Data Mining:

Concepts and Techniques

— Chapter 2 —

Jiawei Han, Micheline Kamber, and Jian Pei

University of Illinois at Urbana-Champaign

Simon Fraser University

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
- 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:

	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

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

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)
  - <u>Symmetric binary</u>: both outcomes equally important
    - e.g., gender
  - <u>Asymmetric binary</u>: 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*={*A+, A, A-, B+, …*}, 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.** No ratios.
- **Ratio**
  - Inherent **zero-point**
  - We can speak of values as being a multiple (or **ratio**) of another value (10 K˚ is twice as high as 5 K˚)
    - e.g., *temperature in Kelvin, length, counts, monetary quantities*
- We can compute the ***difference*** between values, the ***mean***, ***median***, and ***central tendency*** mode

Discrete vs. Continuous Attributes

- **Discrete Attribute**
  - Has only a *finite* or *countably infinite* set of values
    - E.g., zip codes, id numbers, or the set of words in a collection of documents
  - Sometimes, represented as integer variables
  - 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

Discretization

- Divide the range of a continuous attribute into intervals
- Some classification algorithms only accept categorical attributes
- Reduce data size by discretization
- Prepare for further analysis

Discretization and Concept Hierarchy

- Discretization
  - Reduce the number of values for a given continuous attribute by **dividing the range of the attribute into intervals**
  - Interval labels can then be used to replace actual data values
  - **Supervised** vs. **unsupervised**
  - **Split** (top-down) vs. **merge** (bottom-up)
  - Discretization can be performed **recursively** on an attribute
- Concept hierarchy formation
  - **Recursively** reduce the data by collecting and replacing low level concepts (such as numeric values for age) by higher level concepts (such as *young*, *middle-aged*, or *senior*)

Discretization and Concept Hierarchy Generation for Numeric Data

- Typical methods: All the methods can be applied recursively
  - Binning (covered above)
    - Top-down split, unsupervised,
  - Histogram analysis (covered above)
    - Top-down split, unsupervised
  - Clustering analysis (covered above)
    - Either top-down split or bottom-up merge, unsupervised
  - **Entropy-based discretization**: supervised, top-down split
  - **Interval merging** by χ^2 Analysis: supervised, bottom-up merge
  - **Segmentation** by natural partitioning: top-down split, unsupervised

Information and Uncertainty

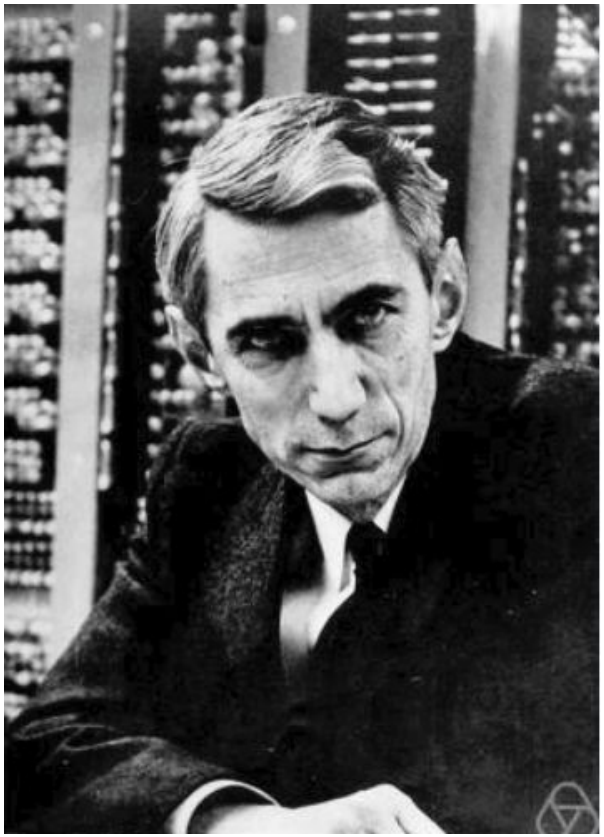
- What is information?
  - Attneave (1959): Information is that which removes or reduces *uncertainty*
- Uncertainty is our limited knowledge about the *outcome* of some (future) event
- Examples of uncertain events
  - Credit card transaction (legitimate / fraudulent)
  - A patient clinical condition (disease = ???)
  - Final grade point average of a student admitted to the university (outcome = value between 0 and 100)
  - More?

How to measure uncertainty?

- A quantitative measure of uncertainty should have at least the following properties
  - If the outcome of an event can be predicted with a 100% accuracy, then the uncertainty of an event is zero
  - The uncertainty of an event increases with the number of possible outcomes (cc vs. student)
  - For the same number of outcomes, the uncertainty is maximal if each outcome has the same probability (examples?)

Who Invented Entropy and Why?

In 1948, Claude Shannon introduced the concept of information entropy in his paper “A Mathematical Theory of Communication”.



Shannon's Information Theory

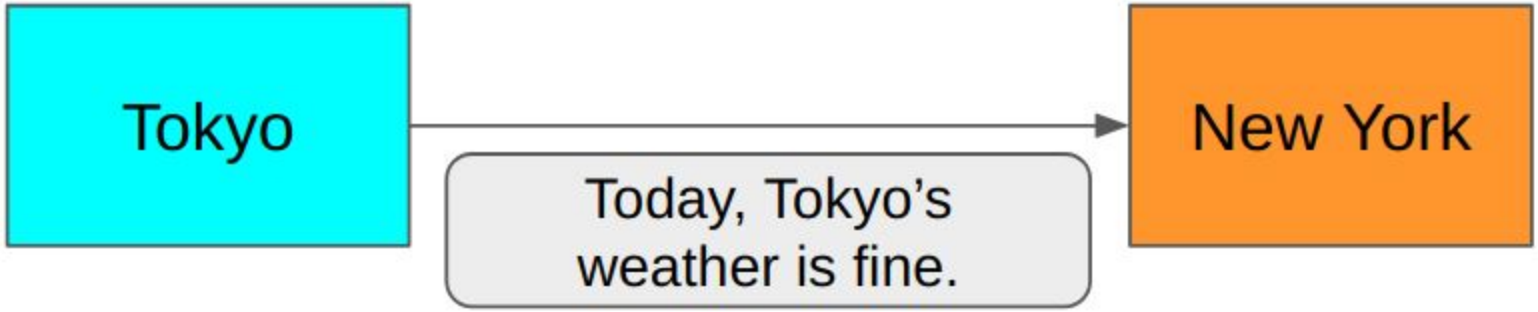
Shannon was looking for a way to efficiently send messages without losing any information.

Shannon measured the efficiency in terms of average message length.

How to encode the original message into the smallest possible data structure without any information loss.

How to Make Efficient and Lossless Encoding?

Is this efficient?

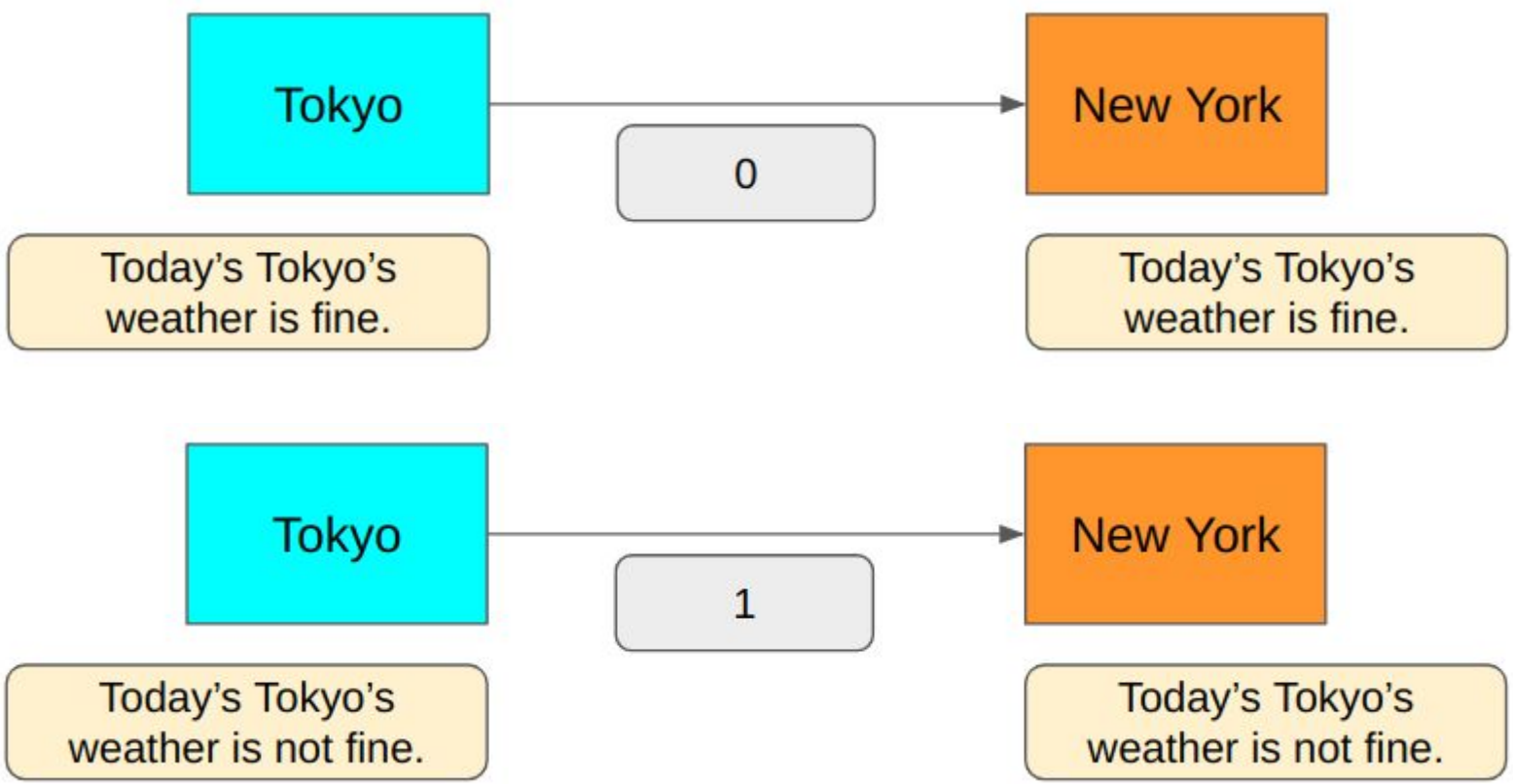


How to Make Efficient and Lossless Encoding?

It is much shorter. Can we do better?

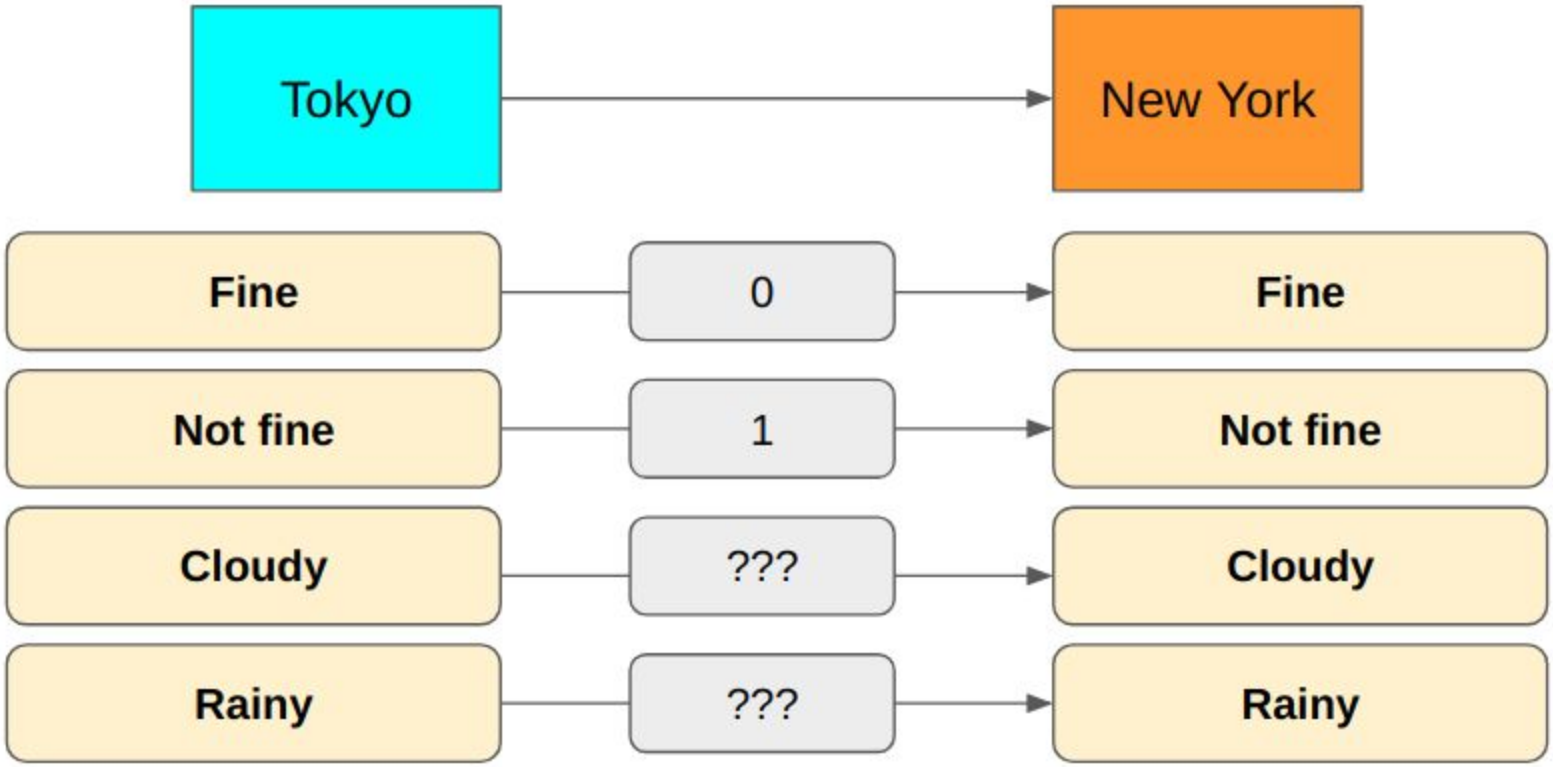
How to Make Efficient and Lossless Encoding?

