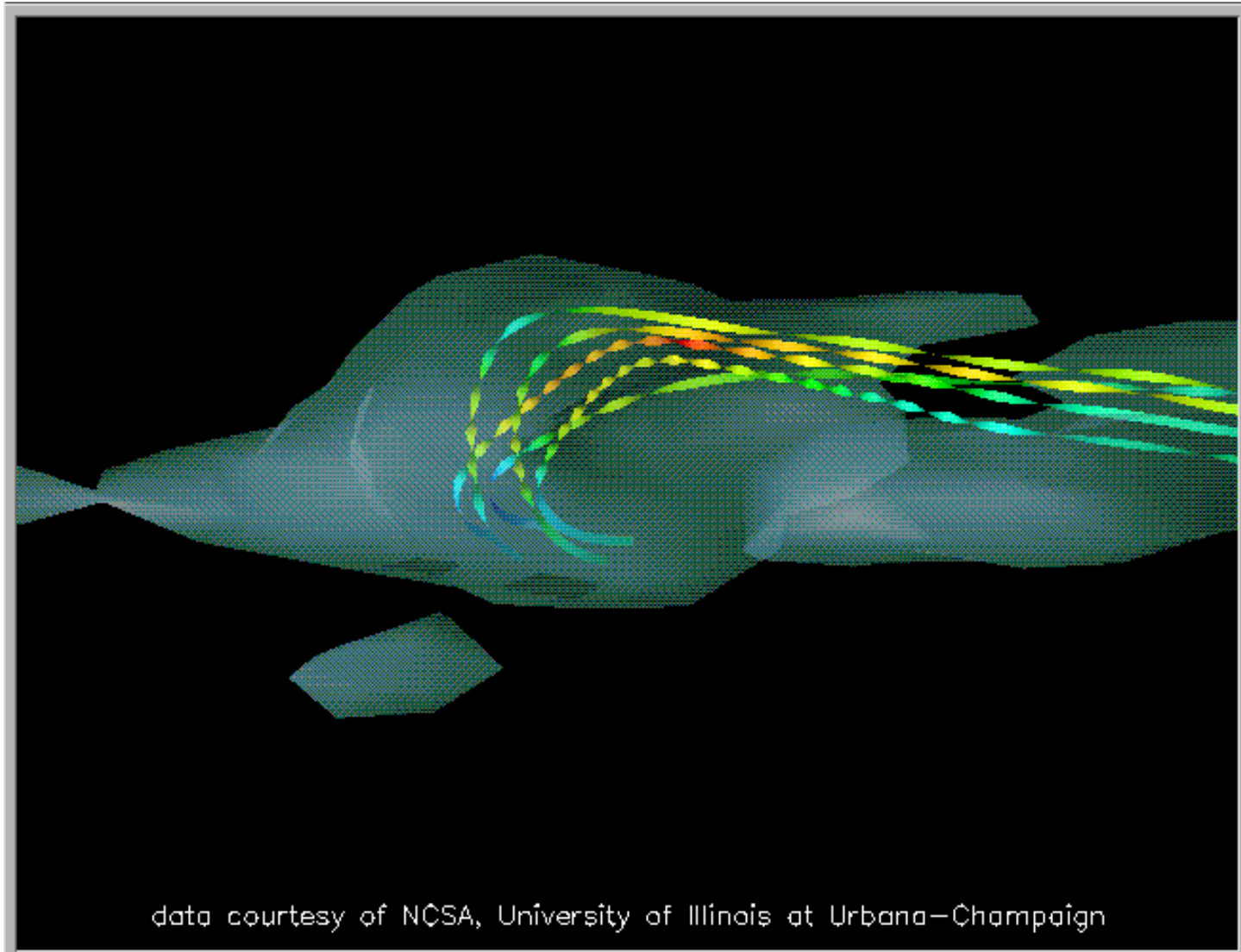
Visualization of the Iris data set using a scatter-plot matrix



<http://support.sas.com/documentation/cdl/en/grstatproc/61948/HTML/default/images/gsgscmat.gif>

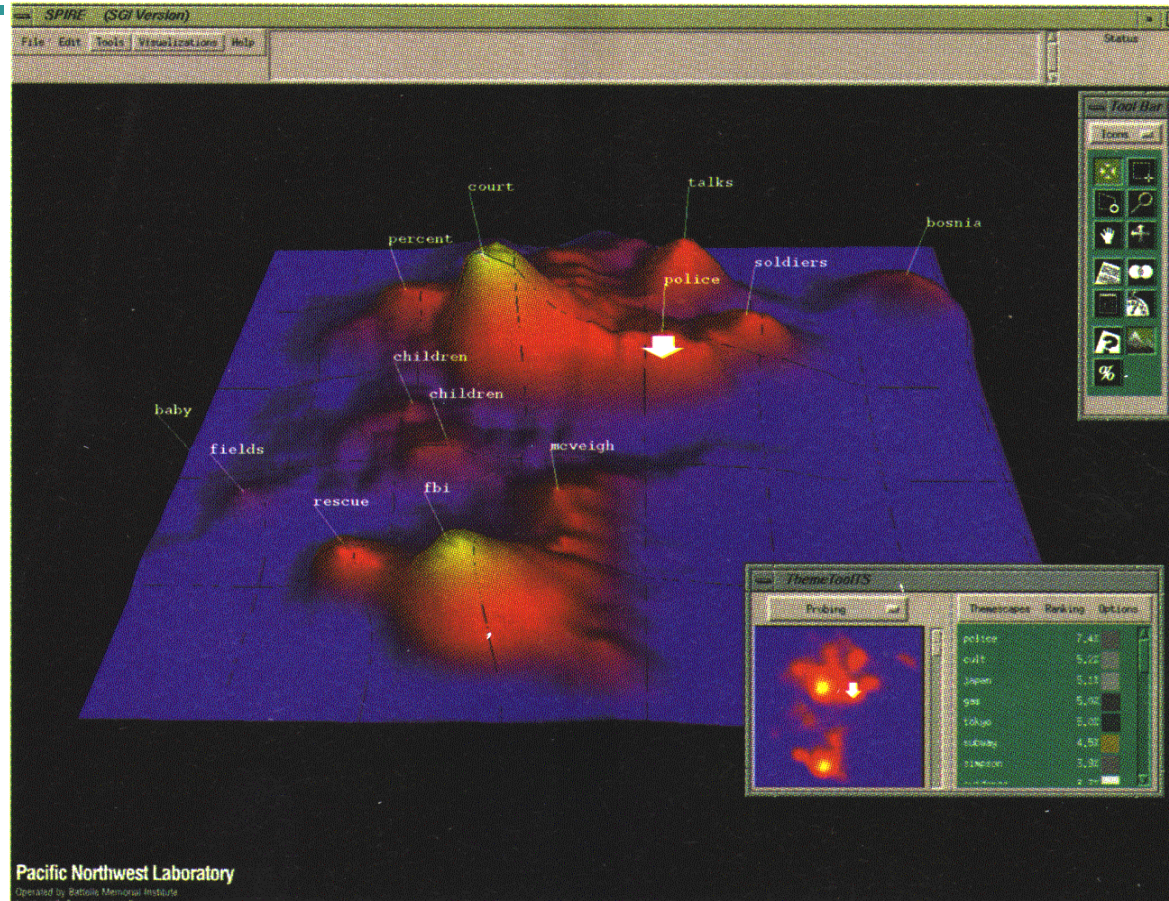
Direct Data Visualization

Ribbons with Twists Based on Vorticity



Landscapes

Used by permission of B. Wright, Visible Decisions Inc.

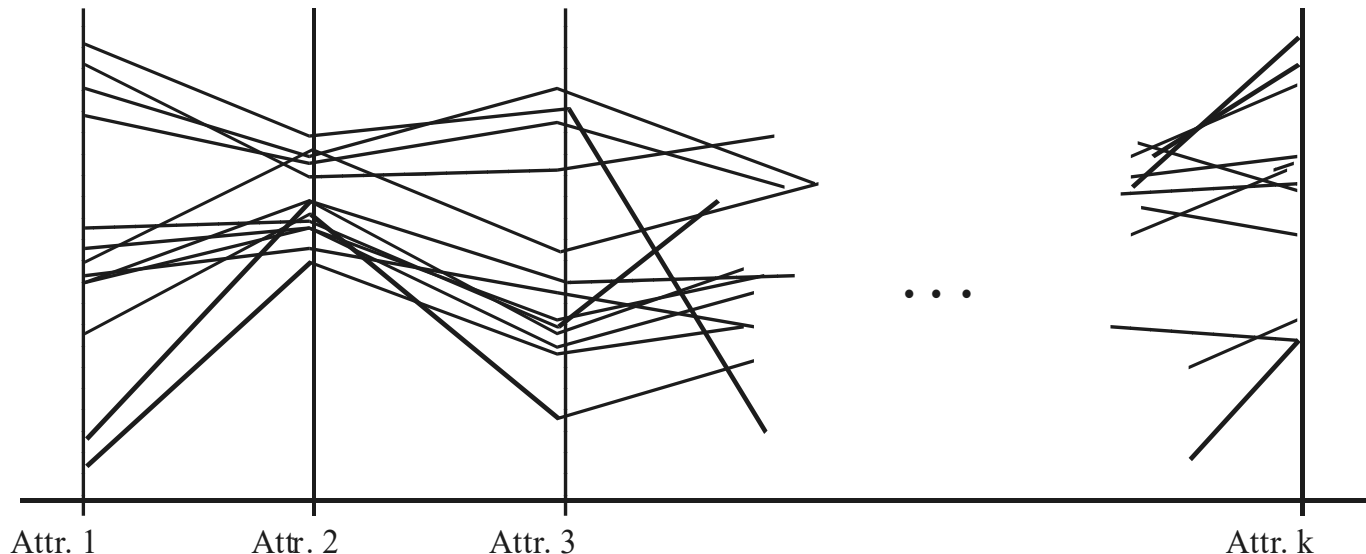


news articles
visualized as
a landscape

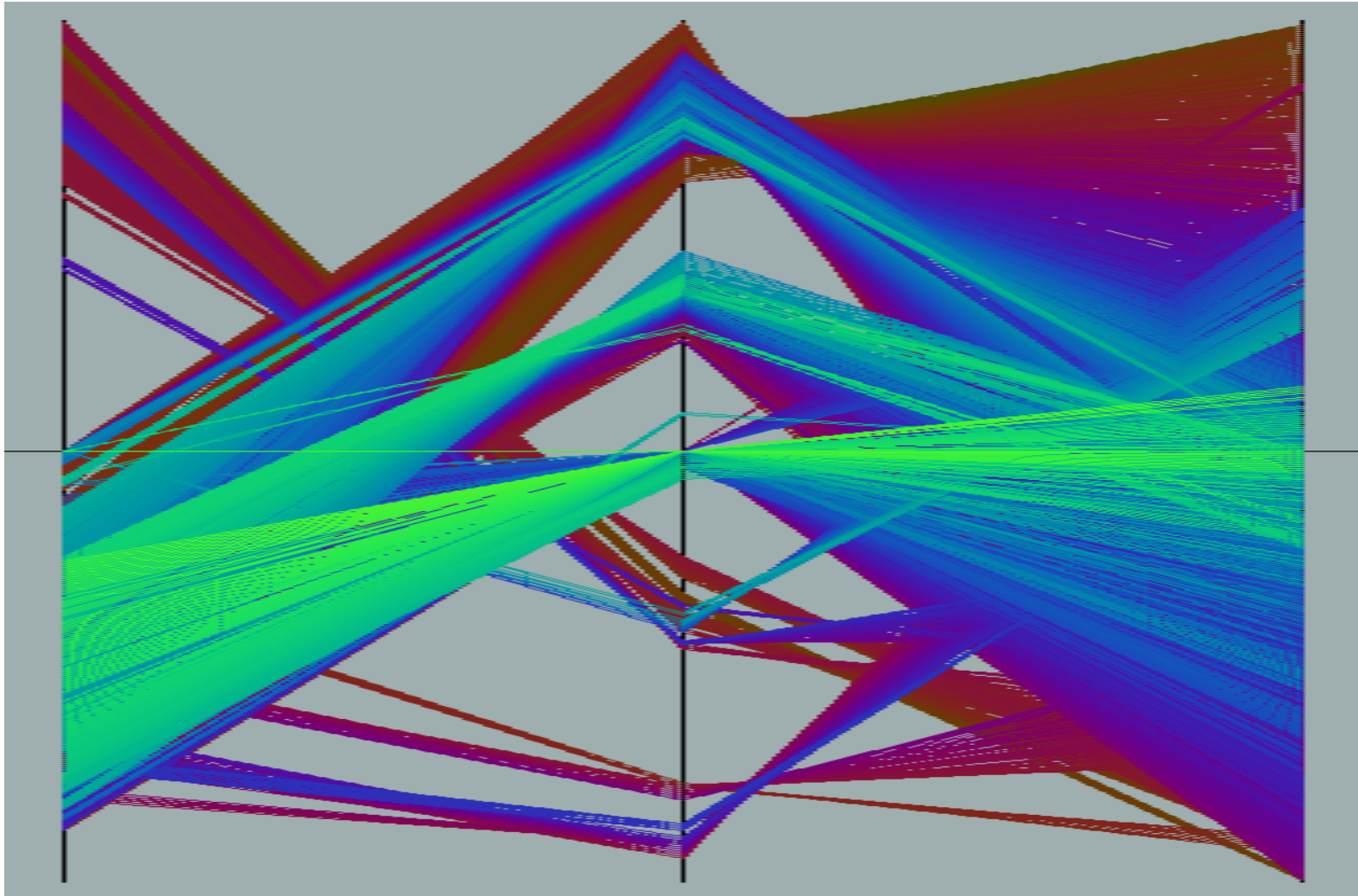
- Visualization of the data as perspective landscape
- The data needs to be transformed into a (possibly artificial) 2D spatial representation which preserves the characteristics of the data

