# MATES ED2MIT

Education and Training for Data Driven Maritime Industry

## Tutorial D03

## Data Preparation and Processing

Yuri Demchenko MATES Project
University of Amsterdam

Maritime Alliance for fostering the European Blue economy through a Marine Technology Skilling Strategy

Co-funded by the Erasmus+ Programme of the European Union

# Outline

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary



Co-funded by the Erasmus+ Programme of the European Union

# Goals

- Understand task related to data preparation and data preprocessing
- Understand techniques used to preprocess data

# Data Preparation

- Coding
- Cleaning
- Preprocessing

# Coding

- Coding – process of translating information gathered from questionnaires or other sources into something that can be analyzed
- Involves assigning a value to the information given—often value is given a label
- Coding can make data more consistent:
  - Example: Question = Sex
  - Answers = Male, Female, M, or F
  - Coding will avoid such inconsistencies

# Coding Systems

- Common coding systems (code and label) for dichotomous variables:
  - 0=No 1=Yes

    (1 = value assigned, Yes= label of value)
  - OR: 1=No 2=Yes
- When you assign a value you must also make it clear what that value means
  - In first example above, 1=Yes but in second example 1=No
  - As long as it is clear how the data are coded, either is fine
- You can make it clear by creating a data dictionary to accompany the dataset

# Coding: Dummy Variables

- A "dummy" variable is any variable that is coded to have 2 levels (yes/no, male/female, etc.)
- Dummy variables may be used to represent more complicated variables
  - Example: # of cigarettes smoked per week--answers total 75 different responses ranging from 0 cigarettes to 3 packs per week
  - Can be recoded as a dummy variable:

    1=smokes (at all)　　　0=non-smoker
- This type of coding is useful in later stages of analysis

# Coding: Attaching ~~Labels~~ to Values

- Many analysis software packages allow you to attach a label to the variable values
  Example: Label 0's as male and 1's as female

- Makes reading data output easier:

| Without Label | Variable Sex | Frequency | Percent |
|---|---|---|---|
| | 0 | 21 | 60% |
| | 1 | 14 | 40% |

| Without Label | Variable Sex | Frequency | Percent |
|---|---|---|---|
| | Male | 21 | 60% |
| | Female | 14 | 40% |

# Coding: Ordinal Variables (1)

- Coding process is similar with other categorical variables
- Example: variable EDUCATION, possible coding:

  0 = Did not graduate from high school
  1 = High school graduate
  2 = Some college or post-high school education
  3 = College graduate

- Could be coded in reverse order (0=college graduate, 3=did not graduate high school)
- For this ordinal categorical variable we want to be consistent with numbering because the value of the code assigned has significance

# Coding: Ordinal Variables (2)

- Example of bad coding:

  0 = Some college or post-high school education

  1 = High school graduate

  2 = College graduate

  3 = Did not graduate from high school

- Data has an inherent order but coding does not follow that order—NOT appropriate coding for an ordinal categorical variable

# Coding: Nominal Variables

- For coding nominal variables, order makes no difference
- Example: variable RESIDE

  1 = Northeast

  2 = South

  3 = Northwest

  4 = Midwest

  5 = Southwest
- Order does not matter, no ordered value associated with each response

# Coding: Continuous Variables (1)

- Creating categories from a continuous variable (ex. age) is common
- May break down a continuous variable into chosen categories by creating an ordinal categorical variable
- Example: variable = AGECAT

  1 = 0–9 years old

  2 = 10–19 years old

  3 = 20–39 years old

  4 = 40–59 years old

  5 = 60 years or older

# Coding: Continuous Variables (2)

- May need to code responses from fill-in-the-blank and open-ended questions
  - Example: “Why did you choose not to see a doctor about this illness?”
- One approach is to group together responses with similar themes
  - Example: “didn’t feel sick enough to see a doctor”, “symptoms stopped,” and “illness didn’t last very long”
  - Could all be grouped together as “illness was not severe”
- Also need to code for “don’t know” responses”
  - Typically, “don’t know” is coded as 9

# Coding Tip for Survey Data

- Although you do not need to code until the data is gathered, you should think about *how* you are going to code while designing your questionnaire, before you gather any data.
  - This will help you to collect the data in a format you can use.

# Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
  - Accuracy: correct or wrong, accurate or not
  - Completeness: not recorded, unavailable, …
  - Consistency: some modified but some not, dangling, …
  - Timeliness: timely update?
  - Believability: how trustable the data are correct?
  - Interpretability: how easily the data can be understood?

# Major Tasks in Data Preprocessing

- **Data cleaning**
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
  - Integration of multiple databases, data cubes, or files
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

# Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=“ ” (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g., *Salary*=“−10” (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - *Age*=“42”, *Birthday*=“03/07/2010”
    - Was rating “1, 2, 3”, now rating “A, B, C”
    - discrepancy between duplicate records
  - Intentional (e.g., *disguised missing* data)
    - Jan. 1 as everyone’s birthday?

# Data Cleaning (1)

- One of the first steps in analyzing data is to “clean” it of any obvious data entry errors:
  - Outliers? (really high or low numbers)
    Example:  Age = 110 (really 10 or 11?)
  - Value entered that doesn’t exist for variable?
    Example: 2 entered where 1=male, 0=female
  - Missing values?
    Did the person not give an answer? Was answer accidentally not entered into the database?

