

Data Preprocessing

Prepared by Dr. Amjad Qtaish

DS601
Data Mining

Lecture 02



Lecture: #02 Data Preprocessing



Dr. Amjad Qtaish

Faculty of Informatics
Midocean University



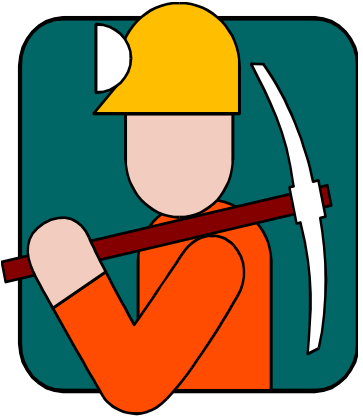
Data Mining (DS601)



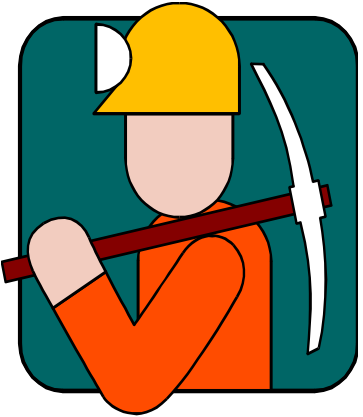
Data Preprocessing

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





- **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?



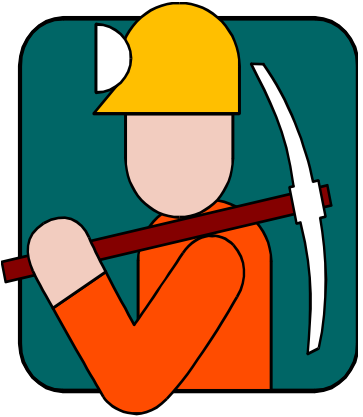
- **Data cleaning**
- **Data integration**
 - Integration of multiple databases, data cubes, or files
 - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data reduction**
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression
- **Data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



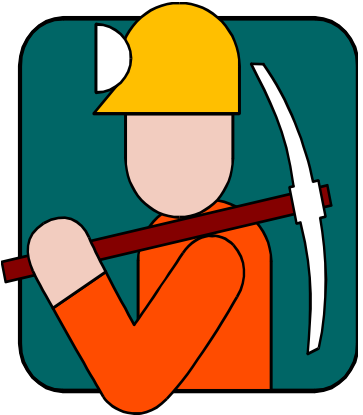
Data Preprocessing

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





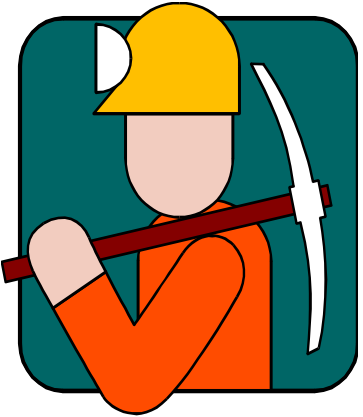
- **Data in the Real World Is Dirty:** Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
 - **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - e.g., *Occupation*=" " (missing data)
 - **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?



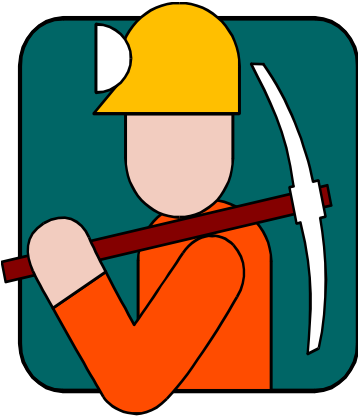
- **Data is not always available**
 - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- **Missing data may be due to**
 - equipment malfunction
 - inconsistent with other recorded data and thus deleted
 - data not entered due to misunderstanding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



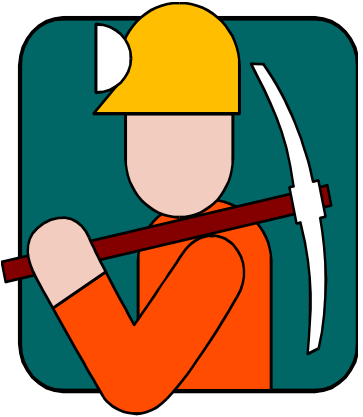
How to Handle Missing Data?