Is this lossless?



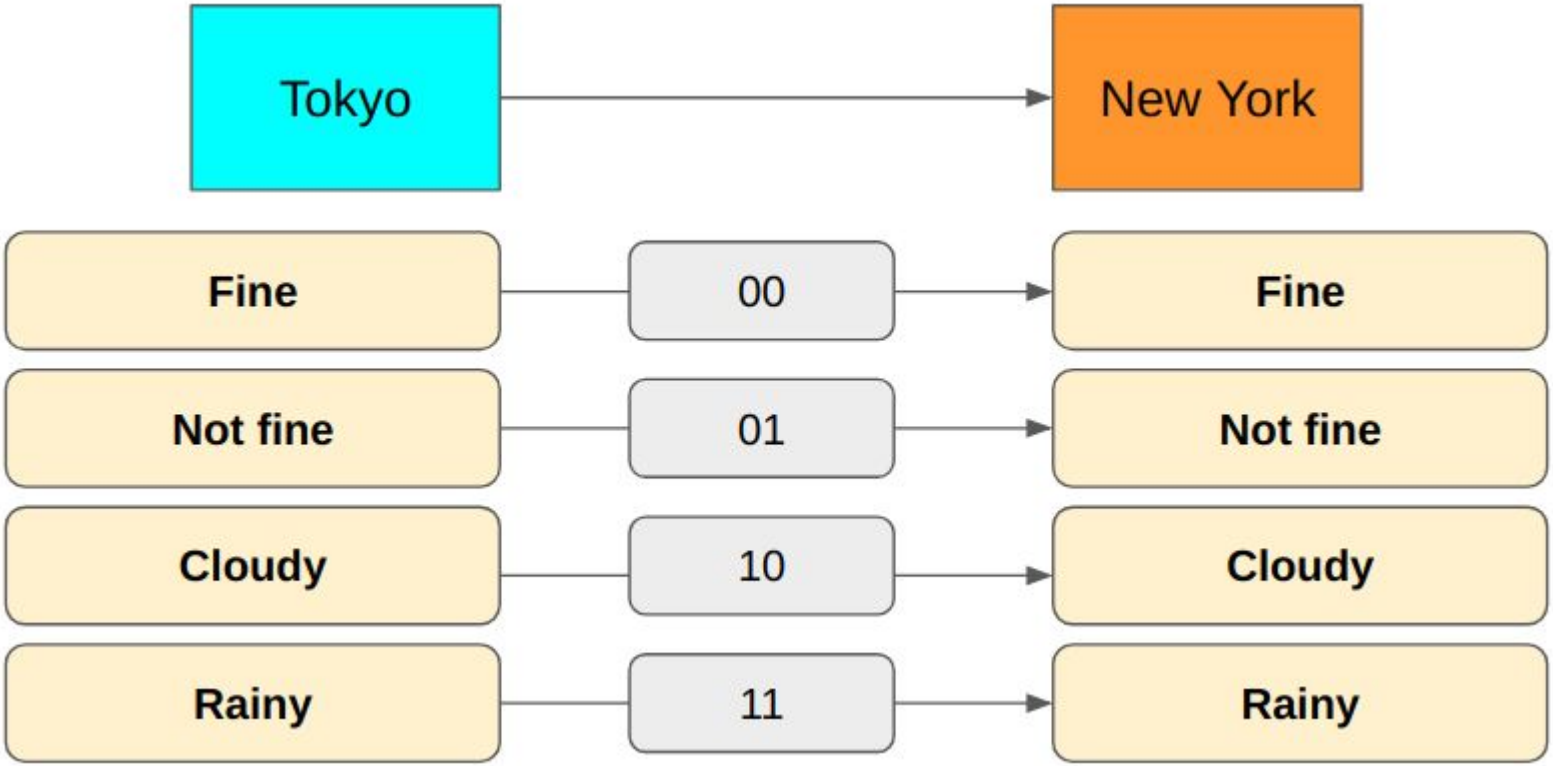
How to Make Efficient and Lossless Encoding?

What if we have more choices ?



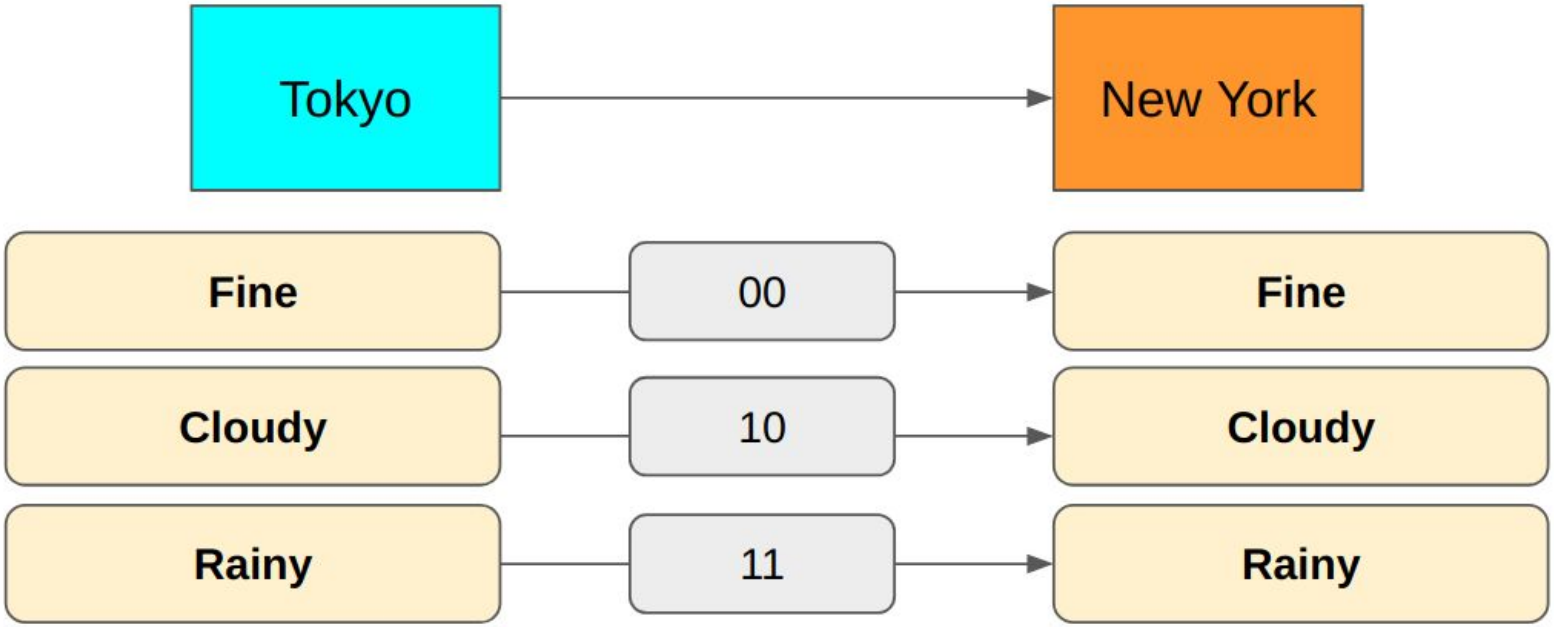
How to Make Efficient and Lossless Encoding?

Using 2 bits



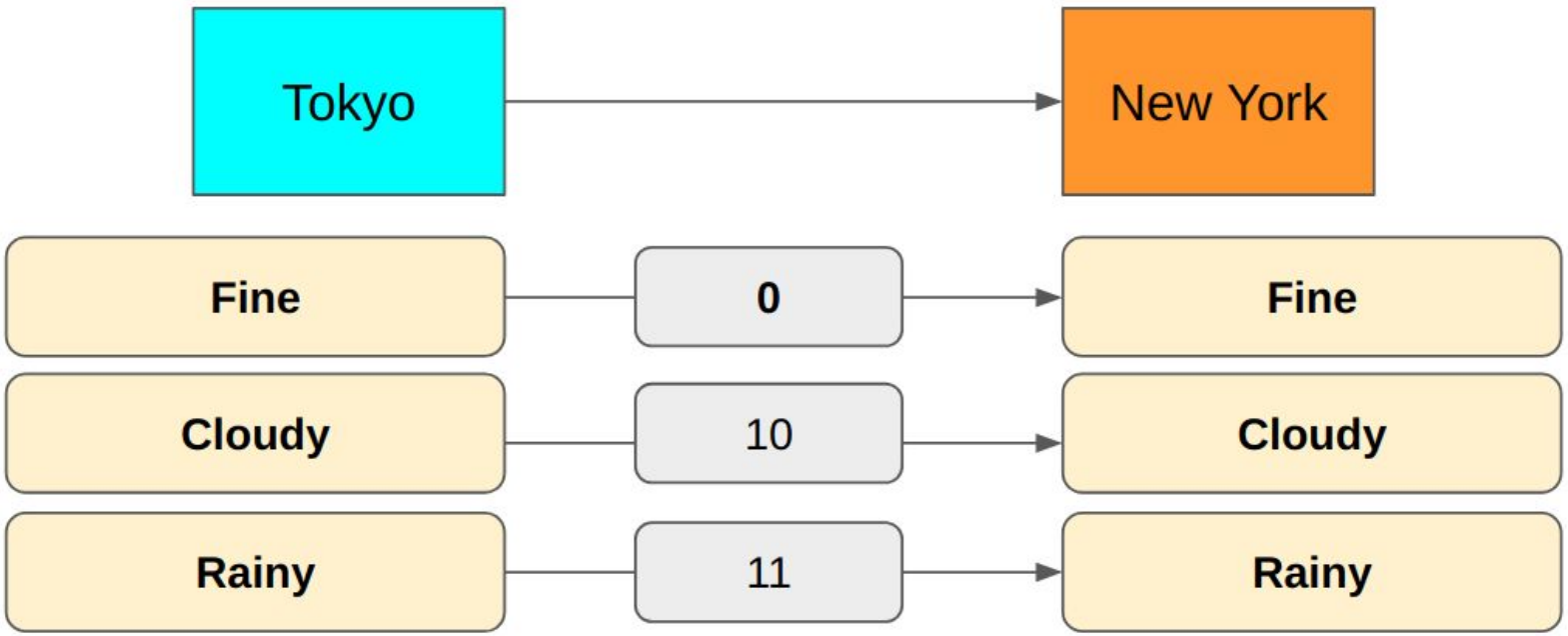
How to Make Efficient and Lossless Encoding?

"Cloudy" and "Rainy" are both "Not fine" so we can omit the explicit message.



How to Make Efficient and Lossless Encoding?