# Data Cleaning (2)

- May be able to set defined limits when entering data
  - Prevents entering a 2 when only 1, 0, or missing are acceptable values
- Limits can be set for continuous and nominal variables
  - Examples: Only allowing 3 digits for age, limiting words that can be entered, assigning field types (e.g. formatting dates as mm/dd/yyyy or specifying numeric values or text)
- Many data entry systems allow “double-entry” – ie., entering the data twice and then comparing both entries for discrepancies
- Univariate data analysis is a useful way to check the quality of the data

# Incomplete (Missing) Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not registered history or changes of the data
- Missing data may need to be inferred

# How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  – a global constant : e.g., “unknown”, a new class?!
  – the attribute mean
  – the attribute mean for all samples belonging to the same class: smarter
  – the most probable value: inference-based such as Bayesian formula or decision tree

# Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which require data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

# How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

# Data Cleaning as a Process

- Data discrepancy detection
  - Use metadata (e.g., domain, range, dependency, distribution)
  - Check field overloading, i.e. consistency across whole datasets
  - Check uniqueness rule, consecutive rule and null rule
  - Use Open Data
    - Such as cense data, taxation data for checking names, regions, addresses, etc
  - Use commercial tools
    - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
    - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)
- Data migration and integration
  - Data migration tools: allow transformations to be specified
  - ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface
- Integration of the two processes
  - Iterative and interactive (e.g., Potter's Wheels data cleaning tool A-B-C)

# Data Integration

- **Data integration**:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id ≡ B.cust-#
  - Integrate metadata from different sources
- Entity identification problem:
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton, Bob Johns = Robert Johns
- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

# Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - *Object identification*:  The same attribute or object may have different names in different databases
  - *Derivable data:* One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by *correlation analysis* and *covariance analysis*
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

# Correlation Analysis (Nominal Data)

- **$X^2$ (chi-square) test**

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- The larger the $X^2$ value, the more likely the variables are related
- The cells that contribute the most to the $X^2$ value are those whose actual count is very different from the expected count
  - Expected count is defined by standard probability for the whole population
- Correlation does not imply causality
  - # of hospitals and # of car-theft in a city are correlated
  - Both are causally linked to the third variable: population

# Chi-Square Calculation: An Example

| | Play chess | Not play chess | Sum (row) |
|---|---|---|---|
| Like science fiction | 250(90) | 200(360) | 450 |
| Not like science fiction | 50(210) | 1000(840) | 1050 |
| Sum(col.) | 300 | 1200 | 1500 |

- $X^2$ (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group

# Chi-Square Calculation: An Example

| | Play chess | Not play chess | Sum (row) |
|---|---|---|---|
| Like science fiction | 250(90) | 200(360) | 450 |
| Not like science fiction | 50(210) | 1000(840) | 1050 |
| Sum(col.) | 300 | 1200 | 1500 |

Red is expected standard count

- $X^2$ (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group

# Correlation Analysis (Numeric Data)

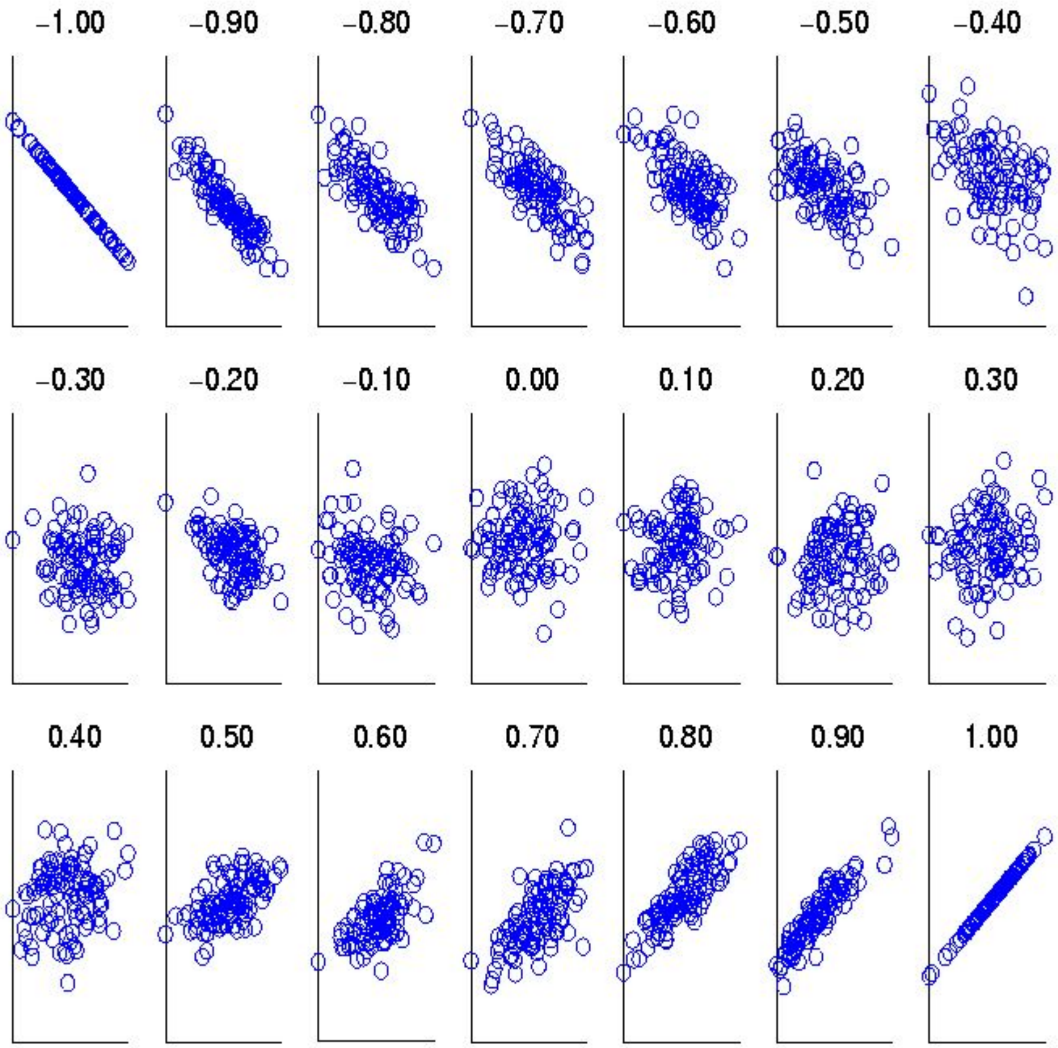
- Correlation coefficient (also called Pearson's product moment coefficient)

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{(n-1)\sigma_A \sigma_B}$$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective means of A and B, $\sigma_A$ and $\sigma_B$ are the respective standard deviation of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

# Visually Evaluating Correlation




Scatter plots showing the similarity from –1 to 1.