- **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



- **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



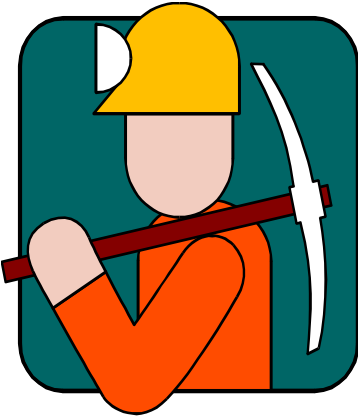
- Binning
 - first sort data and partition into (equal-frequency) bins
 - then one can **smooth by bin means**, **smooth by bin median**, **smooth by bin boundaries**, etc.
- Regression
 - smooth by fitting the data into regression functions
- Clustering
 - detect and remove outliers
- Combined computer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



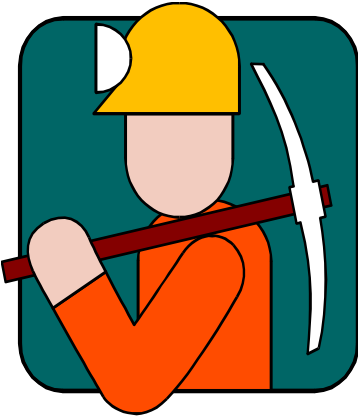
Data Preprocessing

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





- **Data integration:**
 - Combines data from multiple sources into a coherent store
- **Schema integration:** e.g., $A.cust-id \equiv B.cust-\#$
 - Integrate metadata from different sources
- **Entity identification problem:**
 - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- **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



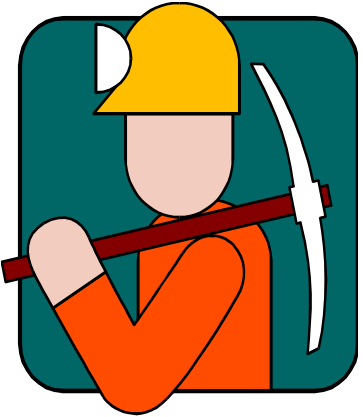
- Redundant data occur often when integration of multiple databases
 - *Object identification*: The same attribute or object may have different names in different databases
 - *Derivable data*: One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant attributes may 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

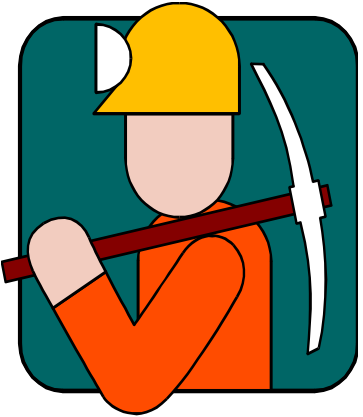


- χ^2 (chi-square) test

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

- The larger the χ^2 value, the more likely the variables are related
- The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count
- 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



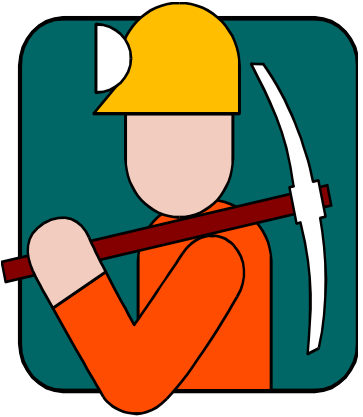


	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- χ^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 coefficient (also called **Pearson's product moment coefficient**)