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



Parallel Coordinates of a Data Set

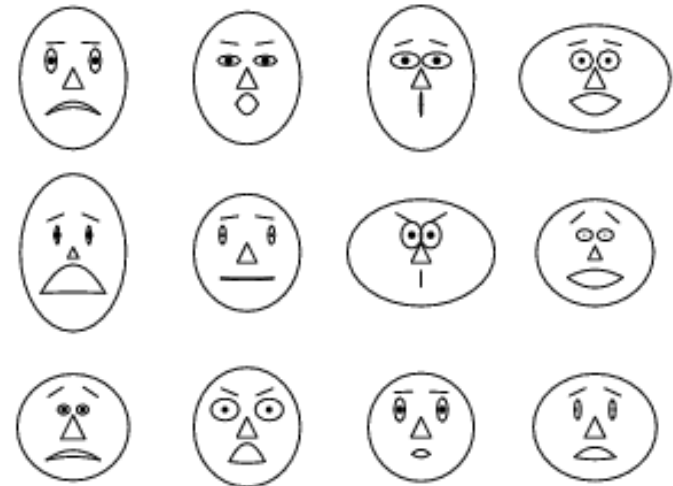


Icon-Based Visualization Techniques

- 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



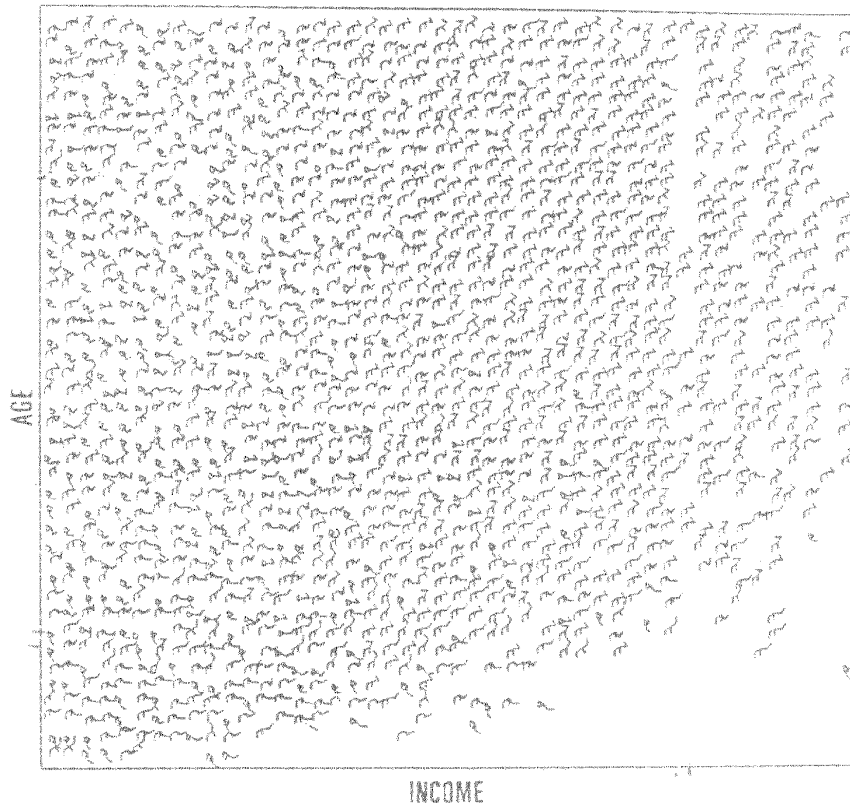
Stick Figure Visualization Technique

- Maps multidimensional data to five-piece stick figures, where each figure has four limbs and a body.
- Two dimensions are mapped to the display (x and y) axes and the remaining dimensions are mapped to the angle and/or length of the limbs.

Stick Figure

- Shows census data, where *age* and *income* are mapped to the display axes, and the remaining dimensions (*gender*, *education*, and so on) are mapped to stick figures.
- If the data items are relatively dense with respect to the two display dimensions, the resulting visualization shows texture patterns, reflecting data trends.

used by permission of G. Grinstein, University of Massachusetts at Lowell



Two attributes mapped to axes, remaining attributes mapped to angle or length of limbs". Look at texture pattern

Hierarchical Visualization Techniques

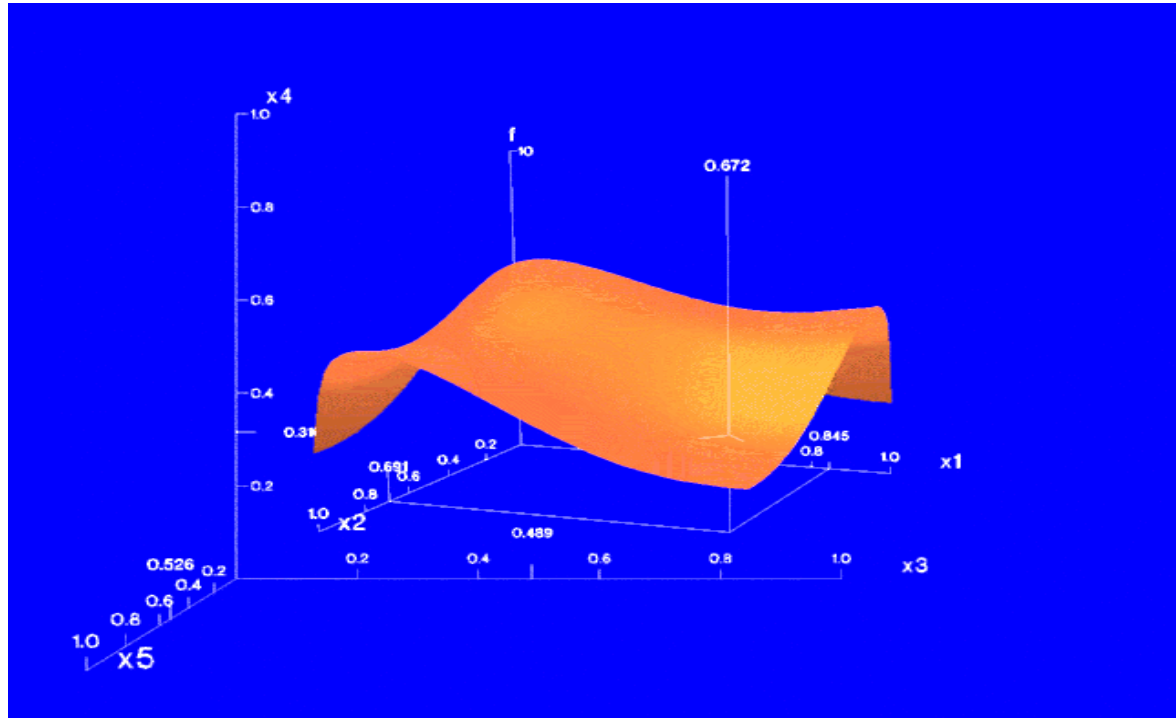
- Visualization of the data using a hierarchical partitioning into subspaces
- Methods
 - Worlds-within-Worlds
 - Tree-Map

Worlds-within-Worlds

- Suppose we want to visualize a 6-D data set, where the dimensions are F, X_1, \dots, X_5 .
- We want to observe how dimension F changes with respect to the other dimensions.
- Fix the values of dimensions X_3, X_4, X_5 to some selected values, say, c_3, c_4, c_5 .
- Visualize F, X_1, X_2 using a 3-D plot, called a **world**.
- The position of the origin of the inner world is located at the point (c_3, c_4, c_5) in the outer world, which is another 3-D plot using dimensions X_3, X_4, X_5 .

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-Maps

- Display hierarchical data as a set of nested rectangles.
- Figure shows a tree-map visualizing Google news stories.
- All news stories are organized into seven categories, each shown in a large rectangle of a unique color.
- Within each category (i.e., each rectangle at the top level), the news stories are further partitioned into smaller subcategories.



- Visualizing non-numerical data: text and social networks
- Tag cloud: visualizing user-generated tags

- [illegible]

Newsmap: Google News Stories in 2005

Visualizing Complex Data and Relations

- In early days, visualization techniques were mainly for numeric data.
- Recently, more and more non-numeric data, such as text and social networks, have become available.
- Visualizing and analyzing such data attracts a lot of interest.
- There are many new visualization techniques dedicated to these kinds of data.

Tag Cloud

- Many people on the Web tag various objects such as pictures, blog entries, and product reviews.
- A **tag cloud** is a visualization of statistics of user-generated tags.
- Often, in a tag cloud, tags are listed alphabetically or in a user-preferred order.
- The importance of a tag is indicated by font size or color.
- Figure shows a tag cloud for visualizing the popular tags used in a Web site.



animals architecture **art** asia australia autumn baby band barcelona **beach** berlin bike bird
birds **birthday** black blackandwhite blue bw **california** canada **canon** car cat
chicago china christmas church **city** clouds color concert cute dance day de dog
england europe fall **family** fashion festival film florida flower flowers food
football france friends fun garden geotagged germany girl girls graffiti green
halloween hawaii holiday home house india iphone ireland island italia **italy** japan july kids la
lake landscape light live london love macro me mexico model mountain mountains museum
music nature new newyork newyorkcity night **nikon** nyc ocean old paris
park **party** people photo photography photos portrait red river rock san
sanfrancisco scotland sea seattle show sky snow spain spring street summer
sun sunset taiwan texas thailand tokyo toronto tour **travel** tree trees trip uk urban
usa vacation washington water **wedding** white winter yellow york zoo

Tag Cloud

- Tag clouds are often used in two ways.
 - In a tag cloud for a single item, we can use the size of a tag to represent the number of times that the tag is applied to this item by different users.
 - When visualizing the tag statistics on multiple items, we can use the size of a tag to represent the number of items that the tag has been applied to.



Weka – Using CSV Files

CSV Files

- CSV files could be easily exported from Microsoft Excel or Google Spreadsheet

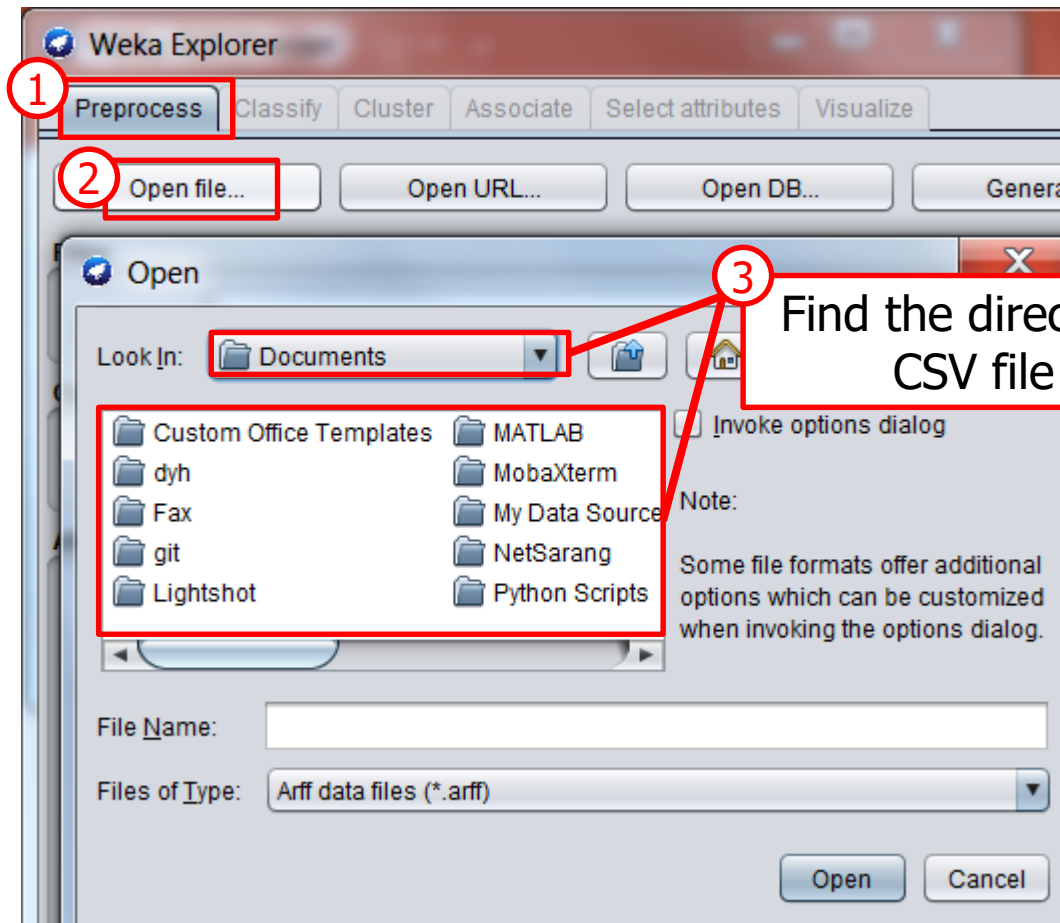
	A	B	C	D	E
1	outlook	temperature	humidity	windy	play
2	sunny	85	85	FALSE	no
3	sunny	80	90	TRUE	no
4	overcast	83	86	FALSE	yes
5	rainy	70	96	FALSE	yes
6	rainy	68	80	FALSE	yes
7	rainy	65	70	TRUE	no

```
outlook,temperature,humidity,windy,play
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
```