# Correlation (viewed as linear relationship)

- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, A and B, and then take their dot product

$$a'_k = (a_k - mean(A)) / std(A)$$

$$b'_k = (b_k - mean(B)) / std(B)$$

$$correlation(A, B) = A' \bullet B'$$

# Covariance (Numeric Data)

- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

Correlation coefficient: $$r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean or **expected values** of A and B, $\sigma_A$ and $\sigma_B$ are the respective standard deviation of A and B.

- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- **Negative covariance**: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- **Independence**: $Cov_{A,B} = 0$ but the converse is not true:
  - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

# Co-Variance: An Example

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

- It can be simplified in computation as

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}$$

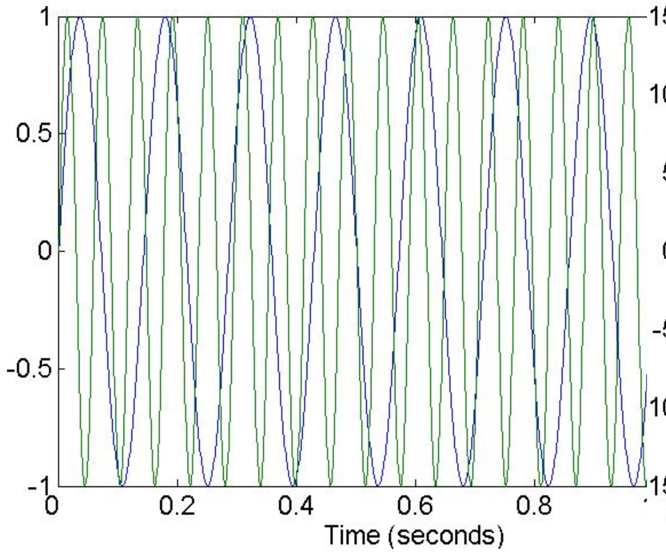
- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?
  - E(A) = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - E(B) = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6
  - Cov(A,B) = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4
- Thus, A and B rise together since Cov(A, B) > 0.

# Data Reduction Strategies

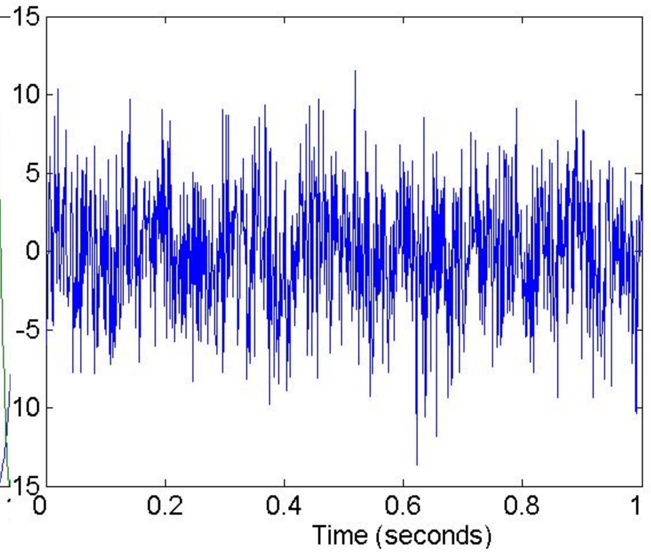
- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
  - **Dimensionality reduction**, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
  - **Numerosity reduction** (some simply call it: Data Reduction)
    - Regression and Log-Linear Models
    - Histograms, clustering, sampling
    - Data cube aggregation
  - **Data compression**