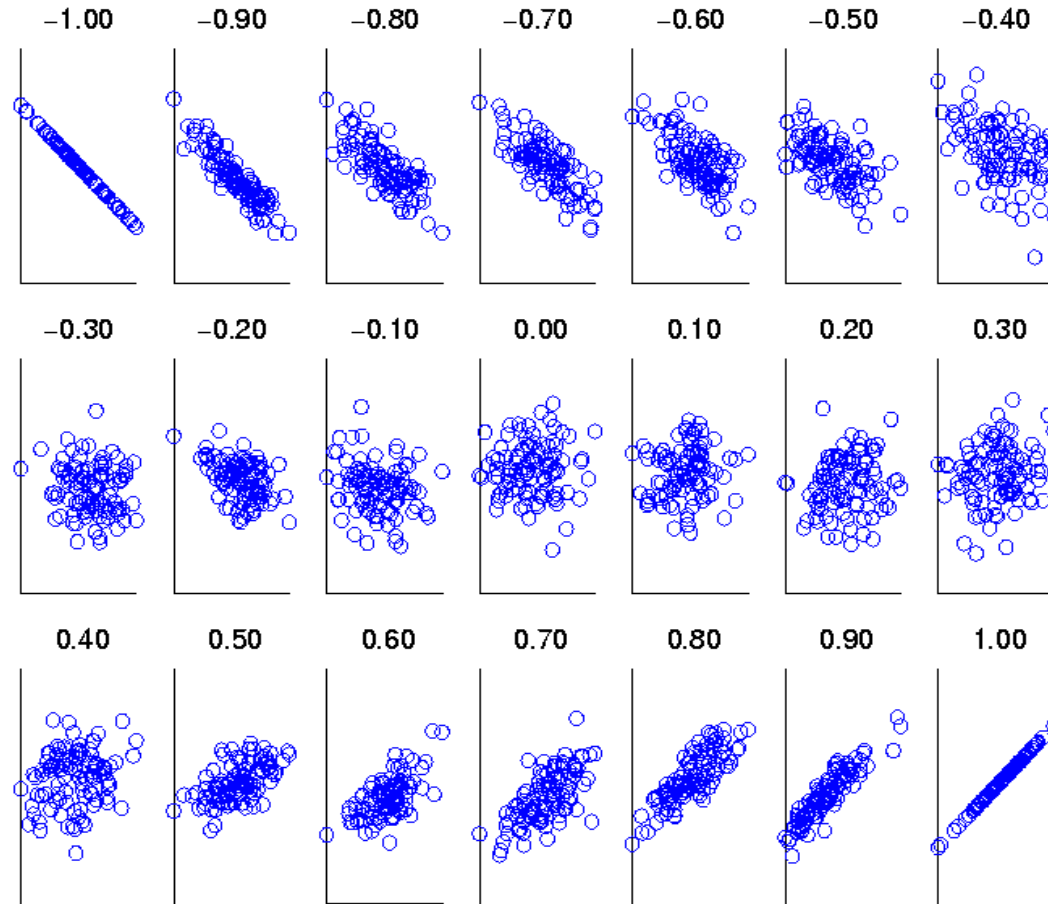
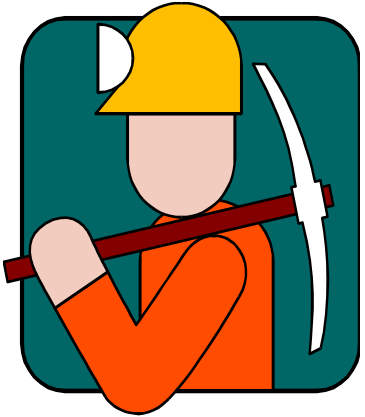
$$r_{A,B} = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^n (a_i b_i) - n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

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

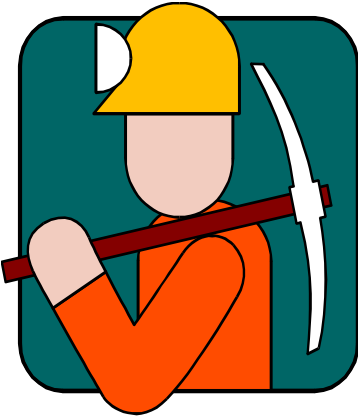
- If $r_{A,B} > 0$, A and B are positively correlated (A 's values increase as B 's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated



Visually Evaluating Correlation



Scatter plots
showing the
similarity from
-1 to 1.