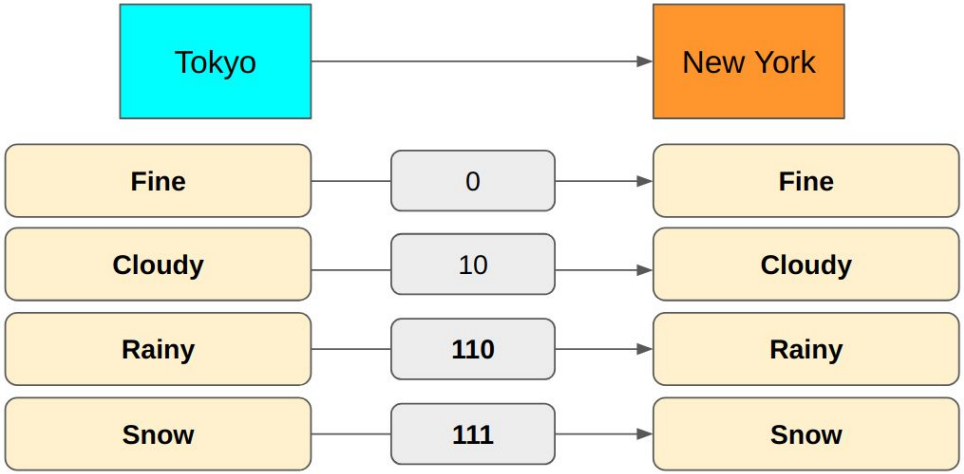
The first bit indicates “Fine or not”. The second bit indicates “Cloudy or Rainy”



How to Make Efficient and Lossless Encoding?

The encoding should prevent ambiguity.

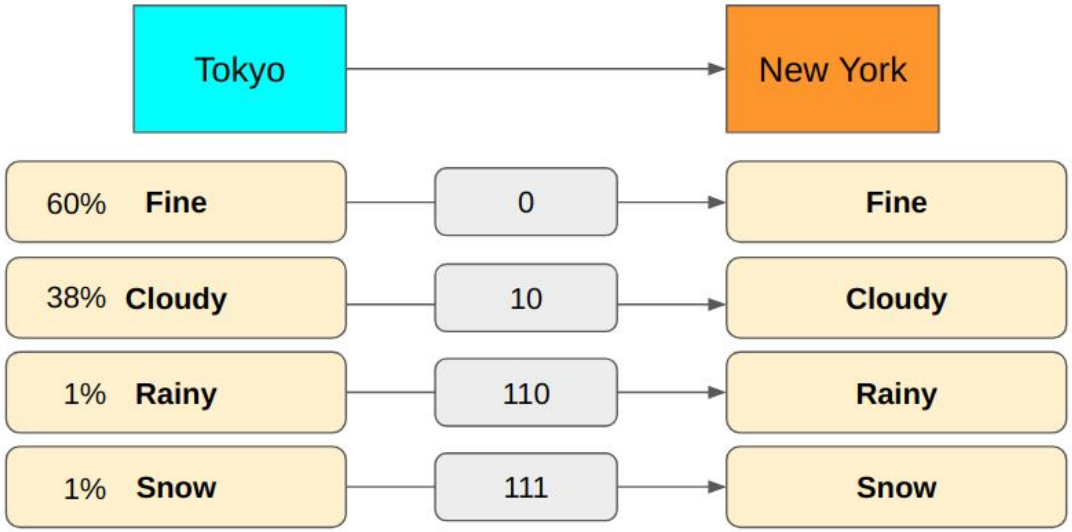
If we use 11 for “Rainy” and 111 for “Snow”, a 5-bit value “11111” could mean either “Rainy, Snow” or “Snow, Rainy”.



So, the encoding now uses the first bit for “Fine or not”, the second bit for “Cloudy or others” and the third bit for “Rain or snow”.

How to Calculate Average Encoding Size

The probability distribution of message types sent from Tokyo to New York.

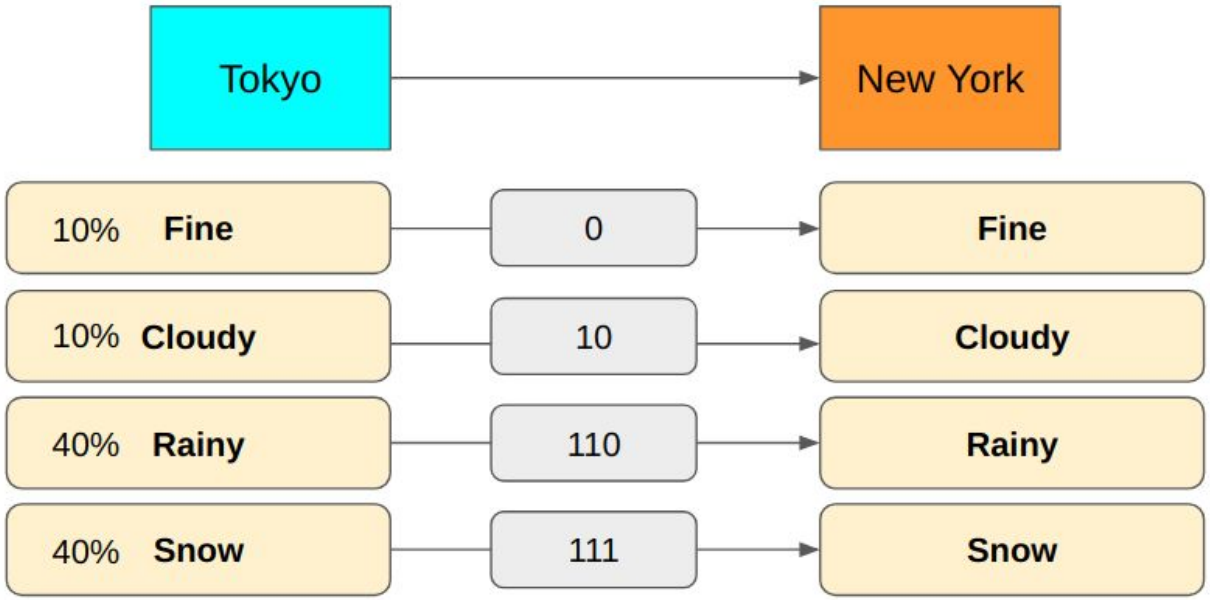


the average number of bits used to send messages from Tokyo to New York.

(0.6 x 1 bit)+(0.38 x 2 bits)+(0.01 x 3 bits)+(0.01 x 3 bits)=1.42 bits

How to Calculate Average Encoding Size

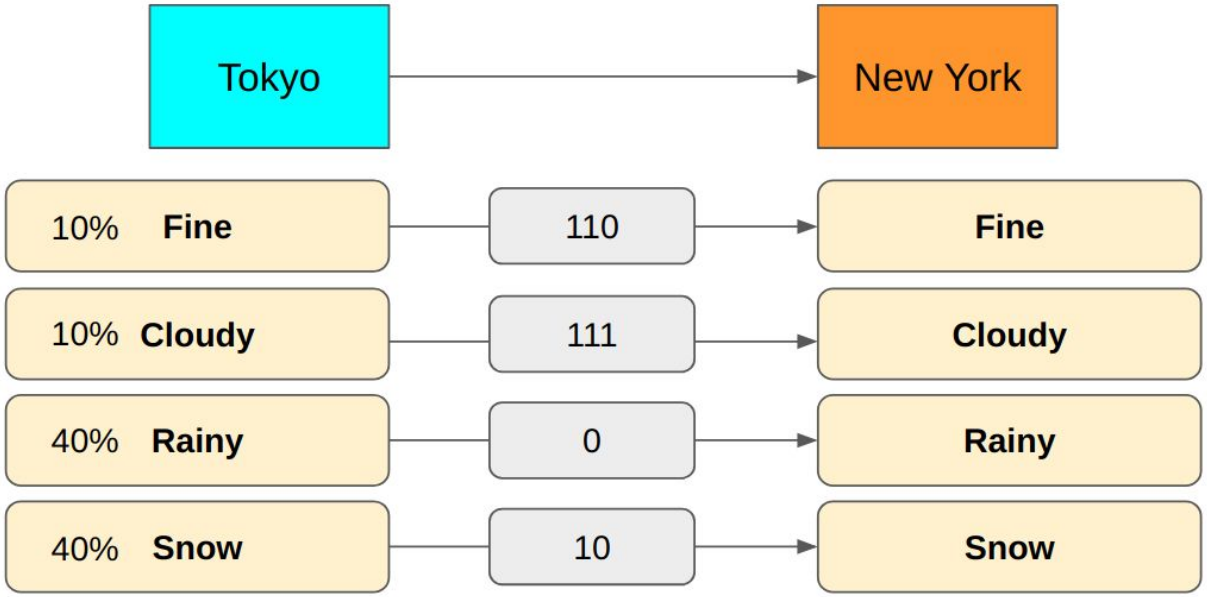
If we have much more rain and snow in Tokyo



(0.1 x 1 bit)+(0.1 x 2 bits)+(0.4 x 3 bits)+(0.4 x 3 bits)=2.7 bits

How to Calculate Average Encoding Size

We could reduce the average encoding size by changing the encoding to use fewer bits for “Rainy” and “Snow”.

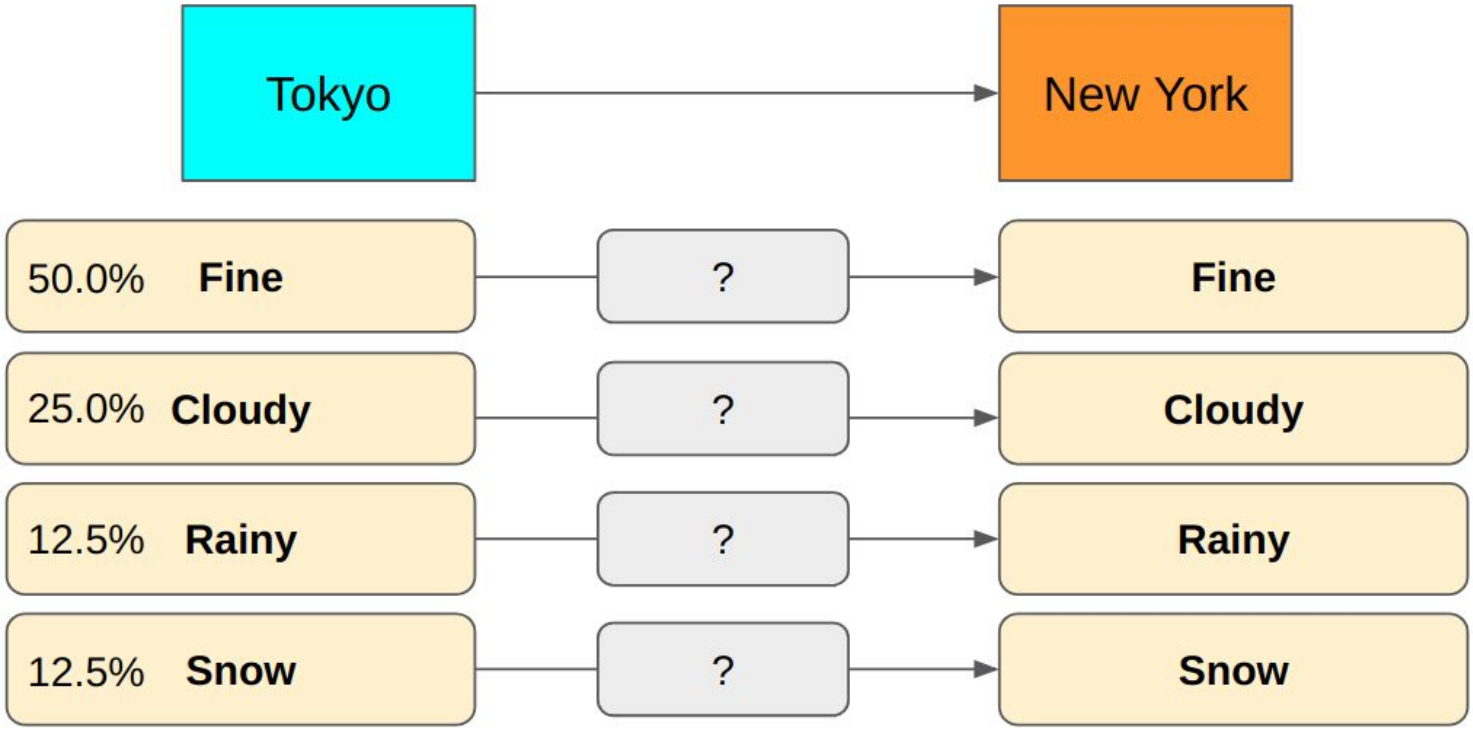


The above encoding requires 1.8 bits on average.

(We are losing the semantic meaning we had though)

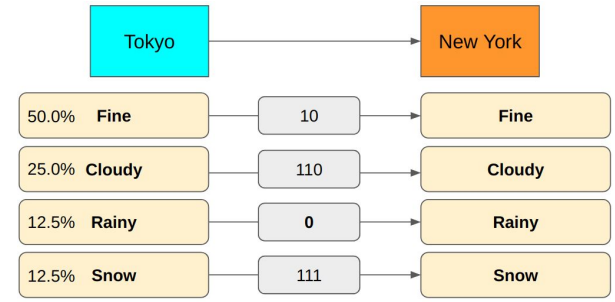
Finding the Smallest Average Encoding Size

How do we calculate the minimum lossless encoding size for each message type?