# Mapping Data to a New Space

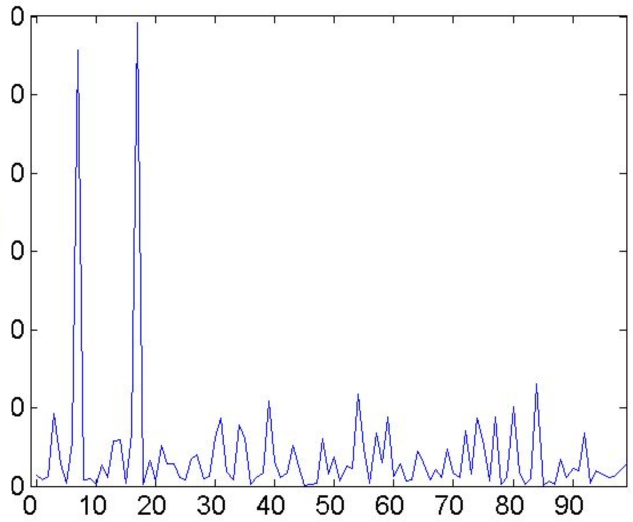
- Fourier transform
- Wavelet transform




Two Sine Waves



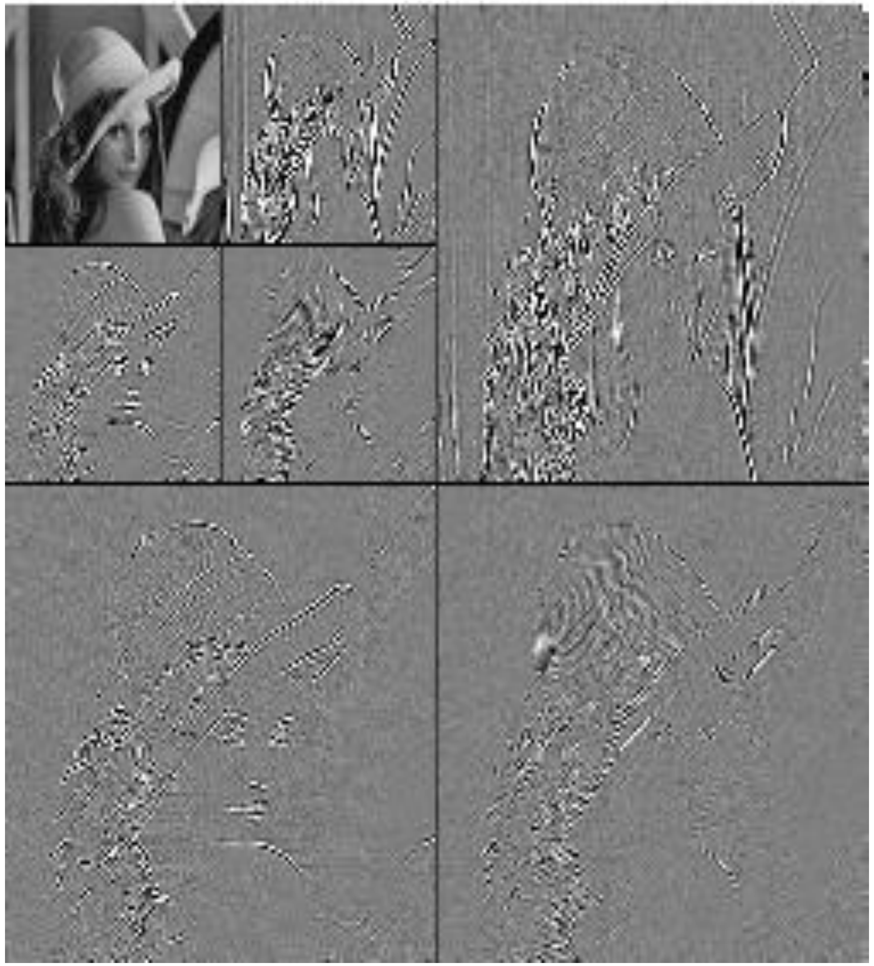

Two Sine Waves + Noise




Frequency

# What Is Wavelet Transform?

- Decomposes a signal into different frequency subbands
  - Applicable to n-dimensional signals
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allow natural clusters to become more distinguishable
- Used for image compression

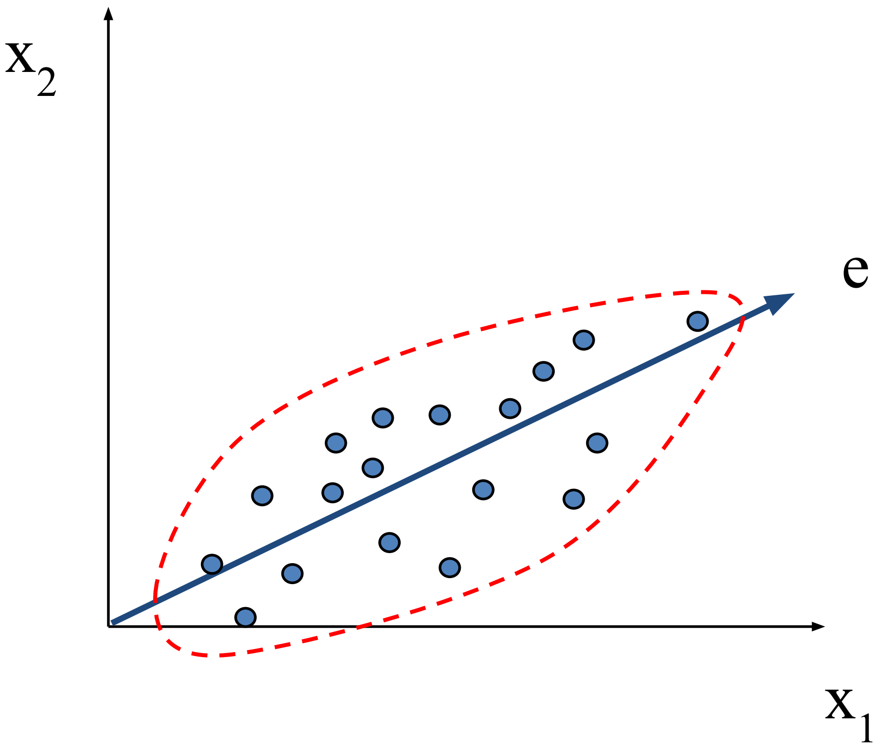


# Why Wavelet Transform?

- Use hat-shape filters
  - Emphasize region where points cluster
  - Suppress weaker information in their boundaries
- Effective removal of outliers
  - Insensitive to noise, insensitive to input order
- Multi-resolution
  - Detect arbitrary shaped clusters at different scales
- Efficient
  - Complexity O(N)
- Only applicable to low dimensional data

# Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



# Principal Component Analysis (Steps)

- Given $N$ data vectors from $n$-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute $k$ orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the $k$ principal component vectors
  - The principal components are sorted in order of decreasing “significance” or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)
- Works for numeric data only

# Attribute Subset Selection

- Another way to reduce dimensionality of data
- Redundant attributes
  – Duplicate much or all of the information contained in one or more other attributes
  – E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
  – Contain no information that is useful for the data mining task at hand
  – E.g., students' ID is often irrelevant to the task of predicting students' GPA (Grade Point Average)

# Attribute Creation (Feature Generation)

- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
  - Attribute extraction
    - Domain-specific
  - Mapping data to new space (see: data reduction)
    - E.g., Fourier transformation, wavelet transformation, manifold approaches (not covered)
  - Attribute construction
    - Combining features (see: discriminative frequent patterns in Chapter 7)
    - Data discretization