Download a Sample CSV File

- <http://kdd.snu.ac.kr/weka/>
 - Download the **test.csv**

Using CSV Files for Weka



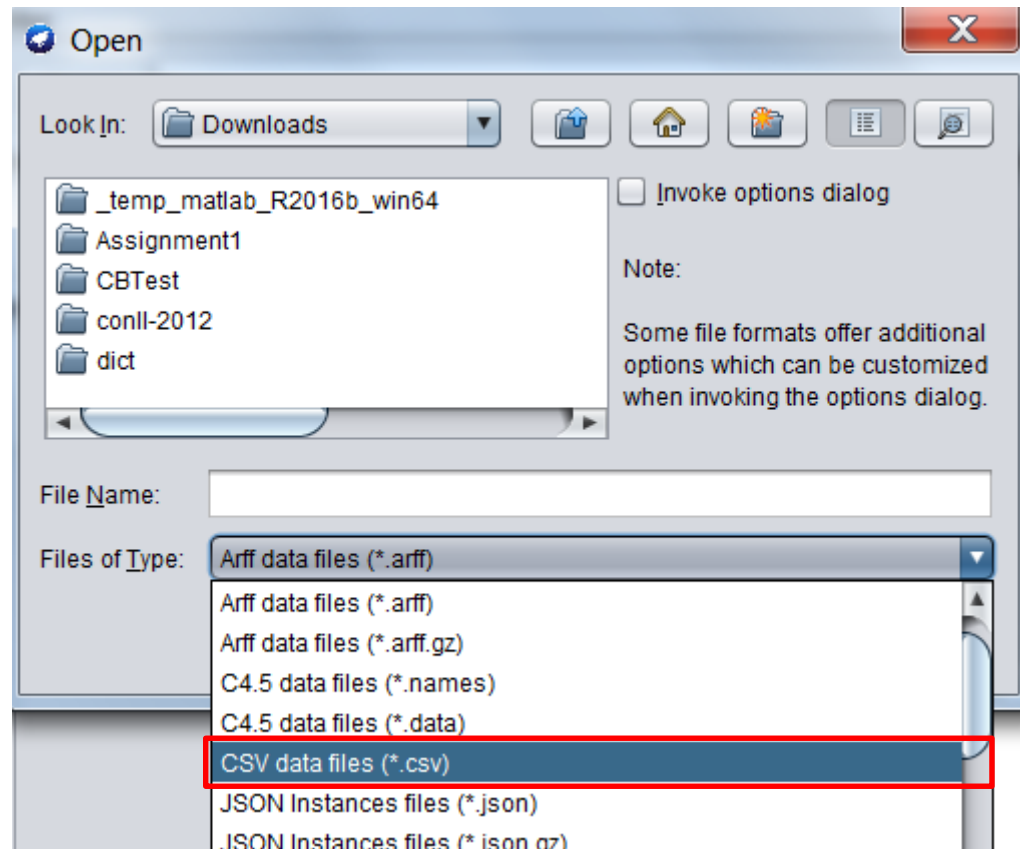
Using CSV Files for Weka

- IF the test.csv is downloaded in C:\Users\dyhong\Downloads



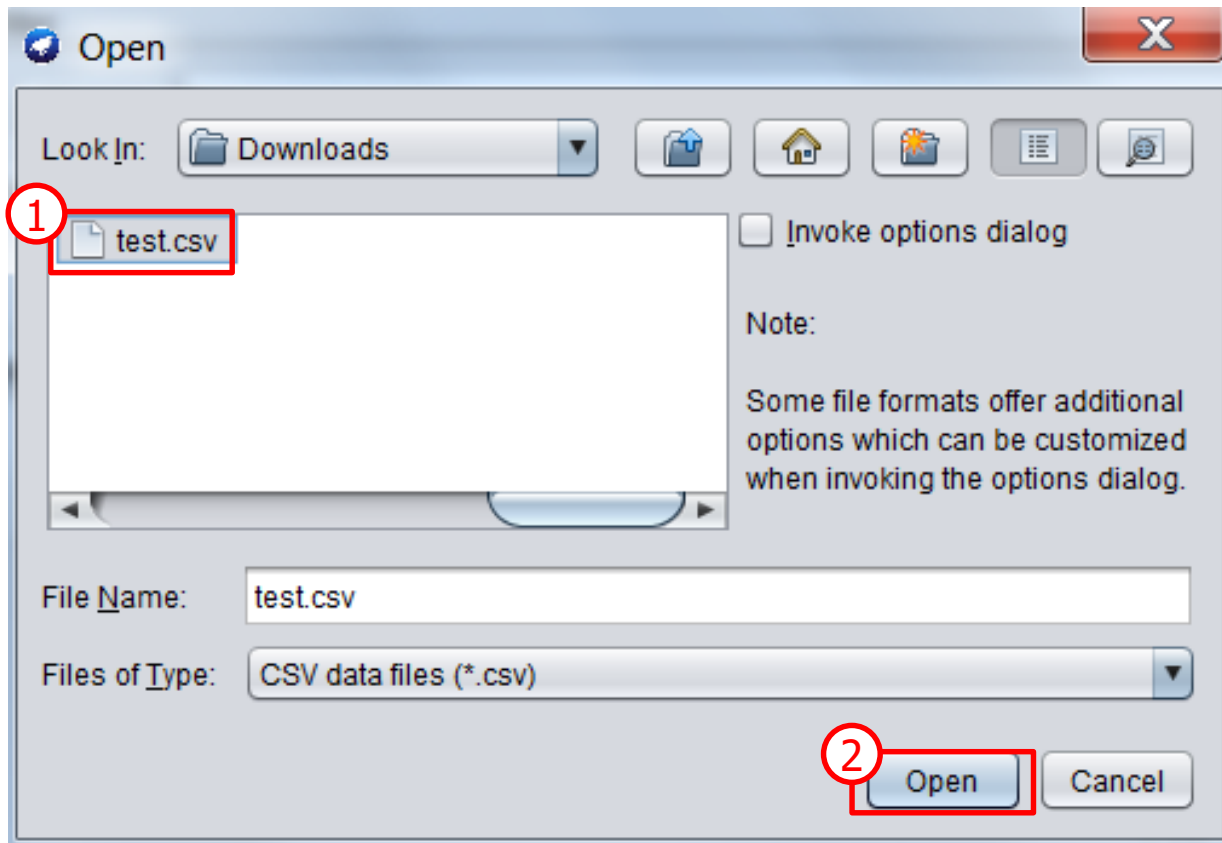
Using CSV Files for Weka

- IF the test.csv is downloaded in C:\Users\dyhong\Downloads



Using CSV Files for Weka

- IF the test.csv is downloaded in C:\Users\dyhong\Downloads



Weka Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open UR...

Open DB...

Generate...

Undo

Edit...

Save...

Filter

Choose

No

The data is loaded successfully

Apply

Stop

Current relation

Relation: test

Instances: 14

Attributes: 5

Sum of weights: 14

Attributes

All

None

Invert

Pattern

No.		Name
1	<input checked="" type="checkbox"/>	outlook
2	<input type="checkbox"/>	temperature
3	<input type="checkbox"/>	humidity
4	<input type="checkbox"/>	windy
5	<input type="checkbox"/>	play

Remove

Selected attribute

Name: outlook

Missing: 0 (0%)

Distinct: 3

Type: Nominal

Unique: 0 (0%)

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

Class: play (Nom)