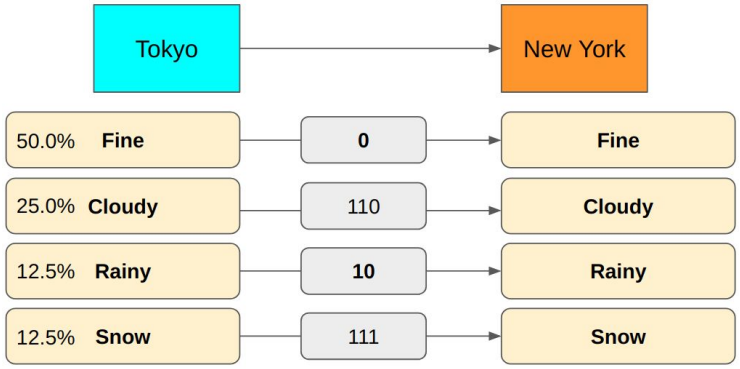


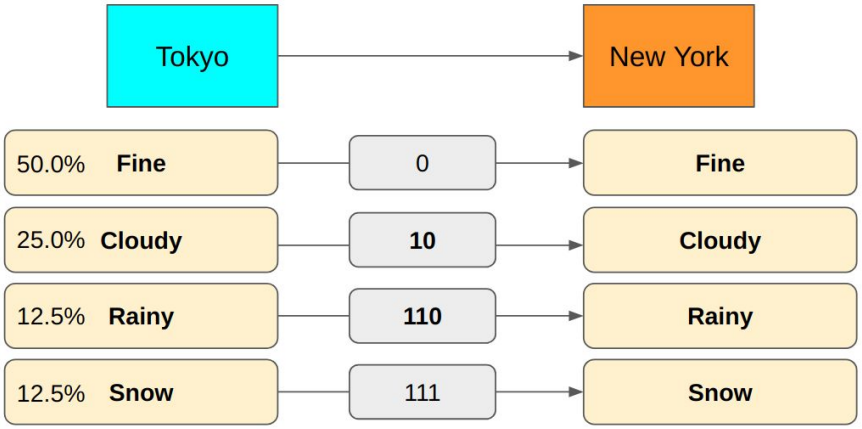
Finding the Smallest Average Encoding Size

We can try trial and error as shown for the “Rainy” message.



Given a probability distribution, is there any easy way to calculate the smallest possible average size of lossless encoding of the messages





How to Calculate Entropy

Suppose we have 8 message types, each of which happens with equal probability (⅛ = 12.5%). What is the minimum number of bits we need to encode them without any ambiguity?

12.5%	A	?
12.5%	B	?
12.5%	C	?
12.5%	D	?
12.5%	E	?
12.5%	F	?
12.5%	G	?
12.5%	H	?

How to Calculate Entropy

How many bits do we need to encode 8 different values?

In general, when we need N different values expressed in bits, we need $\log_2 N$ bits

$$\log_2 8 = \log_2 2^3 = 3 \text{ bits}$$

12.5%	A	000
12.5%	B	001
12.5%	C	010
12.5%	D	011
12.5%	E	100
12.5%	F	101
12.5%	G	110
12.5%	H	111

How to Calculate Entropy

If a message type happens 1 out of N times, the above formula gives the minimum size required.

As P=1/N is the probability of the message, the same thing can be expressed as: $\log_2 N = -\log_2 1/N = -\log_2 P$

How to Calculate Entropy

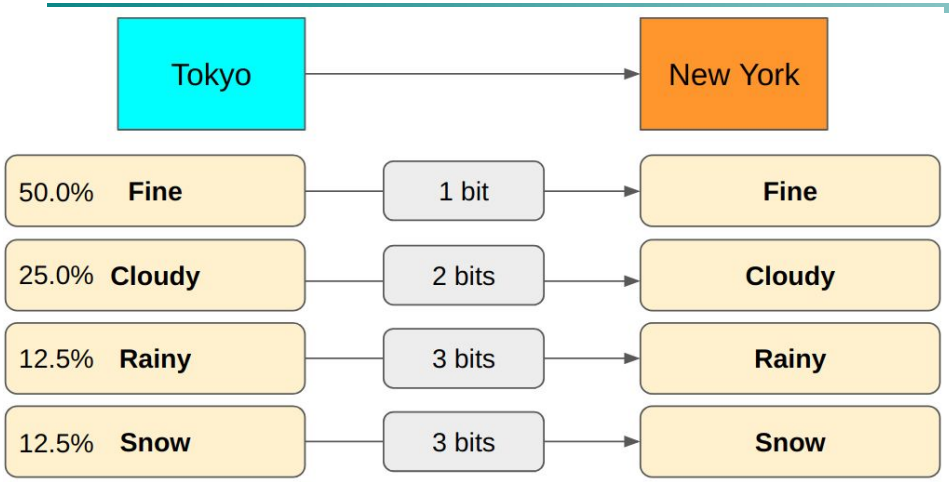
The minimum average encoding size is
∑#bitsForMessage_i*P(i)

$\sum \log_2 N$ *P(i)

$\log_2 N = -\log_2 1/N = -\log_2 P$

$$Entropy = -\sum_i P(i) \log_2 P(i)$$

How to Calculate Entropy - an example



$P(\text{Fine}) = 0.5$

$-\log_2 P(\text{Fine}) = -\log_2 0.5 = -\log_2 1/2 = \log_2 2 = 1 \text{ bit}$

$P(\text{Cloudy}) = 0.25$

$-\log_2 P(\text{Cloudy}) = -\log_2 0.25 = -\log_2 1/4 = \log_2 4 = 2 \text{ bits}$

$P(\text{Rainy}) = 0.125$

$-\log_2 P(\text{Rainy}) = -\log_2 0.125 = -\log_2 1/8 = \log_2 8 = 3 \text{ bits}$

$P(\text{Snow}) = 0.125$

$-\log_2 P(\text{Snow}) = -\log_2 0.125 = -\log_2 1/8 = \log_2 8 = 3 \text{ bits}$

The entropy is:

(0.5 x 1 bit)+(0.25 x 2 bits)+(0.125 x 3 bits)+(0.125 x 3 bits)=1.75 bits

Shannon's Information Theory
Basic Concepts

- **Entropy**
- **Goal**: measure of uncertainty of *X*
- $H(X) = \Sigma - p(x) \log_2 p(x)$

  Where

  X - a discrete random variable

  x - value of *X*

  p (x) - probability of *x*

Properties of Entropy

1. *H (X) = 0* if and only if the outcome is deterministic (all *p(x)* but one are zero): *– 0*log 0 - 1* log 1 = 0*
2. *H(X) ≤ log [number of outcomes]. H (X)* is a maximum, when all outcomes are equiprobable: *max H (X) = log [number of outcomes]*
3. If all the outcomes have the same probability, then *H (X)* is a monotonic increasing function of the number of outcomes

Entropy – Example
Calculating *H (Test)* and *H (Disease)*

Data	Disease = Yes	Disease = No	Total
Test = Negative	1	3	**4**
Test = Positive	4	2	**6**
Total	**5**	**5**	**10**

Max H (Test) = ?
Max H (Disease) = ?

$$H(X) = \Sigma - p(x) \log_2 p(x)$$

H (test)	Test = Negative	Test = Positive	Total
p(test)	0.4	0.6	
-log p(test)	1.322	0.737	
-p(test)*log p(test)	0.529	0.442	**0.971**

Entropy of *Test*

H (disease)	Disease = Yes	Disease = No	Total
P(disease)	0.5	0.5	
-log p(disease)	1.000	1.000	
-p(disease)*log p (disease)	0.500	0.500	**1.000**

Entropy of *Disease*

*

Information Theory

Basic Concepts (cont.)

- **Conditional Entropy**
- **Goal**: measure of uncertainty of *Y*, when *X* is given.
- $H(Y/X) = - \Sigma\, p(x,y) * \log p\,(y/x)$

  Where

  X, Y – discrete random variables

  $p\,(x,y)$ – joint probability of *x* and *y*

  $p\,(y/x)$ – conditional probability of *y* given *x*

  Properties of Conditional Entropy

1. *If* $Y = f\,(X)$ *then* $H(Y/X) = 0$
2. *The uncertainty of Y is never increased by knowledge of X:* $H\,(Y/X) \leq H(Y)$
3. *If X and Y are independent, then* $H(Y/X) = H\,(Y)$

Conditional Entropy – Example 1

Calculating *H (Disease/Test)*

Data	Disease = Yes	Disease = No	Total
Test = Negative	1	3	**4**
Test = Positive	4	2	**6**
Total	**5**	**5**	**10**

$$H(Y/X) = - \Sigma\, p(x,y) * \log p\,(y/x)$$

Max H (Disease/Test) = ?

P(test,disease)	Disease = Yes	Disease = No	Total
Test = Negative	0.10	0.30	**0.4**
Test = Positive	0.40	0.20	**0.6**
Total	**0.5**	**0.5**	**1.00**

Conditional entropy of Disease/Test

P(disease/test)	Disease = Yes	Disease = No	Total
Test = Negative	0.25	0.75	**1.00**
Test = Positive	0.67	0.33	**1.00**

-log p(disease/test)	Disease = Yes	Disease = No
Test = Negative	2.000	0.415
Test = Positive	0.585	1.585

-p (test,disease)*log p(disease/test)	Disease = Yes	Disease = No	Total
Test = Negative	0.200	0.125	**0.325**
Test = Positive	0.234	0.317	**0.551**
Total H(disease/test)	**0.434**	**0.442**	**0.875**

*

Conditional Entropy – Example 2

Calculating *H (Test/Disease)*

Data	Disease = Yes	Disease = No	Total
Test = Negative	1	3	**4**
Test = Positive	4	2	**6**
Total	**5**	**5**	**10**

$$H(Y/X) = - \Sigma\, p(x,y) * \log p\,(y/x)$$

Max H (Test/Disease) = ?

P(test,disease)	Disease = Yes	Disease = No	Total
Test = Negative	0.10	0.30	**0.4**
Test = Positive	0.40	0.20	**0.6**
Total	**0.5**	**0.5**	**1.00**

Conditional entropy of Test/Disease

P(test/disease)	Disease = Yes	Disease = No
Test = Negative	0.20	0.60
Test = Positive	0.80	0.40
Total	**1.00**	**1.00**

-log p(test/disease)	Disease = Yes	Disease = No
Test = Negative	2.322	0.737
Test = Positive	0.322	1.322

-p (test,disease)*log p(test/disease)	Disease = Yes	Disease = No	Total
Test = Negative	0.232	0.221	**0.453**
Test = Positive	0.129	0.264	**0.393**
Total H (test/disease)	**0.361**	**0.485**	**0.846**

*

Information Theory
Basic Concepts (cont.)

- **Mutual Information** (of variables *X* and *Y*)

Goal: the reduction in the uncertainty of *Y* as a result of knowing *X*

$$I(X;Y) = H(Y) - H(Y/X) = \sum_{x,y} p(x,y) \bullet \log \frac{p(y/x)}{p(y)}$$

Properties of Mutual Information (MI)

1. *Symmetry: I (X; Y) = I (Y ; X) = H (X) – H (X/Y)*
2. *Mutual information is a non-negative quantity: I (X; Y) ≥ 0*
3. *Maximum MI: If Y = f (X) then I (X; Y) = H(Y)*
4. *Minimum MI: If X and Y are independent, then I (X; Y) = 0*

Mutual Information – Example

Calculating *I (Test; Disease)*

P(test,disease)	Disease = Yes	Disease = No	Total
Test = Negative	0.10	0.30	**0.4**
Test = Positive	0.40	0.20	**0.6**
Total	**0.5**	**0.5**	**1.00**

$$I(X;Y) = \sum_{x,y} p(x,y) \bullet \log \frac{p(y/x)}{p(y)}$$

Mutual information of Disease and Test

P(disease/test)	Disease = Yes	Disease = No	Total
Test = Negative	0.25	0.75	**1.00**
Test = Positive	0.67	0.33	**1.00**

P(disease/test) / P(disease)	Disease = Yes	Disease = No
test=0	0.50	1.50
test=1	1.33	0.67

p(test,disease) log p(disease/test) / p(disease)	disease=	disease= 1	Total
test=0	-0.100	0.175	**0.075**
test=1	0.166	-0.117	**0.049**
Total I (test;disease)	**0.066**	**0.058**	**0.125**
H (test) - H (test/disease)			**0.125**
H(disease) - H(disease/test)			**0.125**

*

Entropy-Based Discretization

- Given a set of samples S, if S is partitioned into two intervals S_1 and S_2 using boundary T, the **information gain** after partitioning is

$$I(S,T) = \frac{|S_1|}{|S|} Entropy(S_1) + \frac{|S_2|}{|S|} Entropy(S_2)$$

- **Entropy** is calculated based on class distribution of the samples in the set. Given *m* classes, the entropy of S_1 is

$$Entropy(S_1) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

where p_i is the probability of class *i* in S_1

- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization
- The process is recursively applied to partitions obtained until some stopping criterion is met
- Such a boundary may reduce data size and improve classification accuracy

Entropy Gain: The Foundation of Discretization

1. Calculate Entropy for your data.
2. For each potential split in your data...
    - Calculate Entropy in each potential bin
    - Find the net entropy for your split
    - Calculate entropy gain
3. Select the split with the highest entropy gain
4. Recursively (or iteratively in some cases) perform the partition on each split until a termination criteria is met
    - Terminate once you reach a specified number of bins
    - Terminate once entropy gain falls below a certain threshold.