# Parametric Data Reduction: Regression and Log-Linear Models

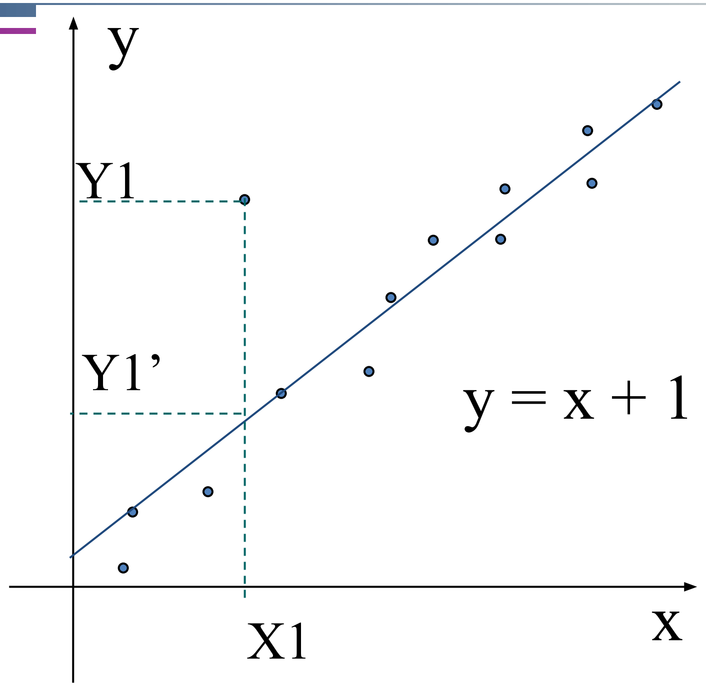
- **Linear regression**
  - Data modeled to fit a straight line
  - Often uses the least-square method to fit the line
- **Multiple regression**
  - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- **Log-linear model**
  - Approximates discrete multidimensional probability distributions

# Regression Analysis and Log-Linear Models

- Linear regression: $Y = w X + b$
  - Two regression coefficients, *w* and *b*, specify the line and are to be estimated by using the data at hand
  - Using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
- Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$
  - Many nonlinear functions can be transformed into the above
- Log-linear models:
  - Approximate discrete multidimensional probability distributions
  - Estimate the probability of each point (tuple) in a multi-dimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations
  - Useful for dimensionality reduction and data smoothing

# Regression Analysis

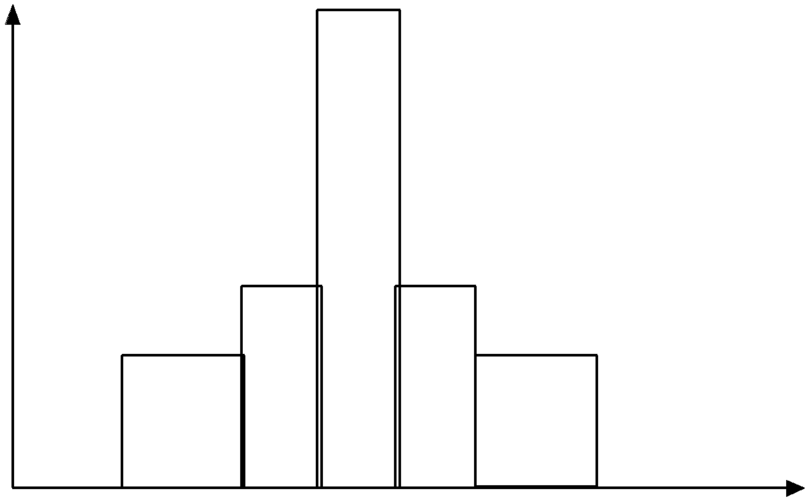
- Regression analysis: A collective name for techniques for the modeling and analysis of numerical data consisting of values of a ***dependent variable*** (also called ***response variable*** or *measurement*) and of one or more *independent variables* (aka. ***explanatory variables*** or ***predictors***)
- The parameters are estimated so as to give a "**best fit**" of the data
- Most commonly the best fit is evaluated by using the ***least squares method***, but other criteria have also been used




- Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships

# Histogram Analysis

- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
  - Equal-width: equal bucket range
  - Equal-frequency (or equal-depth)



# Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms
- Cluster analysis will be studied in depth in Chapter 10