Visualize All

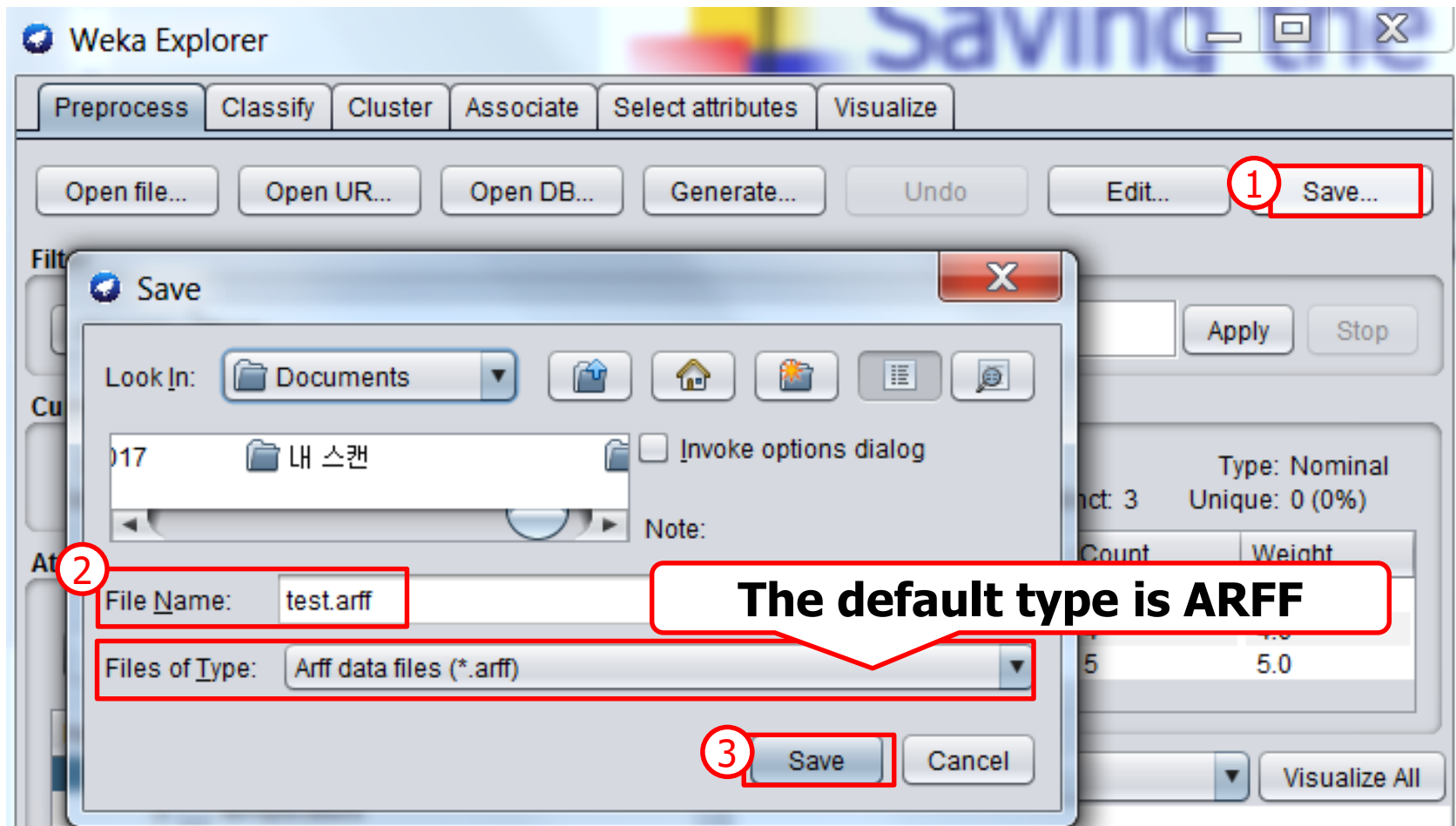
5

4

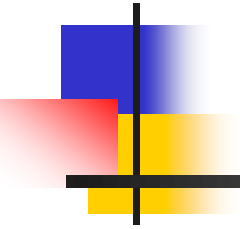
5

Status

Saving the CSV File as ARFF File

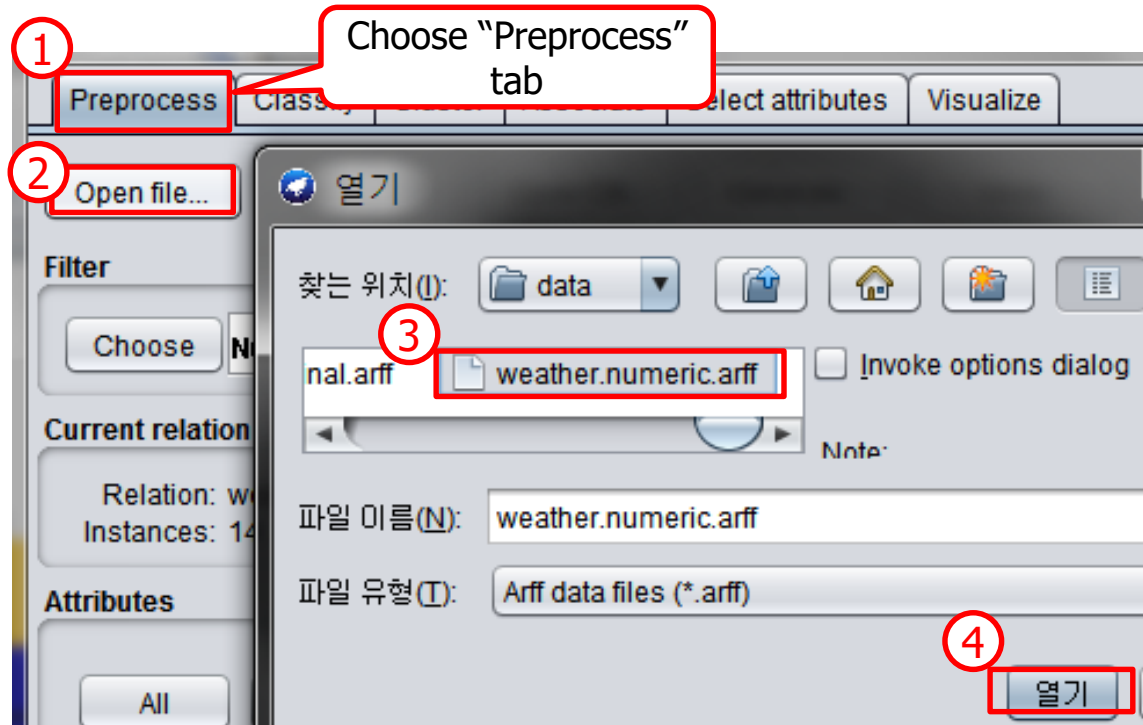


Weka - Statistics of the Dataset



Open the Dataset

- C:\Program Files\Weka-3-8\data\weather.numeric.arff



The Statistics of the Dataset

The sum of weights of instances, which is initially equal to the number of instances since the initial weight of each instance has 1

The name of the dataset

The number of attributes

The number of instances

Current relation

Relation: weather
Instances: 14

Attributes: 5
Sum of weights: 14

Attributes

All None Invert Patte...

No.		Name
1	<input checked="" type="checkbox"/>	outlook
2	<input type="checkbox"/>	temperature
3	<input type="checkbox"/>	humidity
4	<input type="checkbox"/>	windy

Selected attribute

Name: outlook
Missing: 0 (0%)

No.	Label
1	sunny
2	overcast
3	rainy

Class: play (Nom)

Select the Class

Current relation
Relation: weather
Instances: 14
Attributes: 5
Sum of weights: 14

Attributes

AllNoneInvertPattern

No.		Name
1	<input type="checkbox"/>	outlook
2	<input type="checkbox"/>	temperature
3	<input type="checkbox"/>	humidity
4	<input type="checkbox"/>	windy
5	<input checked="" type="checkbox"/>	play

Remove

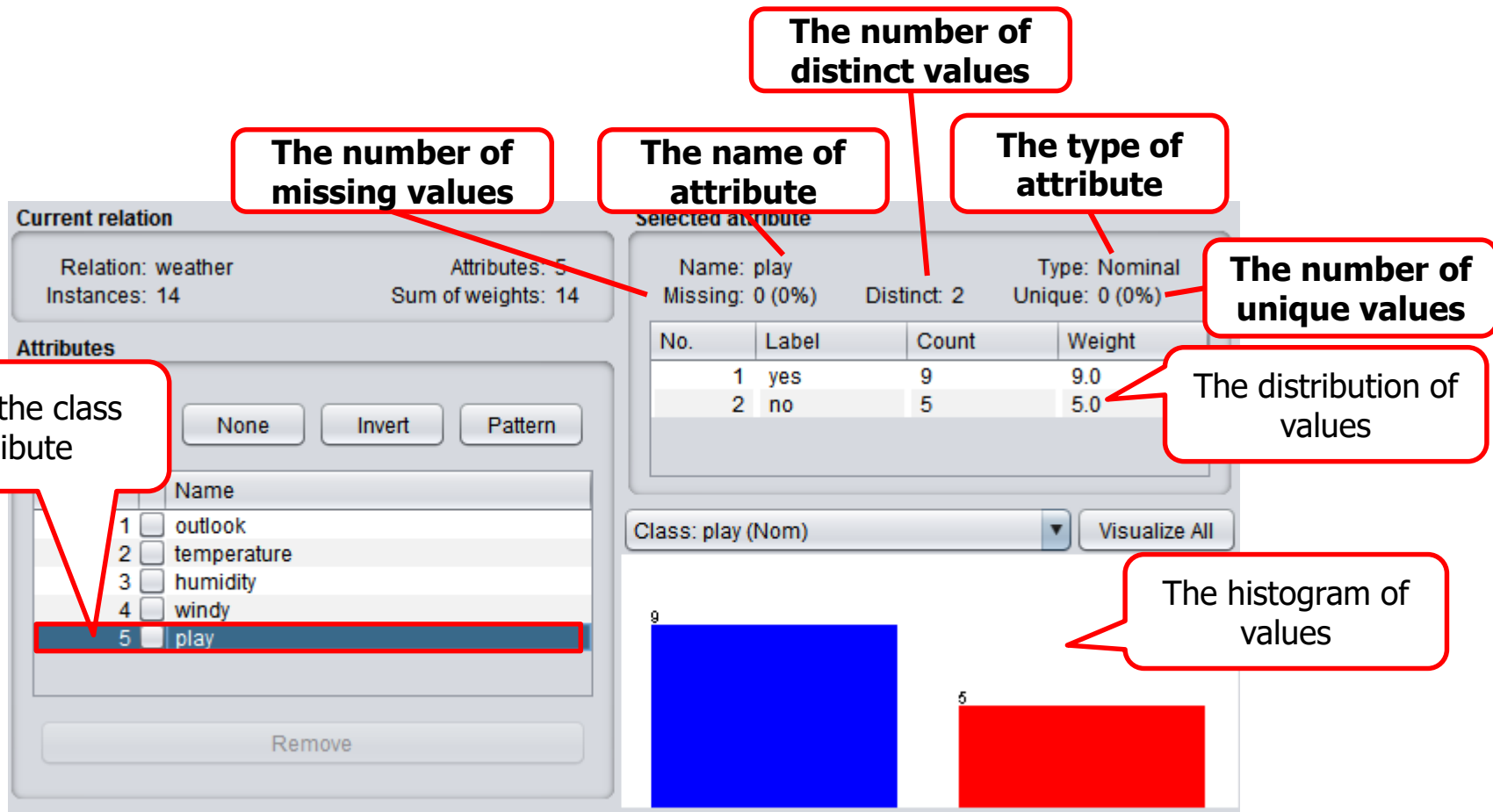
Selected attribute
Name: play
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	yes	9	9.0
2	no	5	5.0

Class: play (Nom)
No class
Class: outlook (Nom)
Class: temperature (Num)
Class: humidity (Num)
Class: windy (Nom)
Class: play (Nom)

The last attribute is selected as default

The Statistics of the Class



The Statistics of an Attribute

Current relation

Relation: weather
Instances: 14

Attributes: 5
Sum of weights: 14

None Invert Pattern

No.	Name
1	<input checked="" type="checkbox"/> outlook
2	<input type="checkbox"/> temperature
3	<input type="checkbox"/> humidity
4	<input type="checkbox"/> windy
5	<input type="checkbox"/> play

Remove

Selected attribute

Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

Class: play (Nom)