Interval Merge by χ^2 Analysis

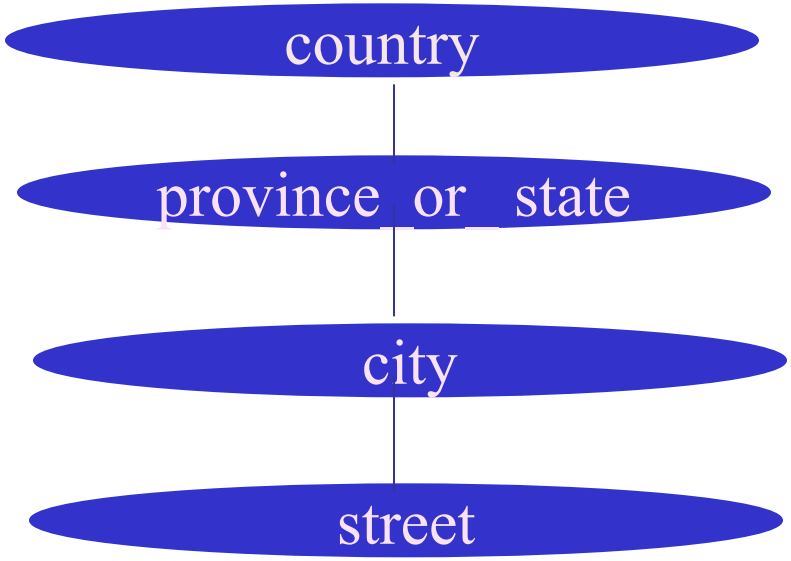
- Merging-based (bottom-up) vs. splitting-based methods
- **Merge**: Find the best neighboring intervals and merge them to form larger intervals recursively
- **ChiMerge** [Kerber AAAI 1992, See also Liu et al. DMKD 2002]
  - Initially, each distinct value of a numerical attr. A is considered to be one interval
  - χ^2 tests are performed for every pair of adjacent intervals
  - Adjacent intervals with the least χ^2 values are merged together, since low χ^2 values for a pair indicate similar class distributions
  - This merge process proceeds recursively until a predefined stopping criterion is met (such as significance level, max-interval, max inconsistency, etc.)

Concept Hierarchy Generation for Categorical Data

- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
  - *street < city < state < country*
  - only *street < city*, not others
- Specification of a hierarchy for a set of values by explicit data grouping
  - {Urbana, Champaign, Chicago} < Illinois
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
  - E.g., for a set of attributes: {street, city, state, country}

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Exceptions, e.g., weekday, month, quarter, year



15 distinct values

365 distinct values

3567 distinct values

674,339 distinct values

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

To better understand the data: *central tendency*, *dispersion*, etc.

- CT measures the location of the middle or center of a data distribution. **Where do most of attribute's values fall?**
  - *mean, median, mode,* and *midrange*
- Data dispersion characteristics. **How are the data spread out?**
  - *Range*, *quartiles*, and *interquartile range*; *variance* and *standard deviation*; median, max, min
  - Useful for identifying **outliers**
- Statistical descriptions and visualization
  - *Charts*, *graphs*, *quantile plots*, *quantile–quantile plots*, *histograms*, *scatter plots*, etc.

Measuring the Central Tendency

- <u>Mean (algebraic measure) (sample vs. population):</u>

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

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

  - Weighted arithmetic mean:

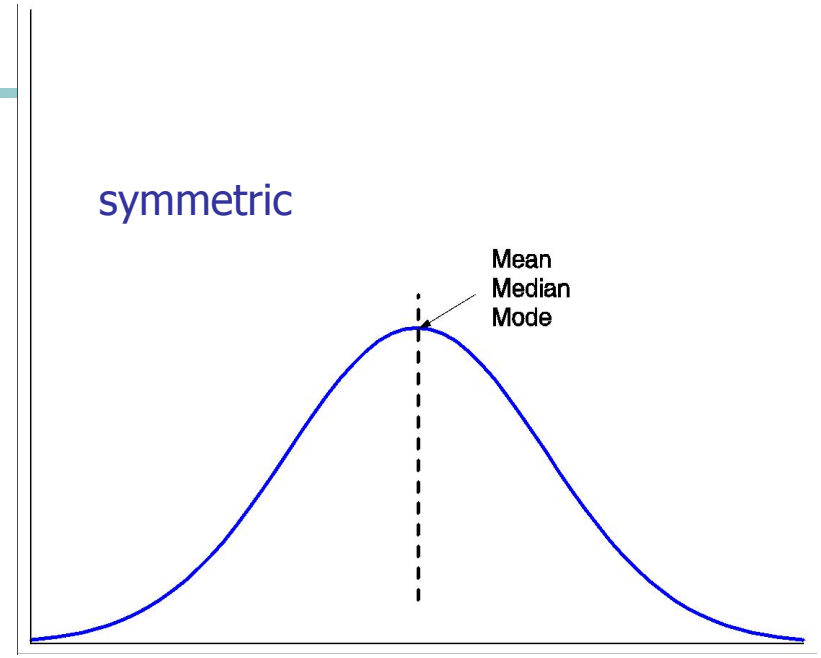
    $$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

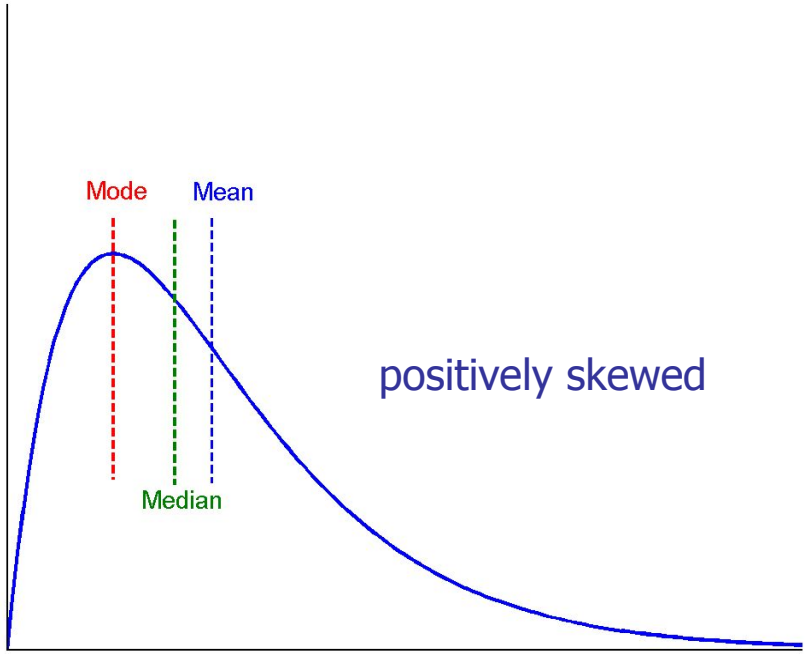
  - Trimmed mean: chopping extreme values
- <u>Median</u>:
  - Middle value if odd number of values, or average of the middle two values otherwise
- <u>Mode</u>
  - Value that occurs most frequently in the data
  - Empirical formula:

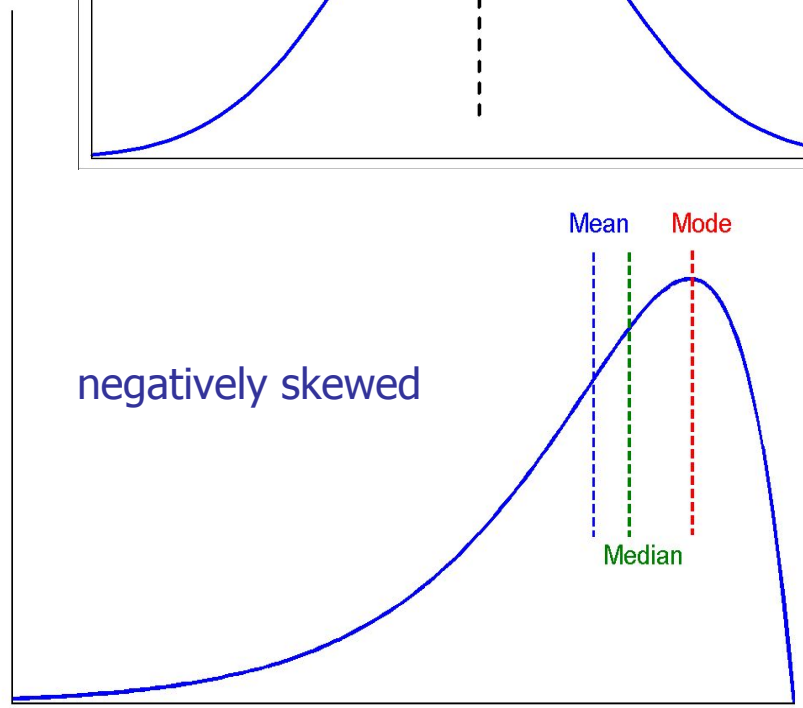
    $$mean - mode = 3 \times (mean - median)$$

Symmetric vs. Skewed Data

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







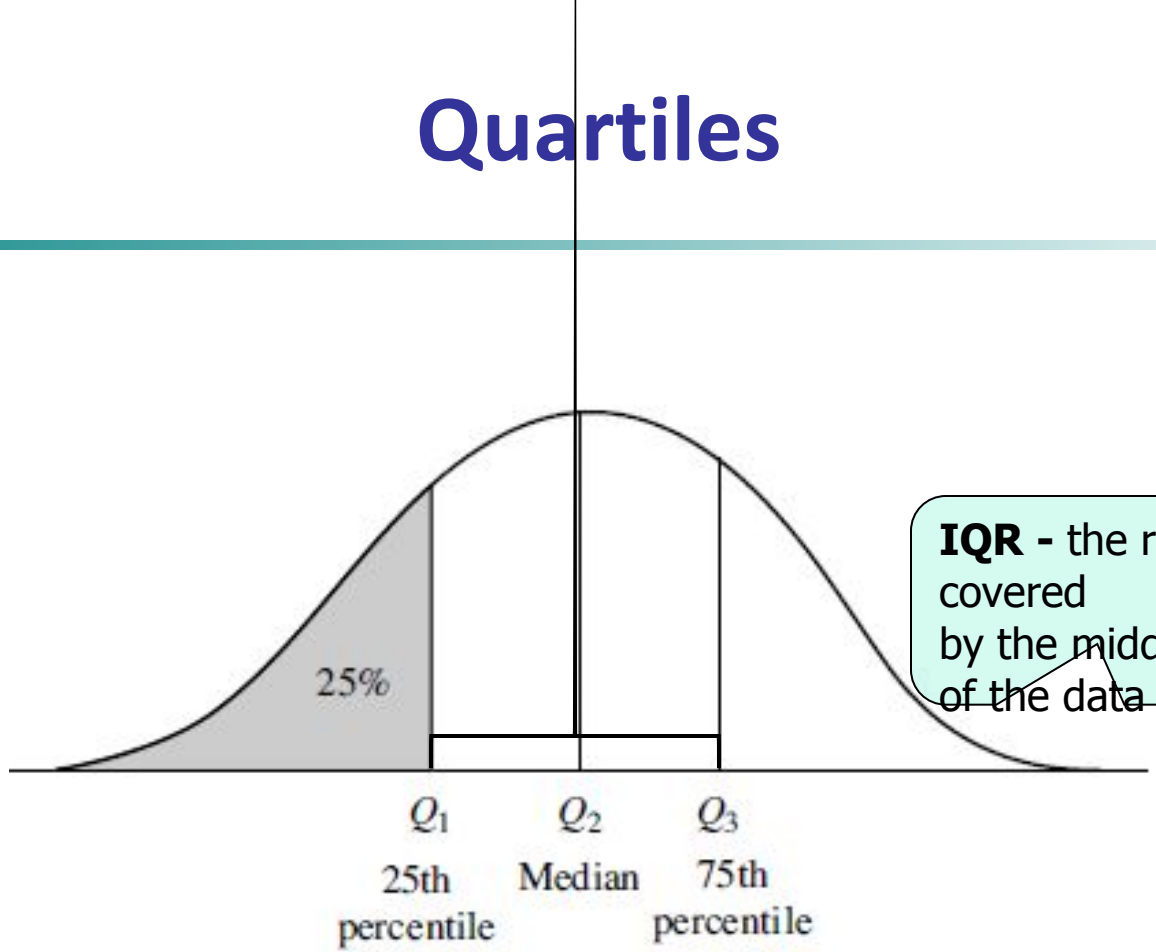
Measuring the Dispersion of Data

- 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)