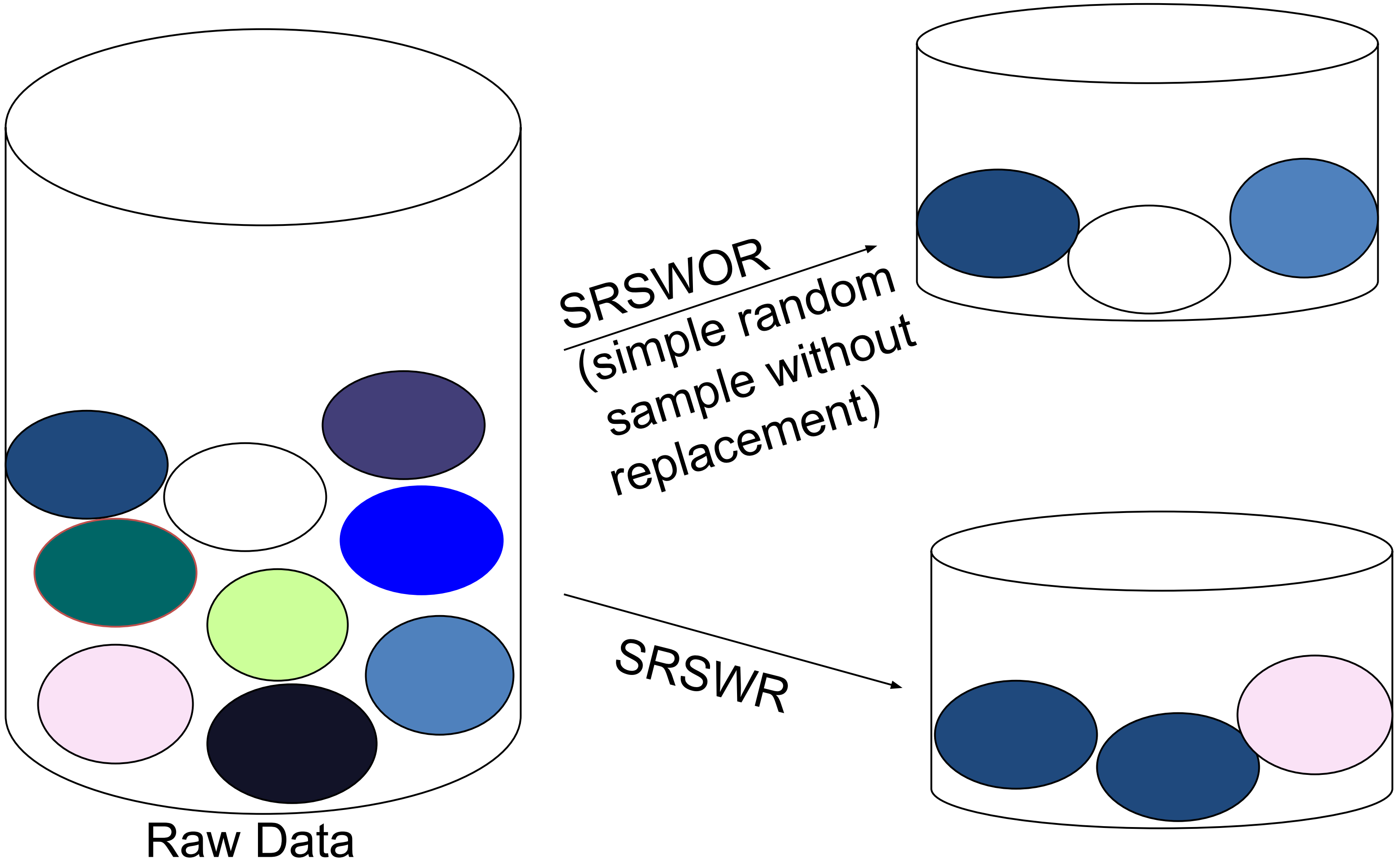
# Sampling

- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
  - Develop adaptive sampling methods, e.g., stratified sampling:
- Note: Sampling may not reduce database I/Os (page at a time)

# Types of Sampling

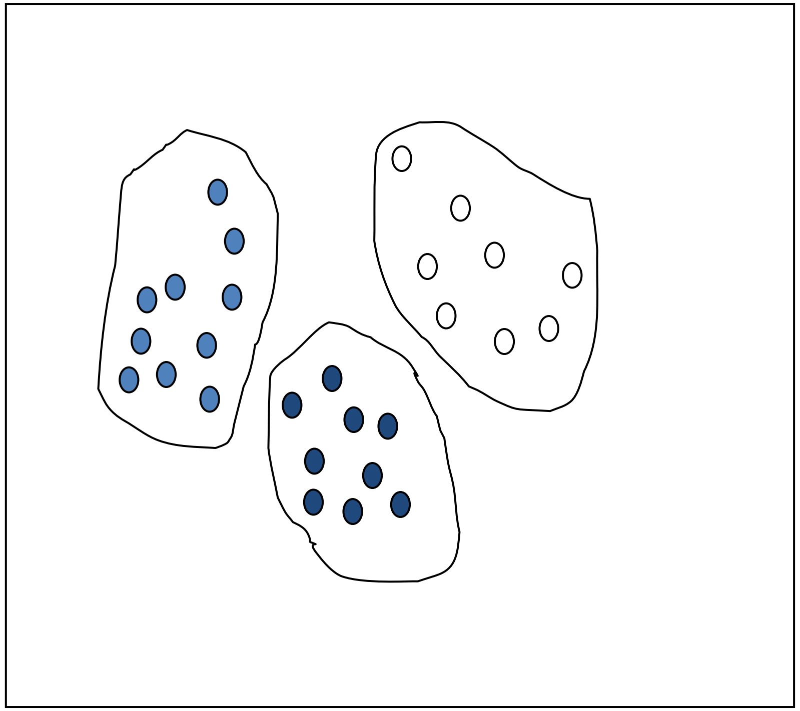
- **Simple random sampling**
  - There is an equal probability of selecting any particular item
- **Sampling without replacement**
  - Once an object is selected, it is removed from the population
- **Sampling with replacement**
  - A selected object is not removed from the population
- **Stratified sampling:**
  - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)
  - Used in conjunction with skewed data