Visualiz

Label	Count
sunny	5
overcast	4
rainy	5

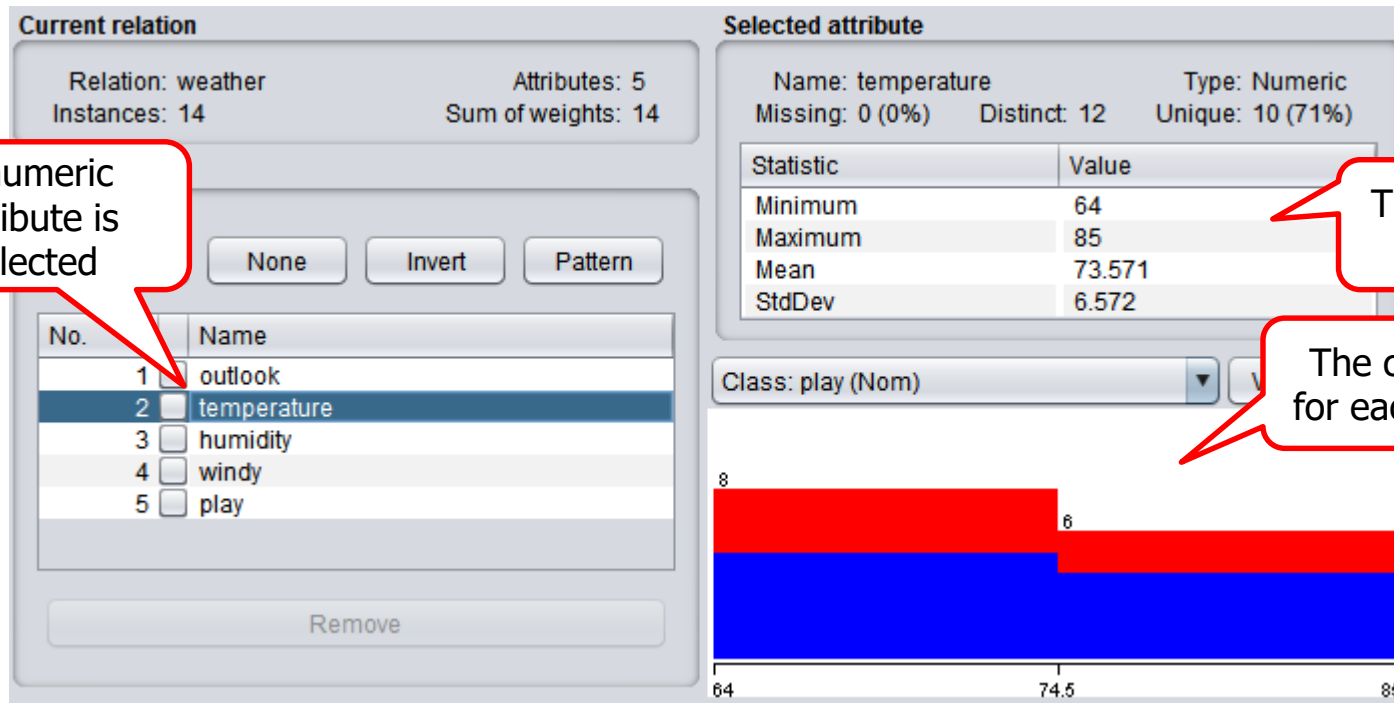
A nominal attribute is selected

The distribution of values

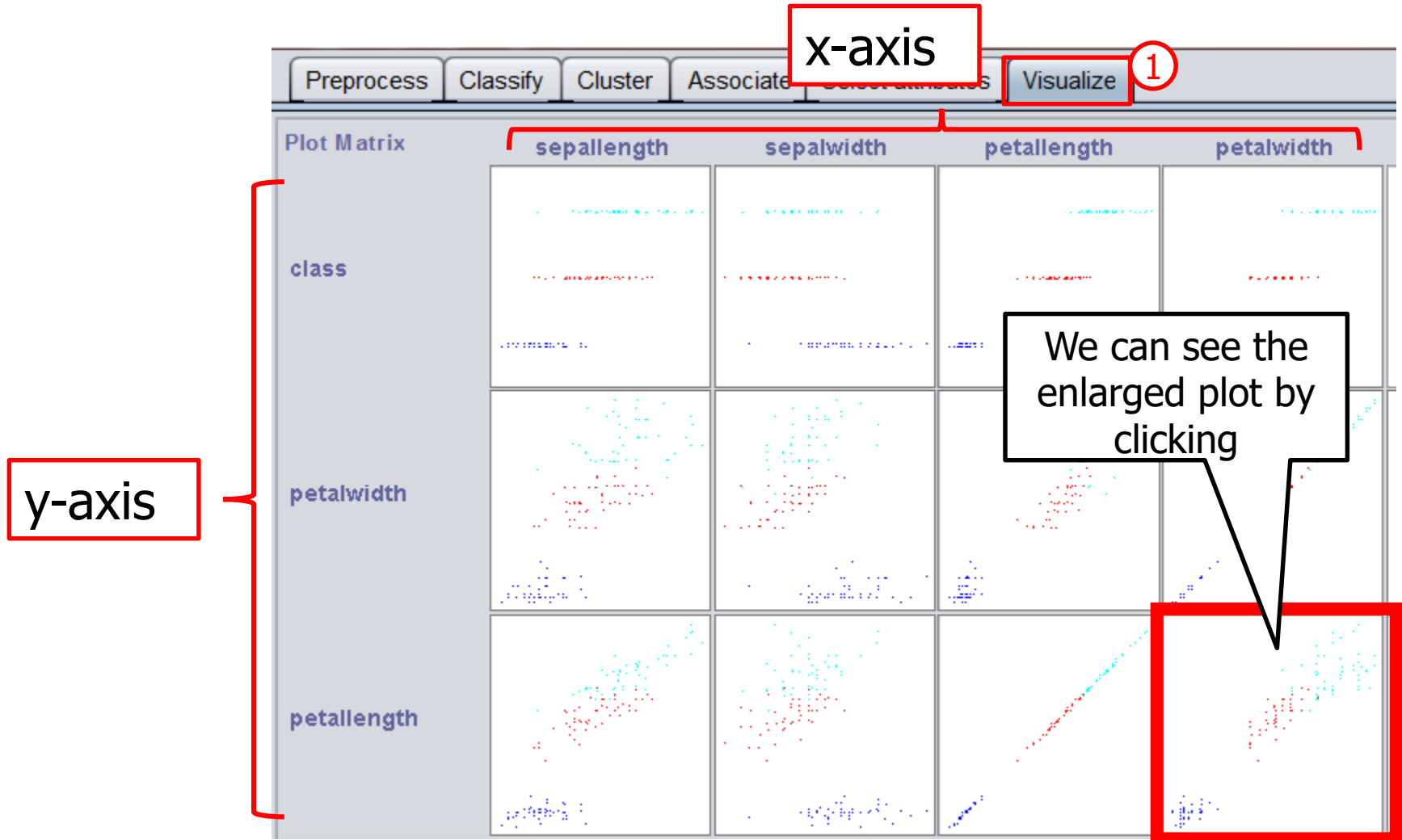
The class distribution of each value

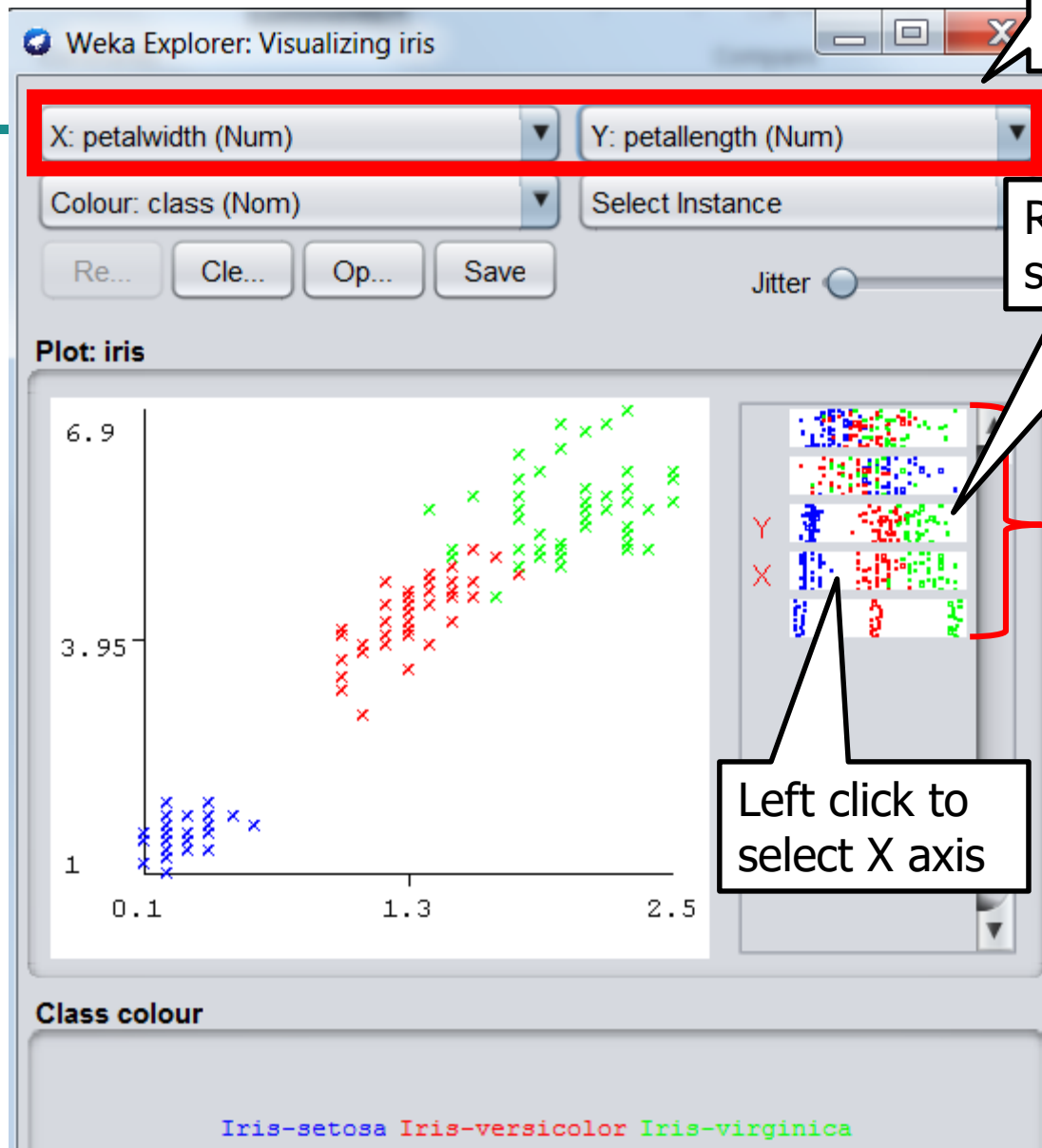
The class of all the 'overcast' instances are 'yes'

The Statistics of an Attribute



Visualize the Data





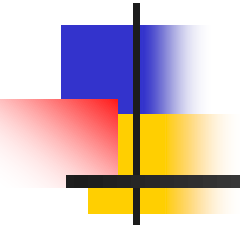
We can change the axes

Right click to select X axis

Left click to select X axis

Represents each attribute (point distribution colored by class for the attribute)

Python - Statistics of the Dataset



Basic statistics - Python

← → ↻ ⓘ 주의 요함 | kdd.snu.ac.kr/python/

- Anaconda installers
 - Windows: [64-Bit](#) [32-Bit](#)
 - macOS: [64-Bit](#)
 - Linux [64-Bit](#) [32-Bit](#)
- Weka installer (Windows)
 - [Weka installer](#)
- Graphviz
 - conda
 - graphviz
 - [Windows 64-bit](#)
 - [OSX 64-bit](#)
 - [Linux 64-bit](#)
 - [python-graphviz](#)
 - pip
 - [whl file](#)
 - Installer
 - [Windows](#)
- ARFF
 - [arff-0.9.tar.gz](#)
- Datasets
 - [cluster2.csv](#)
 - [cluster2.arff](#)
 - [diabetes.csv](#)
 - [glass.csv](#)
 - [glass_missing.csv](#)
 - [iris.csv](#)
 - [cpu.csv](#)
 - [vote.csv](#)

■ Download 'iris.csv' from
kdd.snu.ac.kr/python

Basic statistics - Python

```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

- pandas : A library for data analysis
- numpy : A popular library for vector/matrix operations
- matplotlib : A library for visualization

Basic statistics - Python

```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

We will use 'plt'
instead of long name

- pandas : A library for data
- numpy : A popular library for vector/matrix operations
- matplotlib : A library for visualization

Basic statistics - Python

```
In [2]: df = pd.read_csv('iris.csv')  
df
```

Out [2]:

	sepalength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa

- Load the iris data with pandas

Basic statistics - Python

```
In [2]: df = pd.read_csv('iris.csv')  
df
```

Out[2]:

	sepalength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa

In Jupyter Notebook, if type an object in the end of a shell, it is shown under the shell

- Load the iris data with pandas

Basic statistics - Python

```
In [3]: X = df.values[:, :-1].astype(np.float32)  
        y = df.values[:, -1]
```

':' in the first index means 'all rows'

- Convert pandas dataframe object into numpy arrays
- We will separate attributes (X) and class (y)

Basic statistics - Python

```
In [3]: X = df.values[:, :-1].astype(np.float32)  
        y = df.values[:, -1]
```

':-1' in the second index means
'all columns before the last row'

- Convert pandas dataframe object into numpy arrays
- We will separate attributes (X) and class (y)

Basic statistics - Python

```
In [3]: X = df.values[:, :-1].astype(np.float32)
        y = df.values[:, -1]
```

This means that the type of numpy array is converted into 32bit float type

- Convert pandas dataframe object into numpy arrays
- We will separate attributes (X) and class (y)

Basic statistics - Python

```
In [3]: X = df.values[:, :-1].astype(np.float32)
        y = df.values[:, -1]
```

- Convert pandas dataframe object into numpy arrays
- We will separate attributes (X) and class (y)

Basic statistics - Python

```
In [4]: X[:5]
```

```
Out[4]: array([[5.1, 3.5, 1.4, 0.2],  
               [4.9, 3. , 1.4, 0.2],  
               [4.7, 3.2, 1.3, 0.2],  
               [4.6, 3.1, 1.5, 0.2],  
               [5. , 3.6, 1.4, 0.2]], dtype=float32)
```

```
In [5]: y[:5]
```

```
Out[5]: array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',  
               'Iris-setosa'], dtype=object)
```

- Check that the data is properly separated

	sepalength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Basic statistics - Python

```
In [4]: X[:5]
```

All rows before index 5

```
Out [4]: array([[5.1, 3.5, 1.4, 0.2],
               [4.9, 3. , 1.4, 0.2],
               [4.7, 3.2, 1.3, 0.2],
               [4.6, 3.1, 1.5, 0.2],
               [5. , 3.6, 1.4, 0.2]], dtype=float32)
```

```
In [15]: fig = plt.figure(figsize = (5,10))
          plt.boxplot(X)
          plt.show()
```

```
In [5]: y[:5]
```

```
Out [5]: array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
               'Iris-setosa'], dtype=object)
```

- Check that the data is properly separated

	sepalength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Basic statistics - Python

```
In [6]: X.max(axis = 0)
```

```
Out [6]: array([7.9, 4.4, 6.9, 2.5], dtype=float32)
```

- `max()` method returns the maximum value

Basic statistics - Python

```
In [6]: X.max(axis = 0)
```

```
Out[6]: array([7.9, 4.4, 6.9, 2.5], dtype=float32)
```

- `max()` method returns the maximum value
- `'axis = 0'` option means that return the maximum value of each column

Basic statistics - Python

```
In [6]: X.max(axis = 0)
```

```
Out [6]: array([7.9, 4.4, 6.9, 2.5], dtype=float32)
```

- `max()` method returns the maximum value
- `'axis = 0'` option means that return the maximum value of each column
 - If not specified, one value will be returned which is the maximum value of the whole matrix
 - If `'axis = 1'`, the maximum values among attributes of each row are returned

Basic statistics - Python

```
In [7]: print(X[:, 0].max())  
        print(X[:, 2].max())
```

7.9

6.9

- If you want the maximum value of a specific column i , use `X[:, i].max()`

Basic statistics - Python

```
In [8]: X.min(axis = 0)
```

```
Out[8]: array([4.3, 2. , 1. , 0.1], dtype=float32)
```

- You can use `X.min()` in the same manner

Basic statistics - Python

```
In [9]: np.percentile(X, 70, axis=0)
```

```
Out[9]: array([6.300000019, 3.200000005, 5.          , 1.799999995])
```

- `numpy.percentile(X, k, axis=0)` gives the k th percentile value ($\lceil k * N / 100 \rceil$ th value in increasing order)

Basic statistics - Python

50th percentile = median

0]:

```
print(np.percentile(X, 50, axis=0), np.median(X, axis = 0))  
print(np.percentile(X, 100, axis=0), X.max(axis = 0))  
print(np.percentile(X, 0, axis=0), X.min(axis = 0))
```

```
[5.80000019 3.         4.35000014 1.299999995] [5.8         3.         4.3500004 1.3         ]  
[7.90000001 4.40000001 6.90000001 2.5         ] [7.9 4.4 6.9 2.5]  
[4.300000019 2.         1.         0.1         ] [4.3 2. 1. 0.1]
```

minimum

maximum

101

- There are 99 percentiles formally
- Numpy provides 0(minimum) and 100(maximum) for convenience