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

  - **Standard deviation** *s (or σ)* is the square root of variance s^2 *(or* σ^2*)*

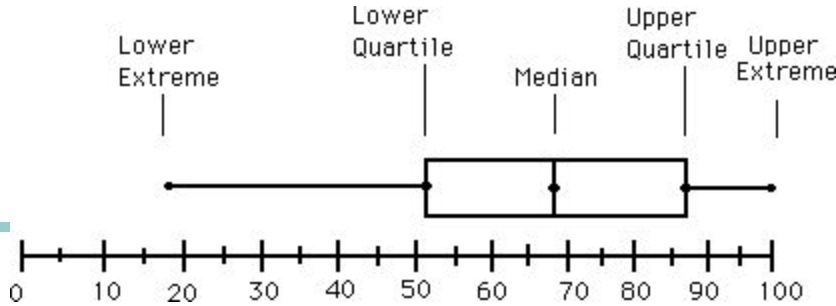
Quartiles



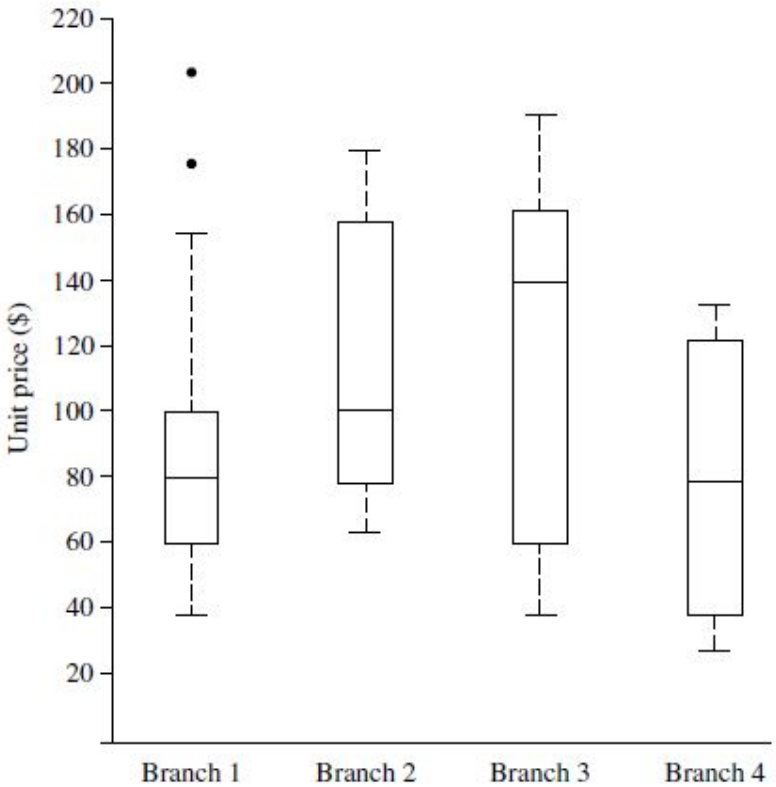
IQR - the range covered
by the middle half
of the data

The three quartiles divide the distribution into four equal-size consecutive subsets

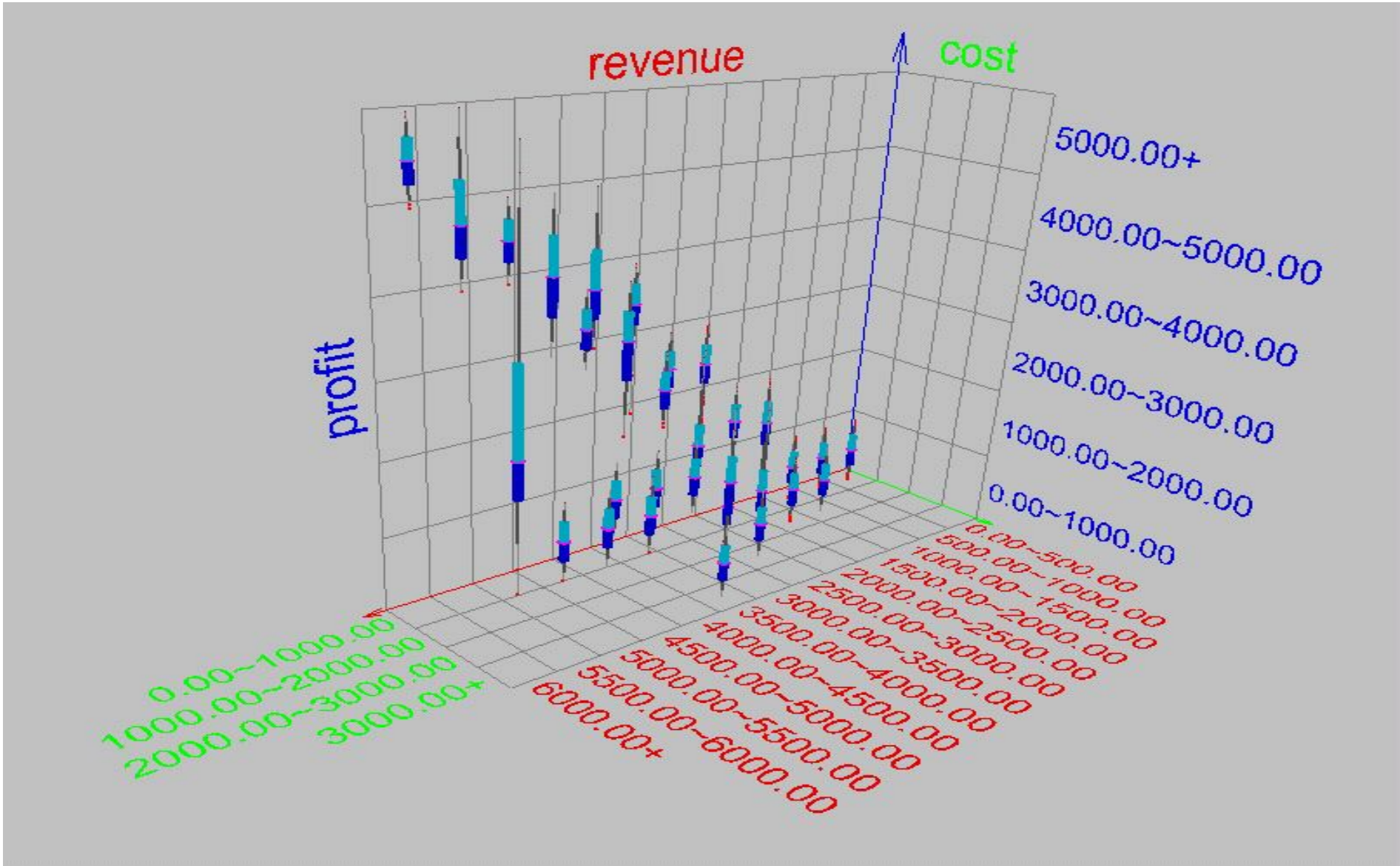
Boxplot Analysis



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



Visualization of Data Dispersion: 3-D Boxplots



Standard Deviation

The basic properties of the standard deviation, $\boldsymbol{\sigma}$, as a measure of spread are:

- σ measures spread about the mean and should be considered only when the **mean** is chosen as the measure of **center**.

- $\boldsymbol{\sigma = 0}$ only when there is **no spread**, that is, when all observations have the same value. Otherwise, $\boldsymbol{\sigma > 0}$.

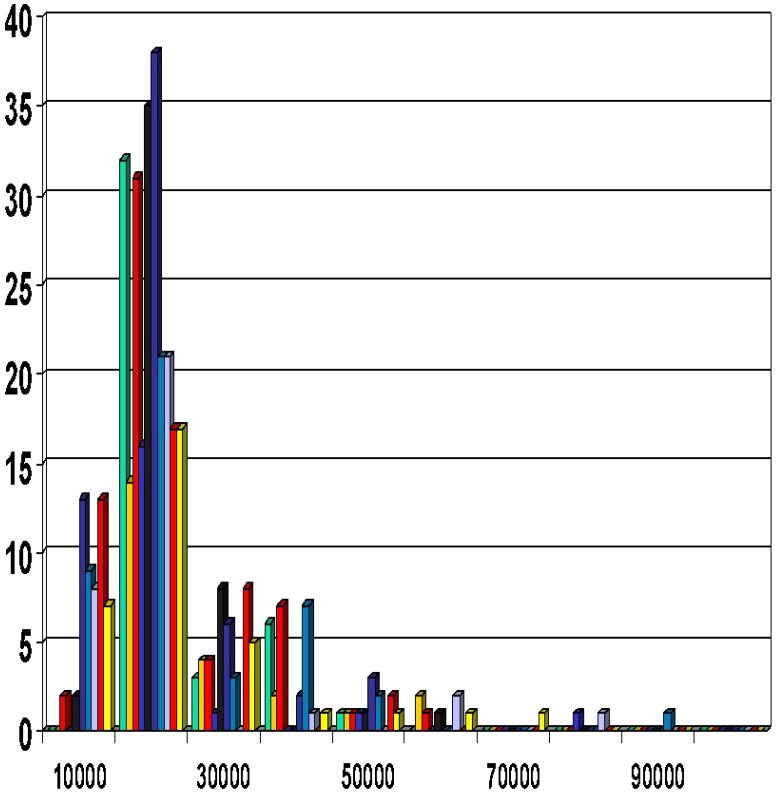
The computation of the variance and standard deviation is scalable in large databases.

Graphic Displays of Basic Statistical Descriptions

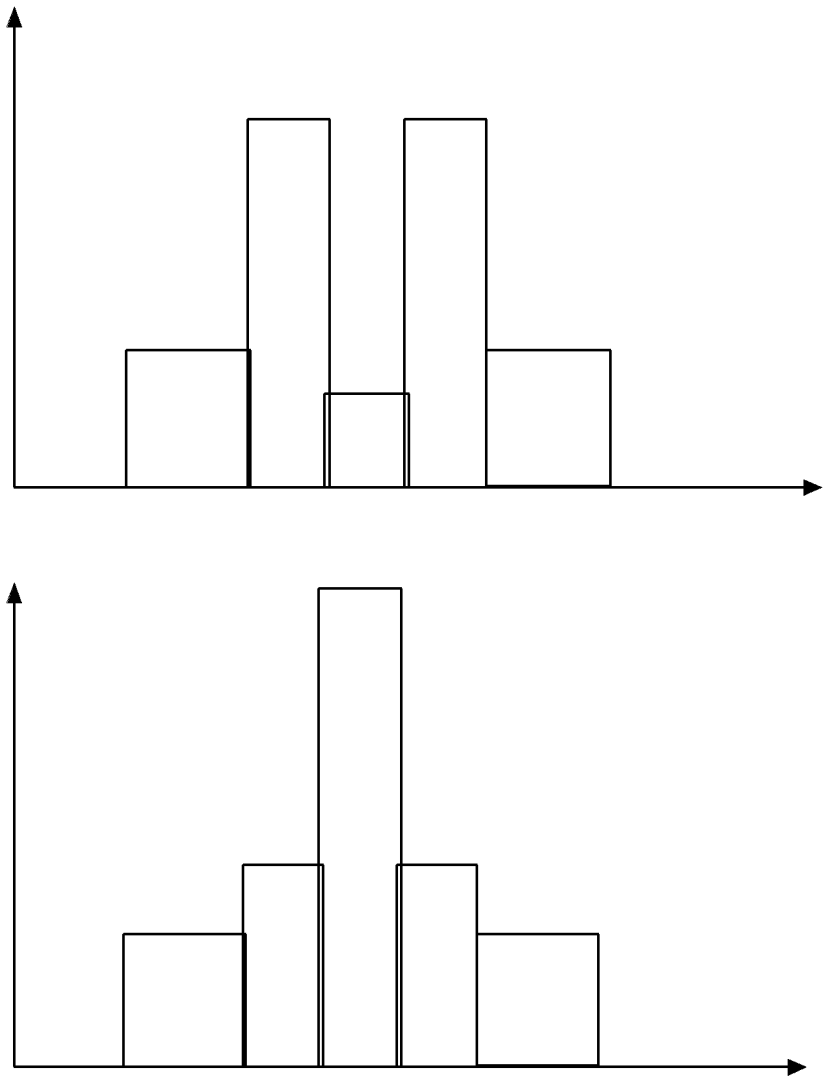
- **Boxplot**: graphic display of five-number summary
- **Histogram**: x-axis are values, y-axis repres. frequencies
- **Quantile plot**: each value x_i is paired with f_i indicating that approximately 100 f_i % of data are $\leq x_i$
- **Quantile-quantile (q-q) plot**: graphs the quantiles of one univariant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