# Sampling: With or without Replacement

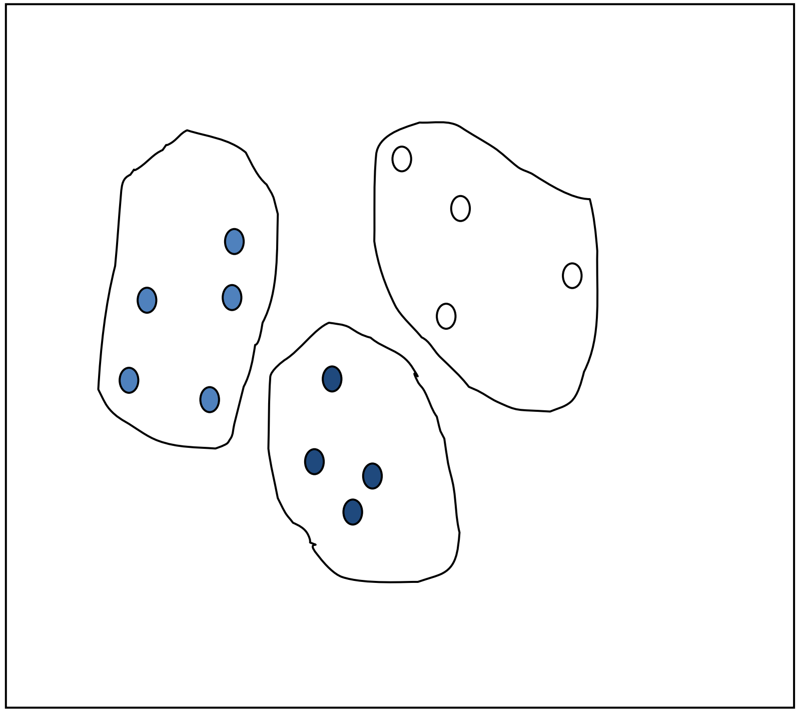


# Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample



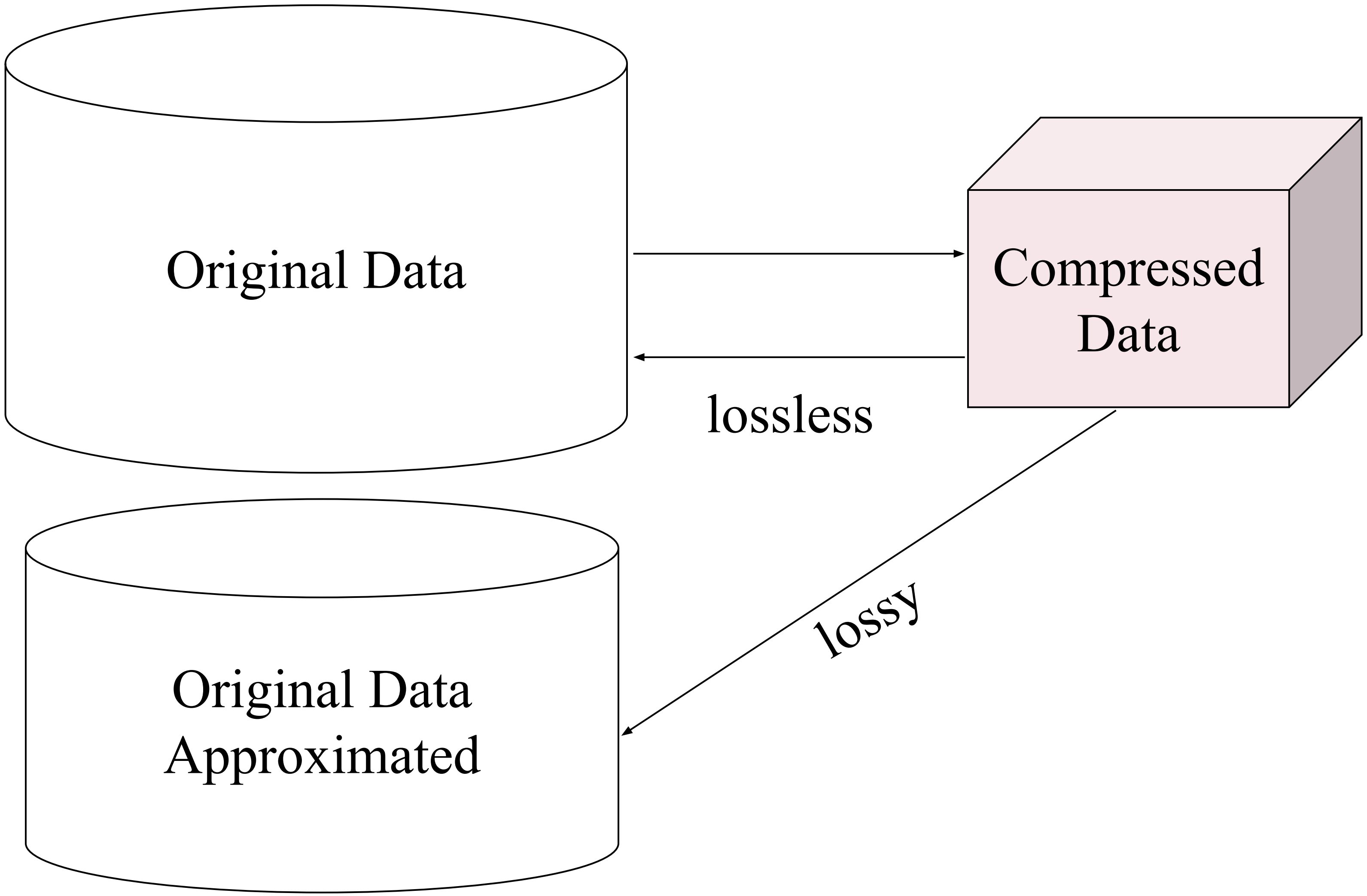
# Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

# Data Reduction: Data Compression

- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless, but only limited manipulation is possible without expansion
- Audio/video compression
  - Typically lossy compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
  - Typically short and vary slowly with time
- Dimensionality and numerosity reduction may also be considered as forms of data compression

# Data Compression



# Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values s. t. each old value can be identified with one of the new values
- Methods
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization, data cube construction
  - Normalization: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - Discretization: Concept hierarchy climbing

# Normalization

- **Min-max normalization**: to [new_min$_A$, new_max$_A$]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  – Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73,600 - 12,000}{98,000 - 12,000}(1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  – Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73,600 - 54,000}{16,000} = 1.225$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j}$$ where $j$ is the smallest integer such that Max(|ν'|) < 1

# Discretization

- Three types of attributes
  - Nominal—values from an unordered set, e.g., color, profession
  - Ordinal—values from an ordered set, e.g., military or academic rank
  - Numeric—real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

# Data Discretization Methods

- Typical methods: All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Histogram analysis
    - Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g., $\chi^2$) analysis (unsupervised, bottom-up merge)

# Simple Discretization: Binning

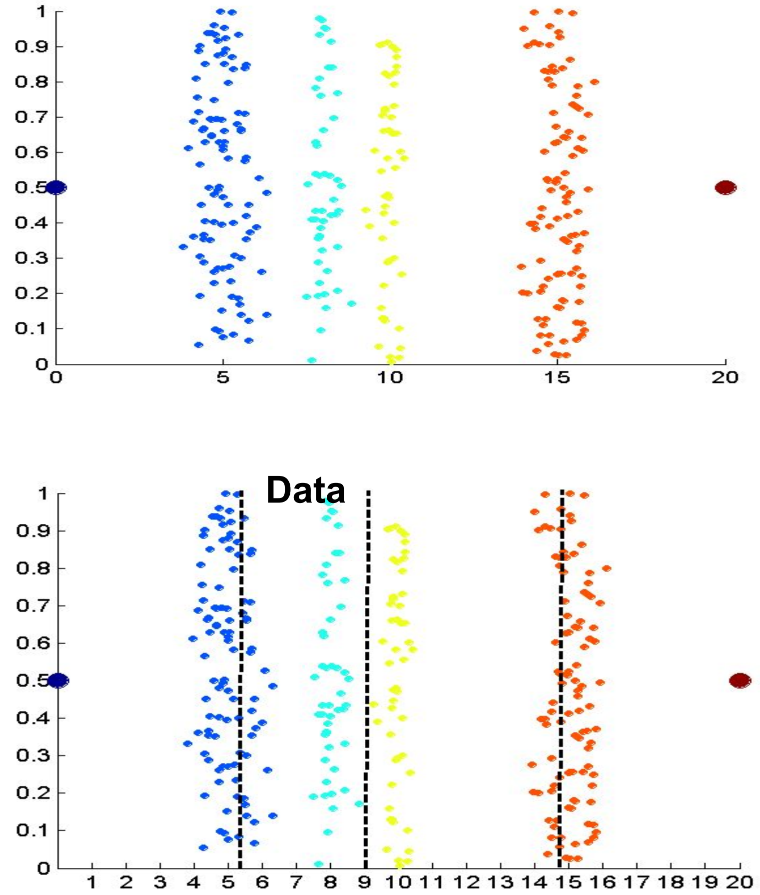
- Equal-width (distance) partitioning
  - Divides the range into $N$ intervals of equal size: uniform grid
  - if $A$ and $B$ are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into $N$ intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