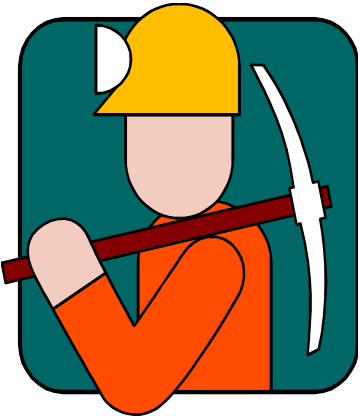


- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, A and B, and then take their dot product

$$a'_k = (a_k - \text{mean}(A)) / \text{std}(A)$$

$$b'_k = (b_k - \text{mean}(B)) / \text{std}(B)$$

$$\text{correlation}(A, B) = A' \bullet B'$$



- Covariance is similar to correlation

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

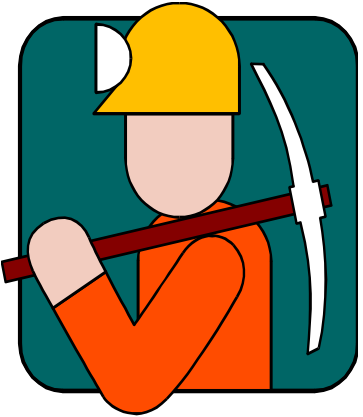
Correlation coefficient: $r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean or **expected values** of A and B , σ_A and σ_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



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



- It can be simplified in computation as

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

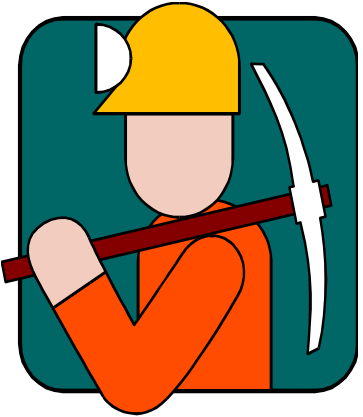
- **Suppose** two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- **Question:** If the stocks are affected by the same industry trends, will their prices rise or fall together?
 - $E(A) = (2 + 3 + 5 + 4 + 6) / 5 = 20 / 5 = 4$
 - $E(B) = (5 + 8 + 10 +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



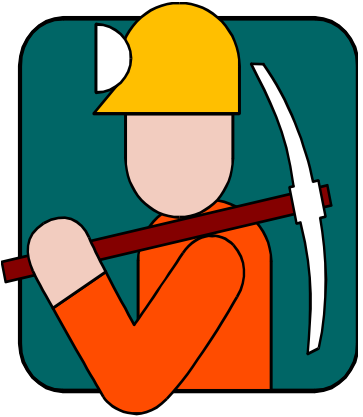
Data Preprocessing

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





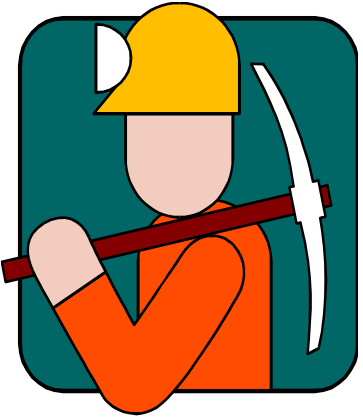
- **Data reduction:** Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run 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**