Basic statistics - Python

```
In [11]: X.mean(axis = 0)
```

```
Out[11]: array([5.8433332, 3.0540001, 3.758667 , 1.1986667], dtype=float32)
```

```
In [12]: X.std(axis = 0)
```

```
Out[12]: array([0.8253013 , 0.43214658, 1.7585291 , 0.7606126 ], dtype=float32)
```

```
In [13]: X.var(axis = 0)
```

```
Out[13]: array([0.68112224, 0.18675067, 3.0924246 , 0.57853156], dtype=float32)
```

- Mean, standard deviation, variance

Basic statistics - Python

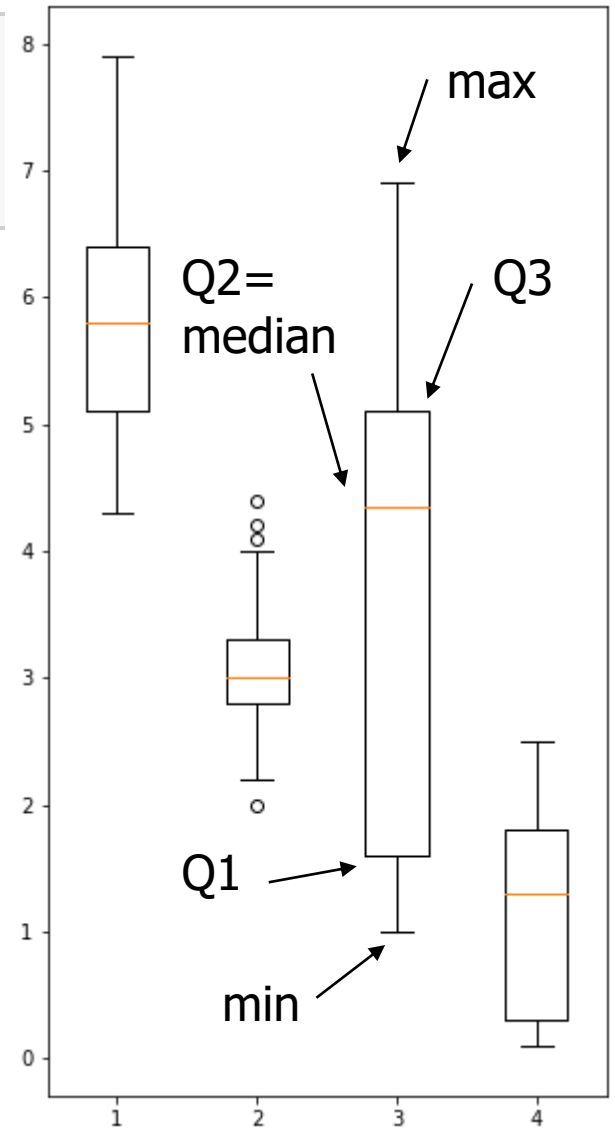
```
In [15]: plt.figure(figsize = (5,10))  
plt.boxplot(X)  
plt.show()
```

Size of the figure in inches

- You can draw boxplots using `matplotlib.pyplot`

Basic statistics - Python

```
In [15]: plt.figure(figsize = (5,10))  
plt.boxplot(X)  
plt.show()
```



Basic statistics - Python

```
In [15]: plt.figure(figsize = (5,10))  
plt.boxplot(X)  
plt.show()
```

Outliers : $> Q3 + 1.5 \cdot IQR$ OR
 $< Q1 - 1.5 \cdot IQR$

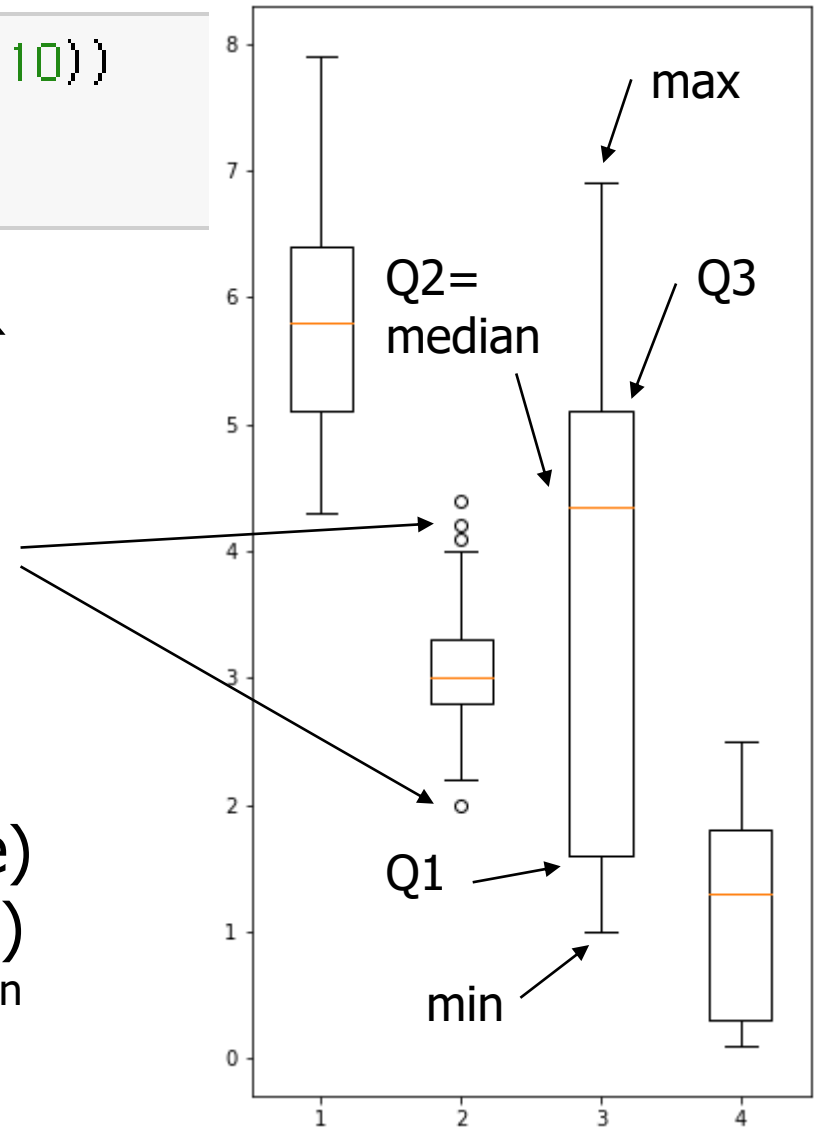
where

IQR(interquartile range)
 $= Q3 - Q1$

Q3 : 3rd quartile (75th percentile)

Q1 : 1st quartile (25th percentile)

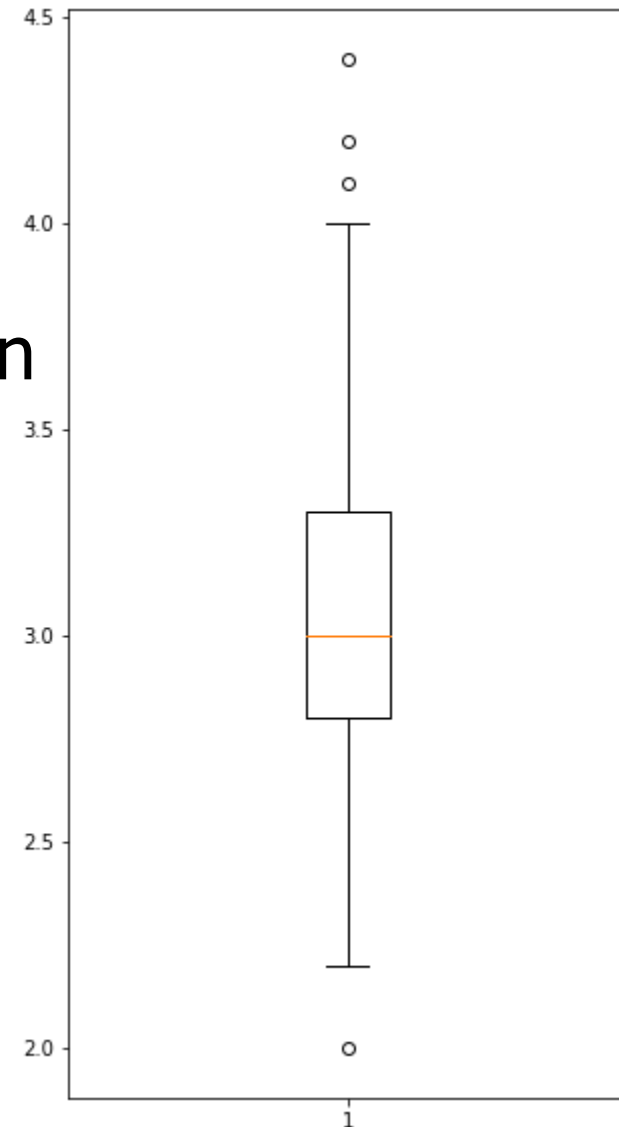
*outliers are excluded from selecting max and min



Basic statistics - Python

```
In [16]: plt.figure(figsize = (5,10))  
plt.boxplot(X[:, 1])  
plt.show()
```

- To draw the boxplot of one column



Basic statistics - Python

```
In [17]: columns = df.columns[:-1]
          #['sepal.length', 'sepal.width', 'petal.length', 'petal.width']
          plt.figure(figsize = (7,7))
          plt.xlabel(columns[0])
          plt.ylabel(columns[1])
          plt.scatter(X[:, 0], X[:, 1])
          plt.show()
```

- matplotlib.pyplot also supports scatter plot

Basic statistics - Python

```
In [17]: columns = df.columns[:-1]
          #['sepal.length', 'sepal.width', 'petal.length']
          plt.figure(figsize = (7,7))
          plt.xlabel(columns[0])
          plt.ylabel(columns[1])
          plt.scatter(X[:, 0], X[:, 1])
          plt.show()
```

Get column names from the dataframe

- matplotlib.pyplot also supports scatter plot

Basic statistics - Python

```
In [17]: columns = df.columns[:-1]
          #['sepal.length', 'sepal.width', 'petal.length', 'petal.width']
          plt.figure(figsize = (7,7))
          plt.xlabel(columns[0])
          plt.ylabel(columns[1])
          plt.scatter(X[:, 0], X[:, 1])
          plt.show()
```

Give labels to x axis and y axis

- matplotlib.pyplot also supports scatter plot

Basic statistics - Python

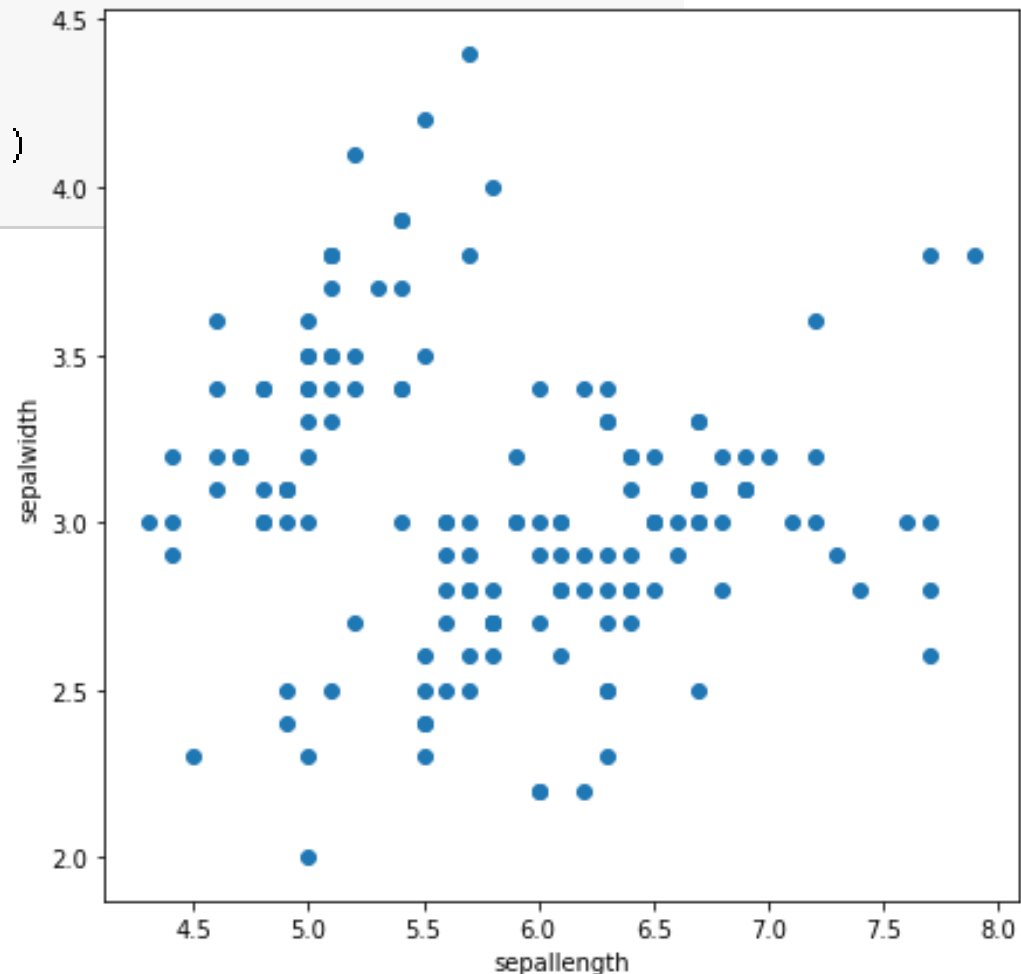
```
In [17]: columns = df.columns[:-1]
          #['sepal.length', 'sepal.width', 'petal.length', 'petal.width']
          plt.figure(figsize = (7,7))
          plt.xlabel(columns[0])
          plt.ylabel(columns[1])
          plt.scatter(X[:, 0], X[:, 1])
          plt.show()
```