# Binning Methods for Data Smoothing

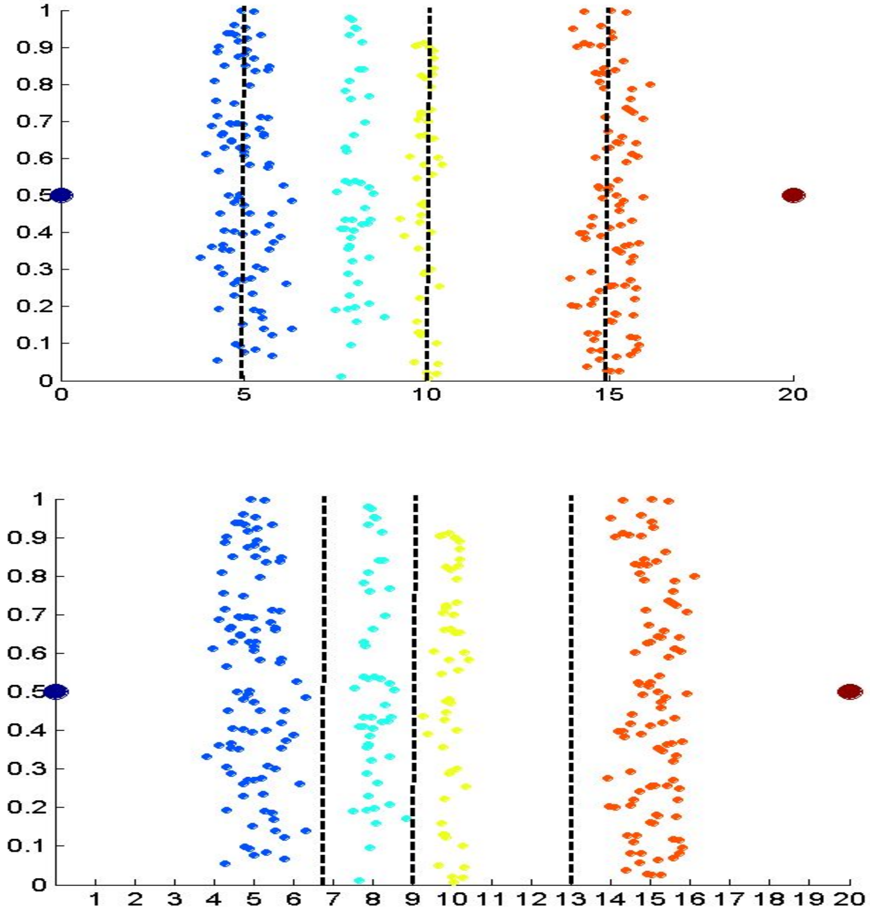
Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (**equi-depth**) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by **bin means**:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by **bin boundaries**:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

# Discretization Without Using Class Labels (Binning vs. Clustering)




**Equal frequency (binning)**



**K-means clustering leads to better results**

# Concept Hierarchy Generation

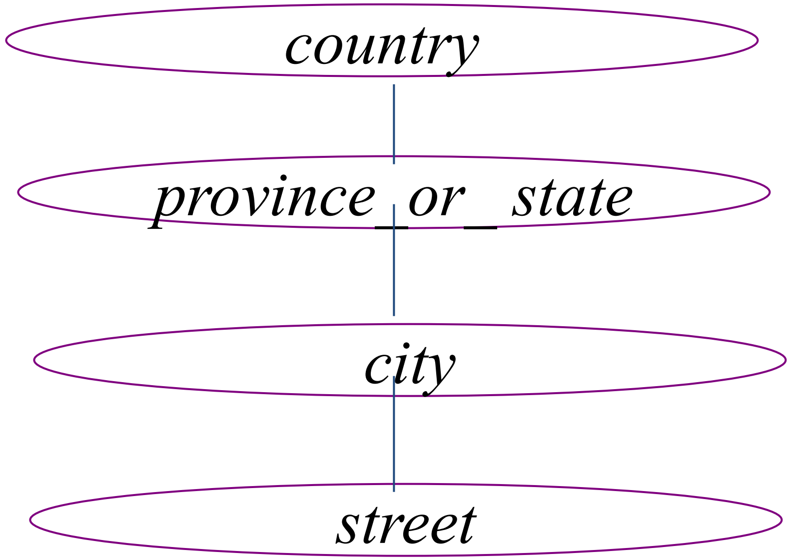
- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively reduce the data by collecting and replacing low level concepts (such as numeric values for *age*) by higher level concepts (such as *youth, adult*, or *senior*)
- Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers
- Concept hierarchy can be automatically formed for both numeric and nominal data.  For numeric data, use discretization methods shown.

# Concept Hierarchy Generation for Nominal Data

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

# Conclusion and Takeaway

- **Data coding**: adding metadata/labels
- **Data quality**: accuracy, completeness, consistency, timeliness, believability, interpretability
- **Data cleaning**: e.g. missing/noisy values, outliers
- **Data integration** from multiple sources:
  - Entity identification problem
  - Remove redundancies
  - Detect inconsistencies
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

# Practice Part – Investigating Selected Dataset

- Use self-study exercises provided for this course both in Python and RapidMiner
- Investigate and visualize dataset characteristics

# References

1.