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



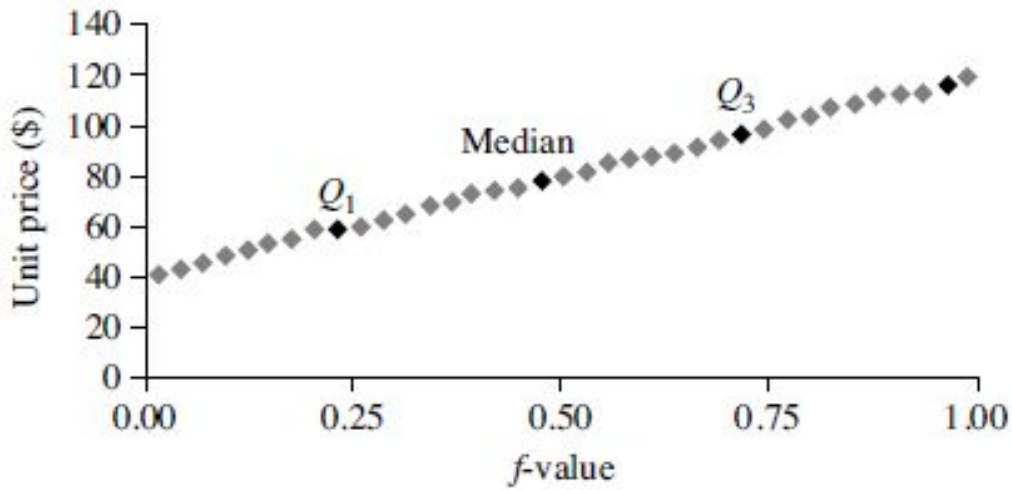
Histograms Often Tell More than Boxplots



- 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

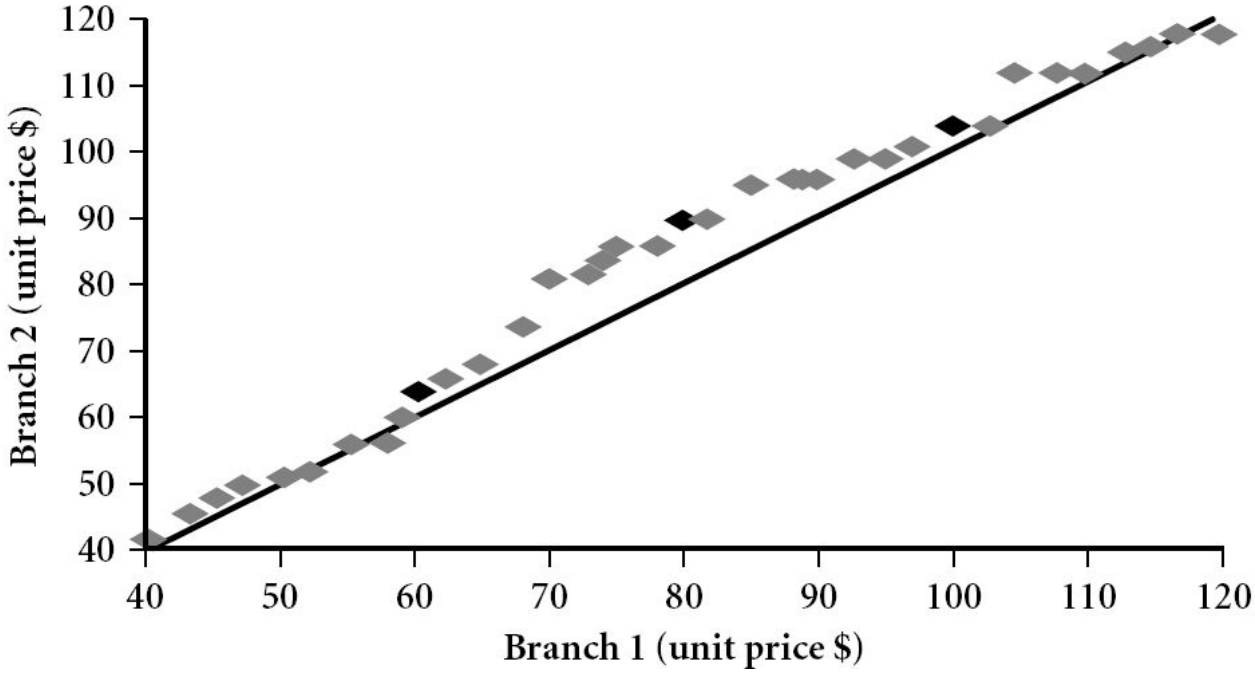
Quantile Plot

- 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 100 f_i% of the data are below or equal to the value x_i
  - $fi = (i - 0.5)/N$.



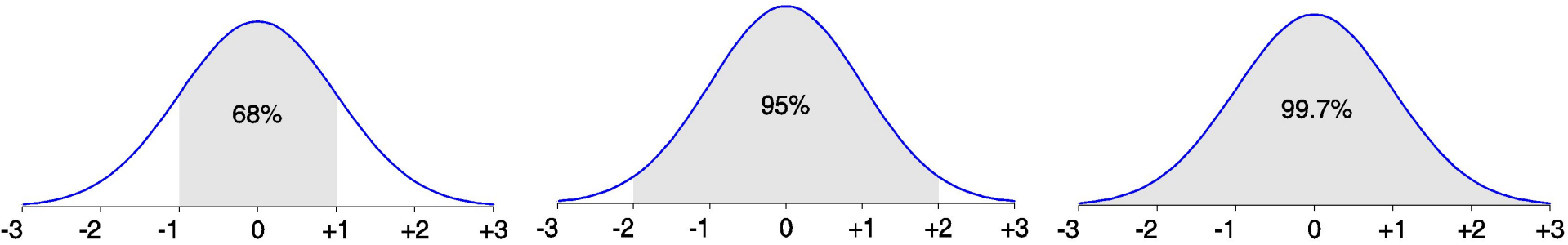
Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- View: Is there 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.