Input two columns you want to compare

- matplotlib.pyplot also supports scatter plot

Basic statistics - Python

```
In [17]: columns = df.columns[:-1]
          #['sepalength', 'sepalwidth', 'petallength', 'petalwidth']
          plt.figure(figsize = (7,7))
          plt.xlabel(columns[0])
          plt.ylabel(columns[1])
          plt.scatter(X[:, 0], X[:, 1])
          plt.show()
```



Basic statistics - Python

```
In [18]: fig, axs = plt.subplots(4, 4, figsize = (20,20))
         for i in range(4):
             for j in range(4):
                 axs[i, j].set(xlabel=columns[i], ylabel=columns[j])
                 axs[i, j].scatter(X[:, i], X[:, j])

         plt.show()
```

- You can plot multiple graphs at once

Basic statistics - Python

```
In [18]: fig, axs = plt.subplots(4, 4, figsize = (20,20))
         for i in range(4):
             for j in range(4):
                 axs[i, j].set(xlabel=columns[i], ylabel=columns[j])
                 axs[i, j].scatter(X[:, i], X[:, j])

         plt.show()
```

- You can plot multiple graphs at once
- fig is the object of entire plot
- axs is the matrix of objects of subplots

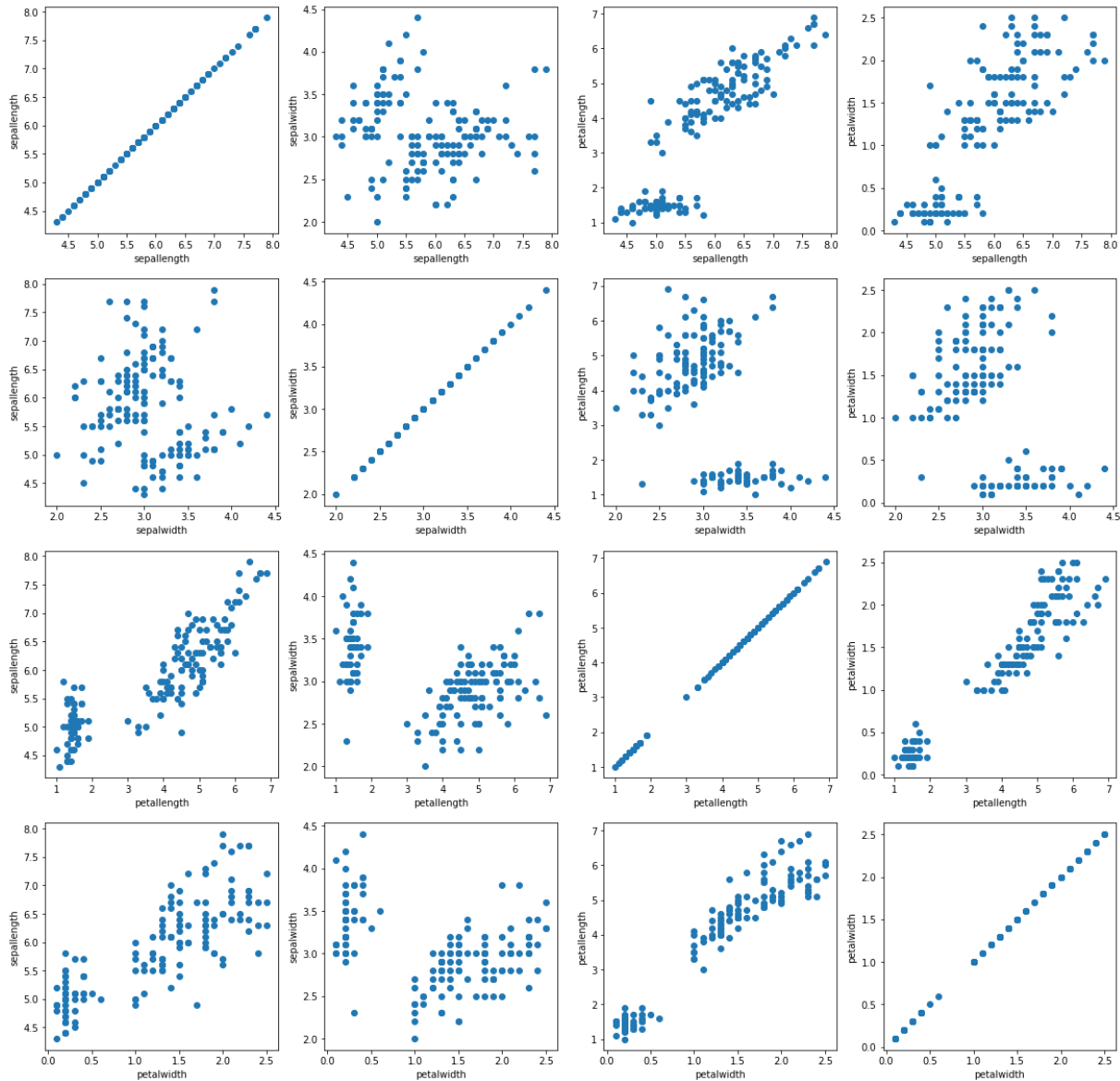
Basic statistics - Python

```
In [18]: fig, axs = plt.subplots(4, 4, figsize = (20,20))
         for i in range(4):
             for j in range(4):
                 axs[i, j].set(xlabel=columns[i], ylabel=columns[j])
                 axs[i, j].scatter(X[:, i], X[:, j])

         plt.show()
```

- You can plot multiple graphs at once
- fig is the object of entire plot
- axs is the matrix of objects of subplots
- Above code plots all-pair scatter plots

Basic statistics - Python



Practice

- Download [glass.csv](http://kdd.snu.ac.kr/python/glass.csv) from <http://kdd.snu.ac.kr/python/>
- Plot the scatter-plot matrix of 'glass.csv', using 1st to 5th columns

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



Measuring Data Similarity and Dissimilarity

- In data mining applications, we need ways to assess how alike or unlike objects are in comparison to one another.
- For example, a store may want to search for clusters of **customer** objects, resulting in groups of customers with similar characteristics (e.g., similar income, area of residence, and age).
- Outlier analysis also employs clustering-based techniques to identify potential outliers as objects that are highly dissimilar to others.
- Knowledge of object similarities can also be used in nearest-neighbor classification schemes where a given object (e.g., **a patient**) is assigned a class label (relating to, say, a **diagnosis**) based on its similarity toward other objects in the model.

Similarity and Dissimilarity Measures

- A similarity measure for two objects, i and j , will typically return the value 0 if the objects are unlike.
- The higher the similarity value, the greater the similarity between objects. (Typically, a value of 1 indicates complete similarity, that is, the objects are identical.)
- A dissimilarity measure works the opposite way. It returns a value of 0 if the objects are the same (and therefore, far from being dissimilar).
- The higher the dissimilarity value, the more dissimilar the two objects are.

Data Matrix versus Dissimilarity Matrix

- Suppose that we have n objects (e.g., persons, items, or courses) described by p attributes (also called measurements or features, such as age, height, weight, or gender).
- The objects are $x_1 = (x_{11}, x_{12}, \dots, x_{1p})$, $x_2 = (x_{21}, x_{22}, \dots, x_{2p})$, and so on, where x_{ij} is the value for object x_i of the j -th attribute.
- For brevity, we hereafter refer to object x_i as object i .
- The objects may be tuples in a relational database, and are also referred to as **data samples** or **feature vectors**.

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

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

■ Dissimilarity matrix

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

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

Data Matrix and Dissimilarity Matrix

- Measures of similarity can often be expressed as a function of measures of dissimilarity.
- For example, for nominal data,
 - $sim(i, j) = 1 - d(i, j)$
where $sim(i, j)$ is the similarity between objects i and j .
- A data matrix is made up of two entities or “things,” namely rows (for objects) and columns (for attributes).
 - Therefore, the data matrix is often called a **two-mode** matrix.
- The dissimilarity matrix contains one kind of entity (dissimilarities) and so is called a **one-mode** matrix.
- Many clustering and nearest-neighbor algorithms operate on a dissimilarity matrix.
- Data in the form of a data matrix can be transformed into a dissimilarity matrix before applying such algorithms.

Proximity Measures for Nominal Attributes

- A nominal attribute can take on two or more states.
- e.g.,
 - **Map color** is a nominal attribute that may have, say, five states: red, yellow, green, pink, and blue.
 - Let the number of states of a nominal attribute be M .
 - The states can be denoted by letters, symbols, or a set of integers, such as $1, 2, \dots, M$.
 - Notice that such integers are used just for data handling and do not represent any specific ordering.

Proximity Measures for Nominal Attributes (Method 1)

- The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

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

- where m is the number of matches (i.e., the number of attributes for which i and j are in the same state)
 - p is the total number of attributes describing the objects.
- Weights can be assigned to increase the effect of m or to assign greater weight to the matches in attributes having a larger number of states.
- Alternatively, similarity can be computed as

$$sim(i, j) = 1 - d(i, j) = \frac{m}{p}$$

Dissimilarity between nominal attributes

- Suppose that we have the following sample data

<i>Object Identifier</i>	<i>test-1 (nominal)</i>
1	code A
2	code B
3	code C
4	code A

- where test-1 is nominal.
- Since here we have one nominal attribute, test-1, we set $p = 1$ in Eq. (2.11) so that $d(i, j)$ evaluates to 0 if objects i and j match, and 1 if the objects differ.
- Thus, we get
$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$
- All objects are dissimilar except objects 1 and 4 (i.e., $d(4,1) = 0$).

Proximity Measures for Nominal Attributes (Method 2)

- Proximity between objects described by nominal attributes can be computed using an alternative encoding scheme.
- Nominal attributes can be encoded using asymmetric binary attributes by creating a new binary attribute for each of the M states.
 - For an object with a given state value, the binary attribute representing that state is set to 1, while the remaining binary attributes are set to 0.

Proximity Measure for Nominal Attributes (Recap.)

- 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

$$d(i, j) = \frac{p - m}{p} \quad (2.11)$$

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

Proximity Measures for Binary Attributes

- Compute a dissimilarity matrix from the given binary data.
- If all binary attributes are thought of as having the same weight, we have the 2 x 2 contingency table below,

		Object j		sum
		1	0	
Object i	1	q	r	$q + r$
	0	s	t	$s + t$
	sum	$q + s$	$r + t$	p

- where q is the number of attributes that equal 1 for both objects i and j ,
- r is the number of attributes that equal 1 for object i but equal 0 for object j ,
- s is the number of attributes that equal 0 for object i but equal 1 for object j ,
- t is the number of attributes that equal 0 for both objects i and j .
- The total number of attributes is p , where $p = q + r + s + t$.

Proximity Measures for Binary Attributes

- The 2 x 2 contingency table

		Object j		sum
		1	0	
Object i	1	q	r	$q + r$
	0	s	t	$s + t$
	sum	$q + s$	$r + t$	p

- Dissimilarity that is based on symmetric binary attributes is called **symmetric binary dissimilarity**.
- If objects i and j are described by symmetric binary attributes, then the **dissimilarity between i and j** is

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

Proximity Measures for Binary Attributes

- The 2 x 2 contingency table

		Object <i>j</i>		sum
		1	0	
Object <i>i</i>	1	<i>q</i>	<i>r</i>	<i>q + r</i>
	0	<i>s</i>	<i>t</i>	<i>s + t</i>
sum		<i>q + s</i>	<i>r + t</i>	<i>p</i>

- For asymmetric binary attributes, the two states are not equally important, such as the positive (1) and negative (0) outcomes of a disease test.
- Given two asymmetric binary attributes, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match).
- Such binary attributes are often considered “monary” (having one state).
- The dissimilarity based on these attributes is called asymmetric binary dissimilarity, where the number of negative matches, *t*, is considered unimportant and is thus ignored in the following computation:

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

Proximity Measures for Binary Attributes

- The 2 x 2 contingency table

		Object <i>j</i>		
		1	0	sum
Object <i>i</i>	1	<i>q</i>	<i>r</i>	<i>q + r</i>
	0	<i>s</i>	<i>t</i>	<i>s + t</i>
	sum	<i>q + s</i>	<i>r + t</i>	<i>p</i>

- Complementarily, we can measure the difference between two binary attributes based on the notion of similarity instead of dissimilarity.
- For example, the asymmetric binary similarity between the objects *i* and *j* can be computed as

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

- It is called the Jaccard coefficient and is popularly referenced in the literature.

Proximity Measure for Binary Attributes (Summary)

- A contingency table for binary data

		Object j		
		1	0	sum
Object i	1	q	r	$q + r$
	0	s	t	$s + t$
	sum	$q + s$	$r + t$	p

- Distance measure for symmetric binary variables:

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

- Distance measure for asymmetric binary variables:

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

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

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

- 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

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

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

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

- Jim and Mary are unlikely to have a similar disease because they have the highest dissimilarity value among the three pairs.
- Jack and Mary are the most likely to have a similar disease.

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

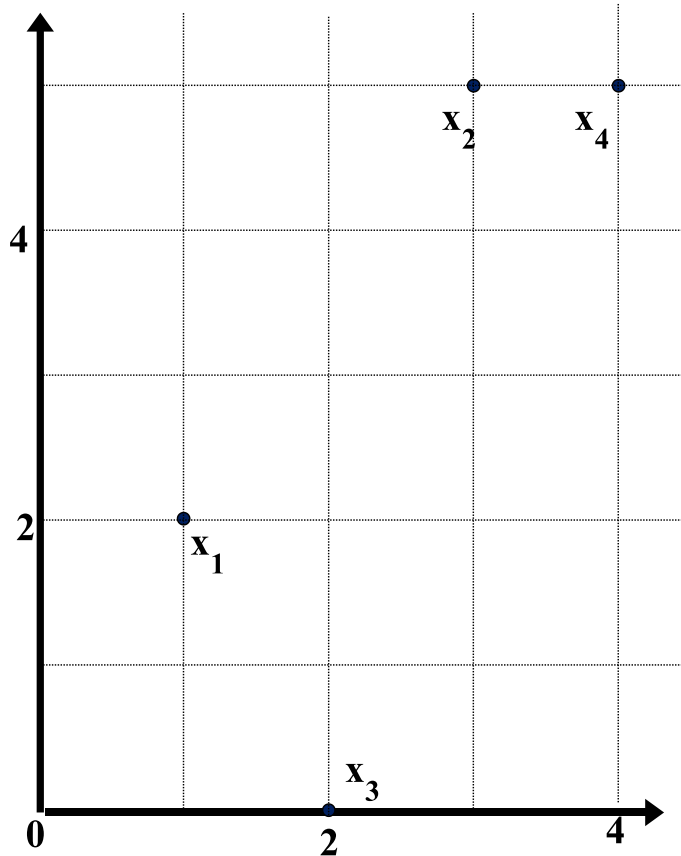
$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

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

- standardized measure (*z-score*): $z_{if} = \frac{x_{if} - m_f}{s_f}$
- Using mean absolute deviation is more robust than using standard deviation

Example:

Data Matrix and Dissimilarity Matrix



Data Matrix

point	attribute1	attribute2
$x1$	1	2
$x2$	3	5
$x3$	2	0
$x4$	4	5

Dissimilarity Matrix
(with **Euclidean Distance**)

	$x1$	$x2$	$x3$	$x4$
$x1$	0			
$x2$	3.61	0		
$x3$	5.1	5.1	0	
$x4$	4.24	1	5.39	0

Distance on Numeric Data: Minkowski Distance

- *Minkowski distance*: A popular distance measure

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

where $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ are two p -dimensional data objects, and h is the order (the distance so defined is also called L- h norm)

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

Special Cases of Minkowski Distance

- $h = 1$: **Manhattan** (city block, L_1 norm) **distance**
 - E.g., the Hamming distance: the number of bits that are different between two binary vectors

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

- $h = 2$: (L_2 norm) **Euclidean** distance

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

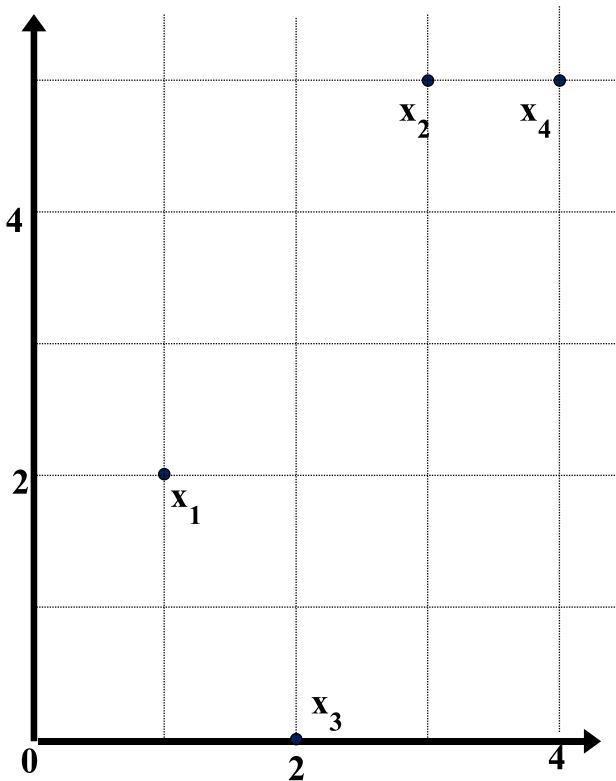
- $h \rightarrow \infty$. **“supremum”** (L_{\max} norm, L_{∞} norm) distance.
 - This is the maximum difference between any component (attribute) of the vectors

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

Example: Minkowski Distance

Dissimilarity Matrices

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean (L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

L_∞	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

Proximity Measures for Ordinal Attributes

- The values have a meaningful order or ranking about them, yet the magnitude between successive values is unknown.
 - An example includes the sequence small, medium, large for a size attribute.
- Ordinal attributes may also be obtained from the discretization of numeric attributes.
- These categories are organized into ranks.
- The range of a numeric attribute can be mapped to an ordinal attribute f having M_f states.
 - e.g., the range of the interval-scaled attribute temperature (in Celsius) can be organized into the following states
 - -30 to -10 (cold temperature)
 - -10 to 10 (moderate temperature)
 - 10 to 30 (warm temperature)

Proximity Measures for Ordinal Attributes

- Let M be the number of possible states that an ordinal attribute can have.
 - Ordered states define the ranking $1, \dots, M_f$.
- Let f be an attribute from a set of ordinal attributes describing n objects.
- The dissimilarity computation with respect to f involves the following steps:
 1. The value of f for the i -th object is x_{if} , and f has M_f ordered states, representing the ranking $1, \dots, M_f$. Replace each x_{if} by its corresponding rank, r_{if} in $\{1, \dots, M_f\}$.
 2. Since each ordinal attribute can have a different number of states, it is often necessary to map the range of each attribute onto $[0.0, 1.0]$ so that each attribute has equal weight. We perform such data normalization by replacing the rank r_{if} of the i th object in the f -th attribute by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$
 3. Dissimilarity can then be computed using any of the distance measures for numeric attributes, using z_{if} to represent the f value for the i -th object.

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
 - replace x_{if} by their rank $r_{if} \in \{1, \dots, M_f\}$
 - map the range of each variable onto $[0, 1]$ by replacing i -th object in the f -th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- compute the dissimilarity using methods for interval-scaled variables

Dissimilarity for Attributes of Mixed Types

- In many real databases, objects are described by a **mixture** of attribute types.
- In general, a database can contain all of these attribute types.
- *"So, how can we compute the dissimilarity between objects of mixed attribute types?"*
- One approach is to group each type of attribute together, performing separate data mining (e.g., clustering) analysis for each type.
 - This is feasible if these analyses derive compatible results.
 - However, in real applications, it is unlikely that a separate analysis per attribute type will generate compatible results.
- A more preferable approach is to process all attribute types together, performing a single analysis.
 - One such technique combines the different attributes into a single dissimilarity matrix, bringing all of the meaningful attributes onto a common scale of the interval $[0.0, 1.0]$.

Attributes of Mixed Type

- Suppose the data set contains p attributes of mixed type.
 - Nominal, symmetric binary, asymmetric binary, numeric, ordinal
- The dissimilarity $d(i, j)$ between objects i and j is defined as

$$d(i, j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p \delta_{ij}^{(f)}}$$

- f is binary or nominal:
 - $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise
- f is numeric: use the normalized distance

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$$

- f is ordinal
 - Compute ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$
 - Treat z_{if} as numeric

Cosine Similarity

- A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as a keyword) or phrase in the document.
- Thus, each document is an object represented by what is called a term-frequency vector.
- For example, we see that Document1 contains five instances of the word team, while hockey occurs three times.
- The word coach is absent from the entire document, as indicated by a count value of 0. Such data can be highly asymmetric.

Document Vector or Term-Frequency Vector

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

Cosine Similarity

- Term-frequency vectors are typically very long and sparse (i.e., they have many 0 values).
- Applications using such structures include information retrieval, text document clustering, biological taxonomy, and gene feature mapping.
- The traditional distance measures that we have studied in this chapter do not work well for such sparse numeric data.
- For example, two term-frequency vectors may have many 0 values in common, meaning that the corresponding documents do not share many words, but this does not make them similar.
- We need a measure that will focus on the words that the two documents do have in common, and the occurrence frequency of such words.
- In other words, we need a measure for numeric data that ignores zero-matches.

Cosine Similarity

- Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words.
- Let x and y be two vectors for comparison.
- Using the cosine measure as a similarity function, we have
 - $\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| ||d_2||$
 - where \bullet indicates vector dot product and $||x||$ is the Euclidean norm of vector $x = (x_1, x_2, \dots, x_p)$, defined as $\sqrt{(x_1^2 + x_2^2 + \dots + x_p^2)}$
- It computes the cosine of the angle between vectors x and y .
- A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match.
- The closer the cosine value to 1, the smaller the angle and the greater the match between vectors.
- Since the cosine similarity measure does not obey all of the properties of metric measures, it is referred to as a nonmetric measure.

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 d_1 and d_2 are two vectors (e.g., term-frequency vectors), then
$$\cos(d_1, d_2) = (d_1 \bullet d_2) / (||d_1|| ||d_2||),$$
where \bullet indicates vector dot product, $||d||$: the length of vector d

Example: Cosine Similarity

- $\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| ||d_2||$,
where \bullet indicates vector dot product, $||d||$: the length of vector d

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

$$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$$

$$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$$

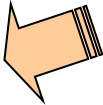
$$d_1 \bullet d_2 = 5*3 + 0*0 + 3*2 + 0*0 + 2*1 + 0*1 + 0*1 + 2*1 + 0*0 + 0*1 = 25$$

$$\begin{aligned} ||d_1|| &= (5*5 + 0*0 + 3*3 + 0*0 + 2*2 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0)^{1/2} \\ &= (42)^{1/2} = 6.481 \end{aligned}$$

$$\begin{aligned} ||d_2|| &= (3*3 + 0*0 + 2*2 + 0*0 + 1*1 + 1*1 + 0*0 + 1*1 + 0*0 + 1*1)^{1/2} \\ &= (17)^{1/2} = 4.12 \end{aligned}$$

$$\cos(d_1, d_2) = 0.94$$

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 

Summary

- Data attribute types: nominal, binary, ordinal, interval-scaled, ratio-scaled
- Many types of data sets, e.g., numerical, text, graph, Web, image.
- Gain insight into the data by:
 - Basic statistical data description: central tendency, dispersion, graphical displays
 - Data visualization: map data onto graphical primitives
 - Measure data similarity
- Above steps are the beginning of data preprocessing.
- Many methods have been developed but still an active area of research.

References

- W. Cleveland, Visualizing Data, Hobart Press, 1993
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- U. Fayyad, G. Grinstein, and A. Wierse. Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann, 2001
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- H. V. Jagadish, et al., Special Issue on Data Reduction Techniques. Bulletin of the Tech. Committee on Data Eng., 20(4), Dec. 1997
- D. A. Keim. Information visualization and visual data mining, IEEE trans. on Visualization and Computer Graphics, 8(1), 2002
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- S. Santini and R. Jain, "Similarity measures", IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(9), 1999
- E. R. Tufte. The Visual Display of Quantitative Information, 2nd ed., Graphics Press, 2001
- C. Yu , et al., Visual data mining of multimedia data for social and behavioral studies, Information Visualization, 8(1), 2009