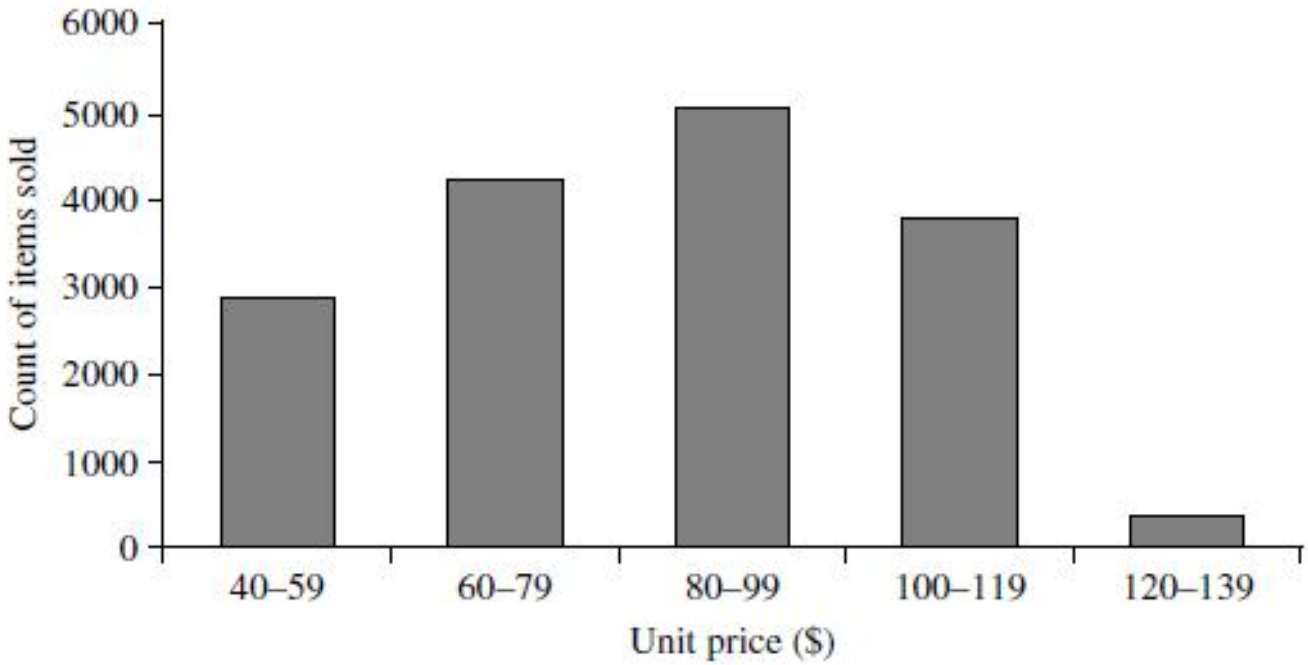


Properties of Normal Distribution Curve

- 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

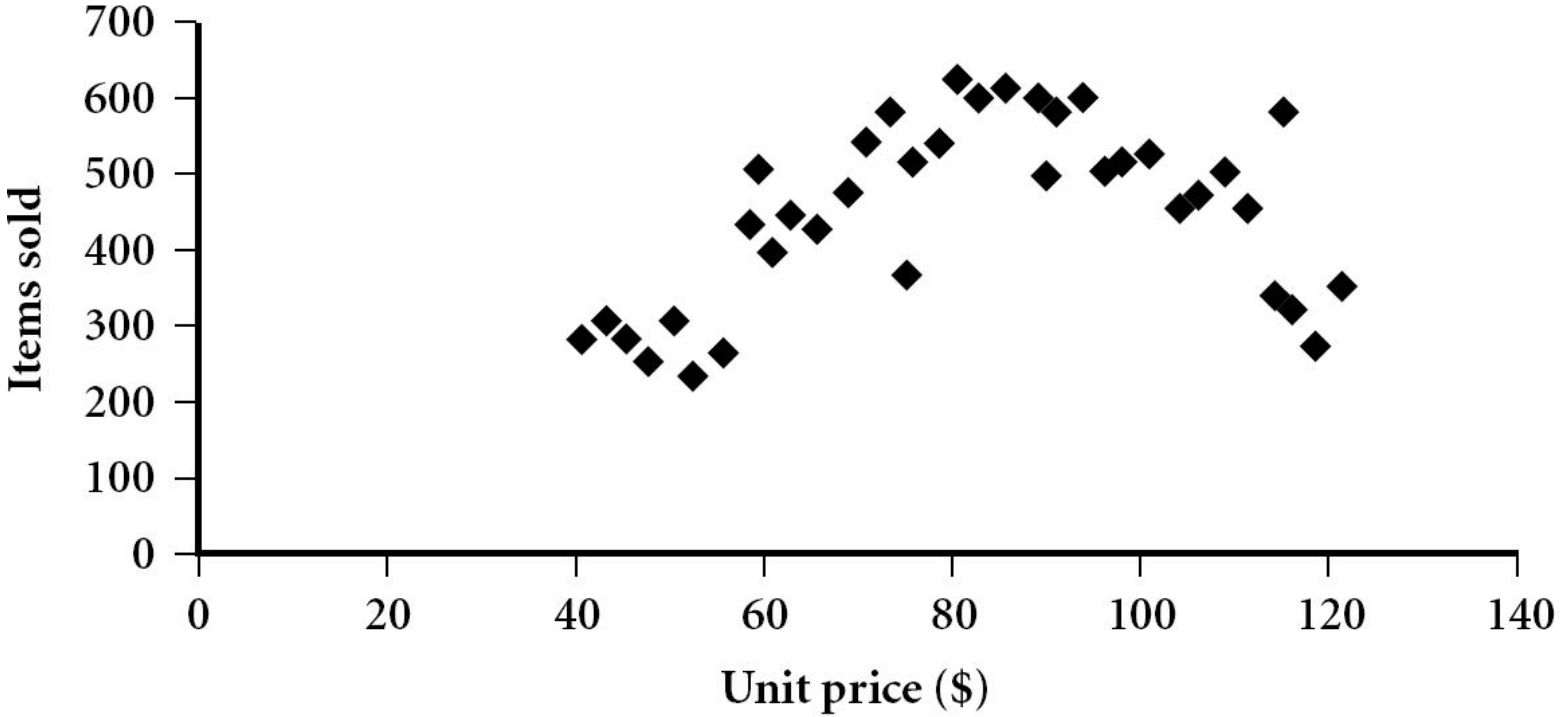


Histogram vs. Scatter plot

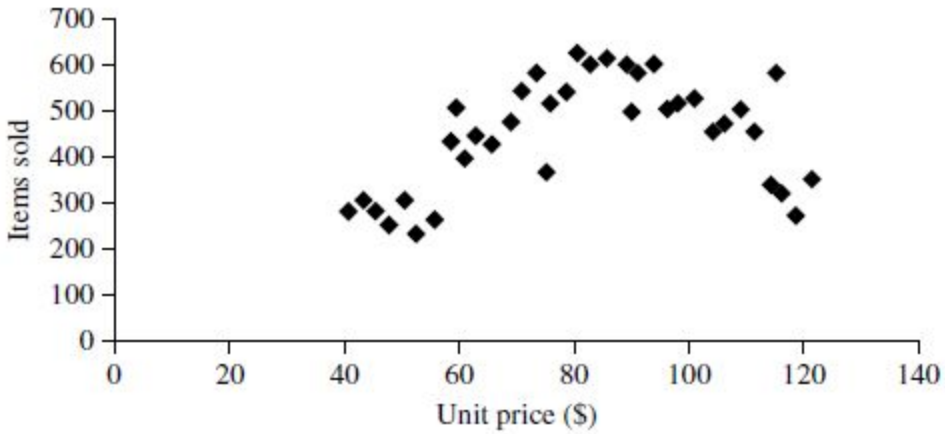


Scatter plot

- 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

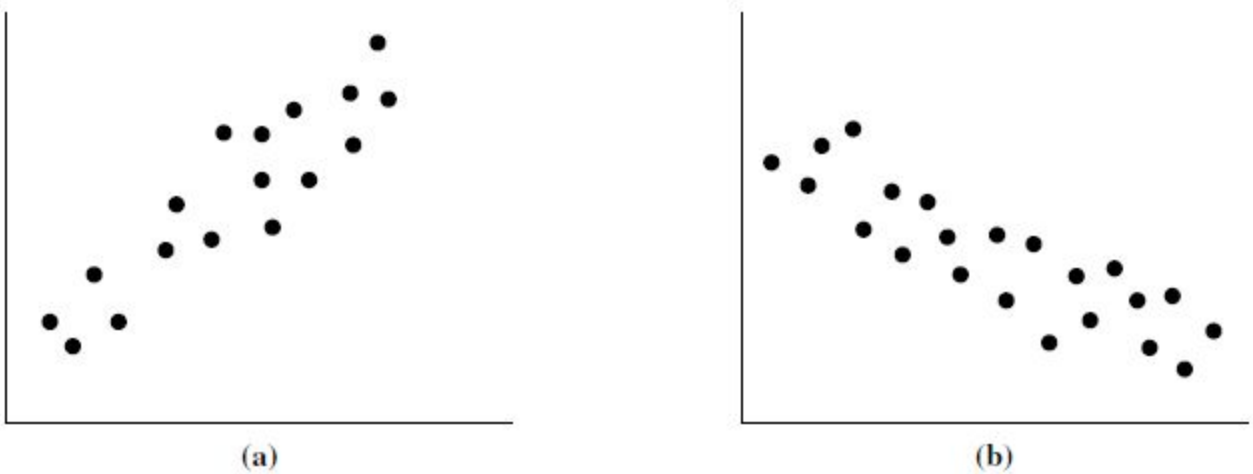


Positively and Negatively Correlated Data

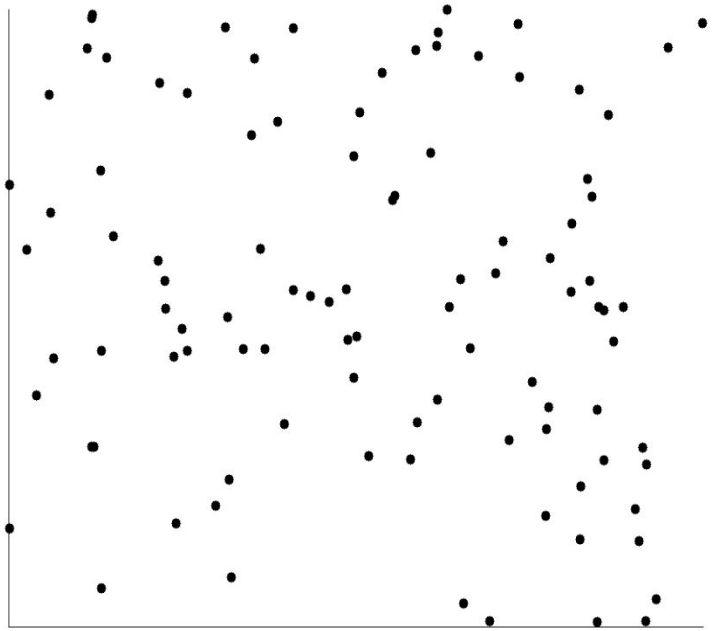


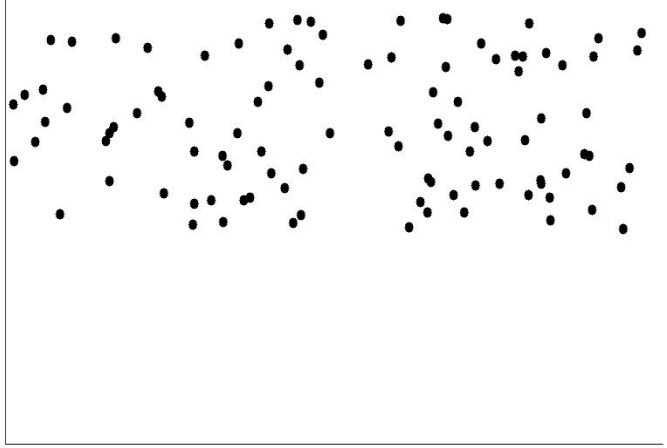
A scatter plot for the Table 2.1 data set.

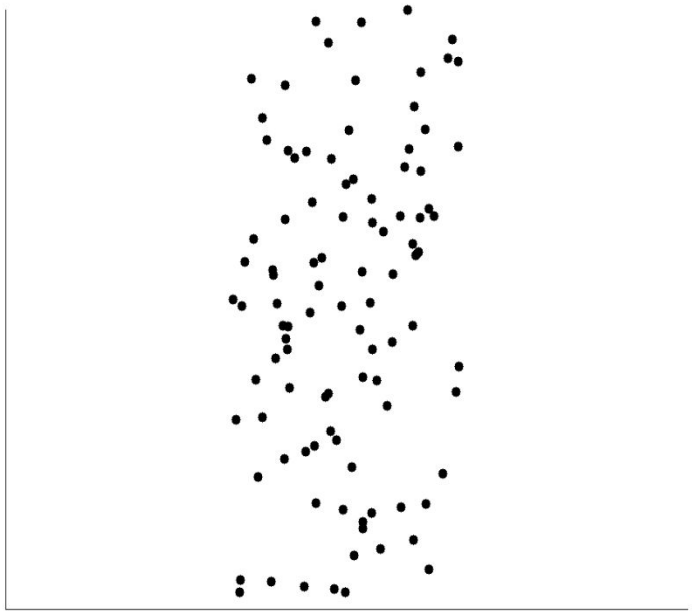
- The left half fragment is **positively correlated**
- The right half is **negative correlated**



Uncorrelated Data







Information-Theoretic Approaches to Data Mining

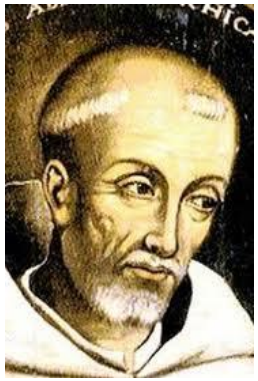
- The Uncertainty Approach
  - Data mining is aimed at reducing uncertainty of the target (predicted) variables
  - Uncertainty can be represented by entropy
  - Data mining algorithms look for models that minimize entropy or maximize mutual information (information gain)
  - Usage: ID3, C4.5, IFN, etc.

Information-Theoretic Approaches to Data Mining (cont.)

- The Data Compression Approach (see Mannila, 2000)
  - Smaller models are more comprehensible to the user
  - The goal of data mining is to *compress the data* by finding some structure (model) for it
  - Data mining algorithms should choose a hypothesis that compresses data the most (the MDL Principle)
  - Usage: Bayesian learning

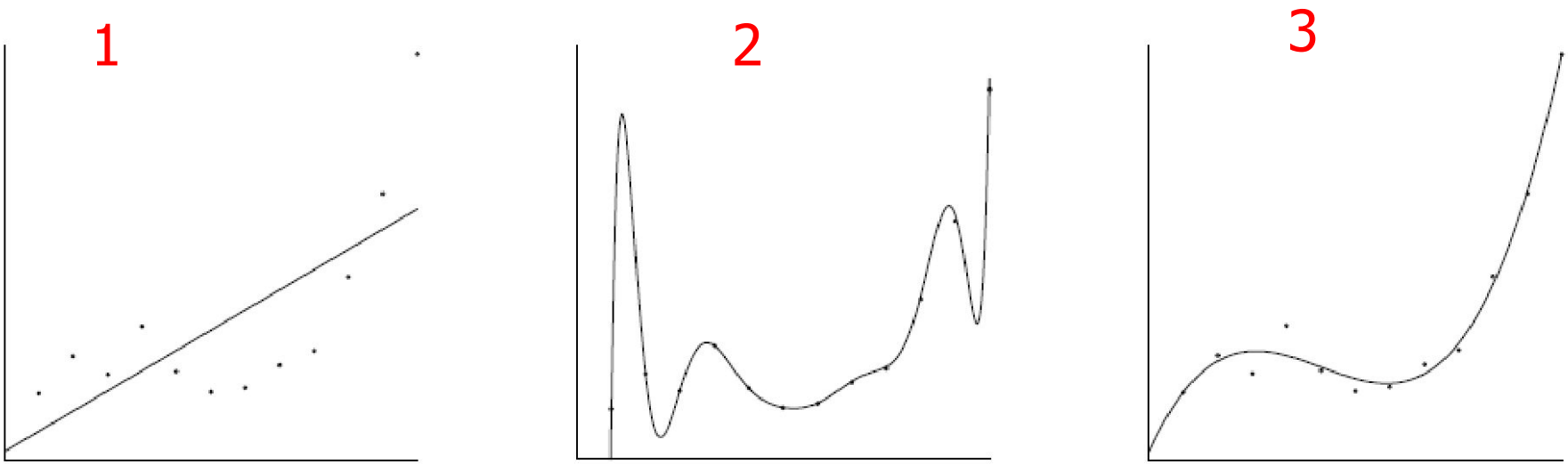
Occam's Razor

- Commonly attributed to William of Ockham (1290--1349). In sharp contrast to the principle of multiple explanations, it states:
  - *Entia non sunt multiplicanda praeter necessitatem*
  - Entities should not be multiplied beyond necessity.
- Commonly explained as:
  - when have choices, choose the simplest theory.
- Bertrand Russell: ``It is vain to do with more what can be done with fewer.''
- Newton (*Principia*):
  - *Natura enim simplex est, et rerum causis superfluis non luxuriat*
  - הטבע פשוט ואין לו עודף של סיבות מיותרות למהות הדברים



The Data Compression Approach

- Example: regression (line fitting)
  - Which model is the best?
    - Model selection and overfitting
    - Complexity of the model vs. Goodness of fit



Source: Grunwald et al. (2005) *Advances in Minimum Description Length: Theory and Applications.*

Minimum Description Length (MDL) Principle

- Problem Statement
  - The attribute values in each case are available to both a *sender* and a *receiver*
  - Only the sender knows the class to which each case belongs
  - The sender must transmit the classification information to the receiver *by using a minimum number of bits*
- Decision Variable
  - The model instance ("hypothesis") to be used by the "sender" out of a given family of models (e.g., decision trees, info-fuzzy networks, *k*th degree polynomials, etc.)

The MDL Preliminaries

- *L (h)* - the length, in bits, of the description of the hypothesis (the *theory cost*)
  - Also called *parametric complexity* (measure of the model "richness", related to the number of model parameters)
  - Example – decision tree: *L (h) = f (number of nodes)*
- In the case of noisy data, the *exceptions* to the hypothesis should also be transmitted
- *L (D/h)* - the length, in bits, of the description of the data under the assumption that both the sender and the receiver know the hypothesis (encoded with the help of the hypothesis)
  - Complex hypotheses lead to small *L (D/H)* and vice versa

The MDL Principle

- Choose the hypothesis h_{MDL} which satisfies the following

$$h_{MDL} = \arg\min_{h \in H} \{L_{C_1}(h) + L_{C_2}(D/h)\}$$

  - L –description length (bits)
  - C_1 - the optimal encoding of the hypothesis h
  - C_2 – the optimal encoding of data D given the hypothesis
- Interpretation
  - The MDL principle represents the trade-off between the model complexity and the number of errors committed by it in the training data
- Practical Usage
  - The MDL principle proved to be an efficient tool for dealing with the problem of *overfitting*

Summary

- Information theory provides a nice formal framework for the process of data mining from both the uncertainty reduction aspect and the aspect of data compression
- The usage of information-theoretic heuristics in numerous data mining algorithms has brought satisfactory results in terms of predictive accuracy and model compactness
- Many other aspects of the information theory are still waiting for their implementation by the KDD researchers and practitioners (see Song et al., 2010)