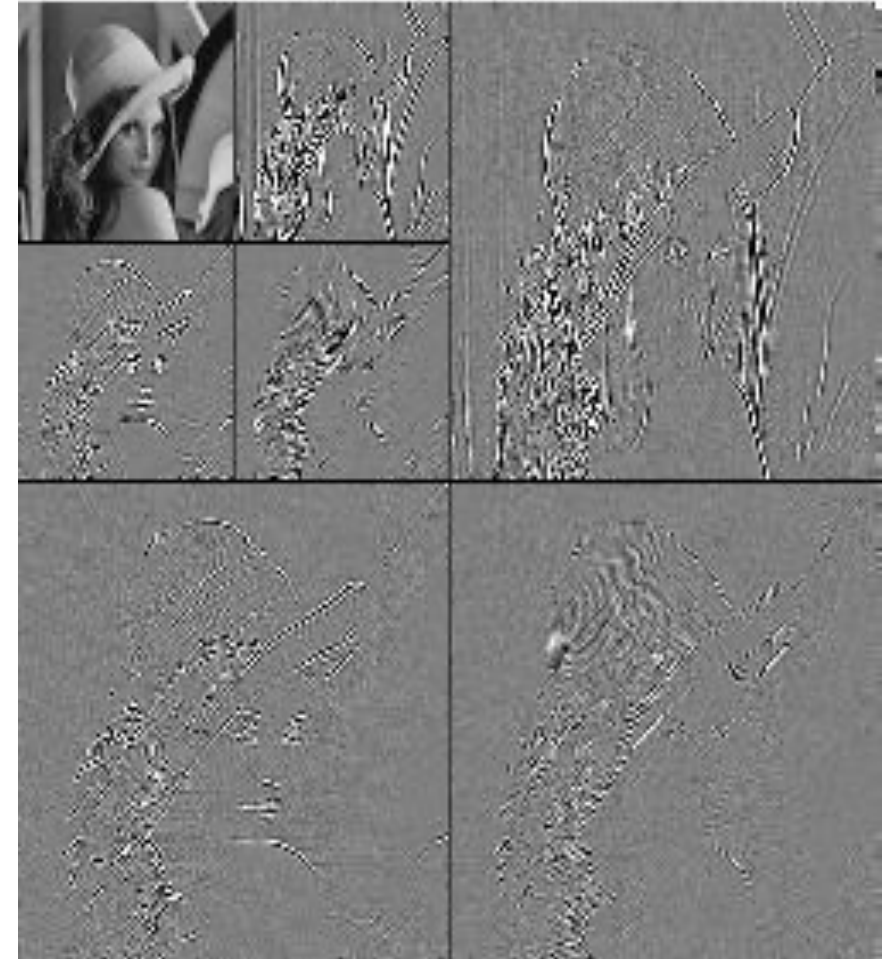
- **Curse of dimensionality**
 - When dimensionality increases, data becomes increasingly sparse
 - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
 - The possible combinations of subspaces will grow exponentially
- **Dimensionality reduction**
 - Avoid the curse of dimensionality
 - Help eliminate irrelevant features and reduce noise
 - Reduce time and space required in data mining
 - Allow easier visualization
- **Dimensionality reduction techniques**
 - Wavelet transforms
 - Principal Component Analysis
 - Supervised and nonlinear techniques (e.g., feature selection)

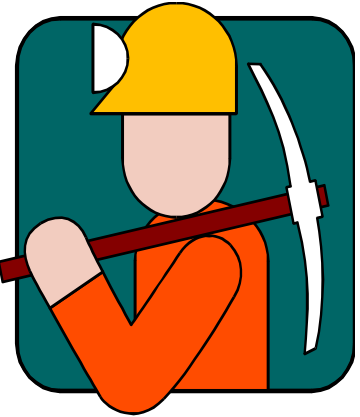


What Is Wavelet Transform?

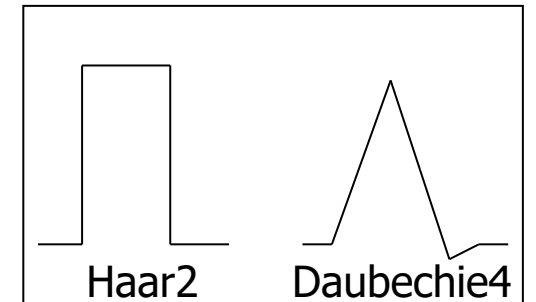


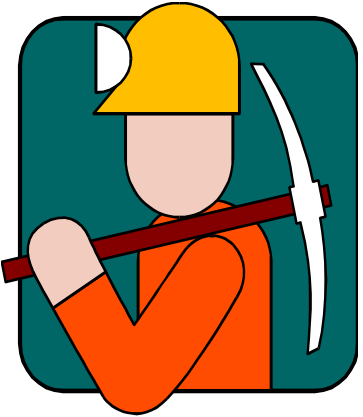
- Decomposes a signal into different frequency subbands
 - Applicable to n-dimensional signals
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allow natural clusters to become more distinguishable
- Used for image compression





- Discrete wavelet transform (DWT) for linear signal processing, multi-resolution analysis
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space
- Method:
 - Length, L , must be an integer power of 2 (padding with 0's, when necessary)
 - Each transform has 2 functions: smoothing, difference
 - Applies to pairs of data, resulting in two set of data of length $L/2$
 - Applies two functions recursively, until reaches the desired length



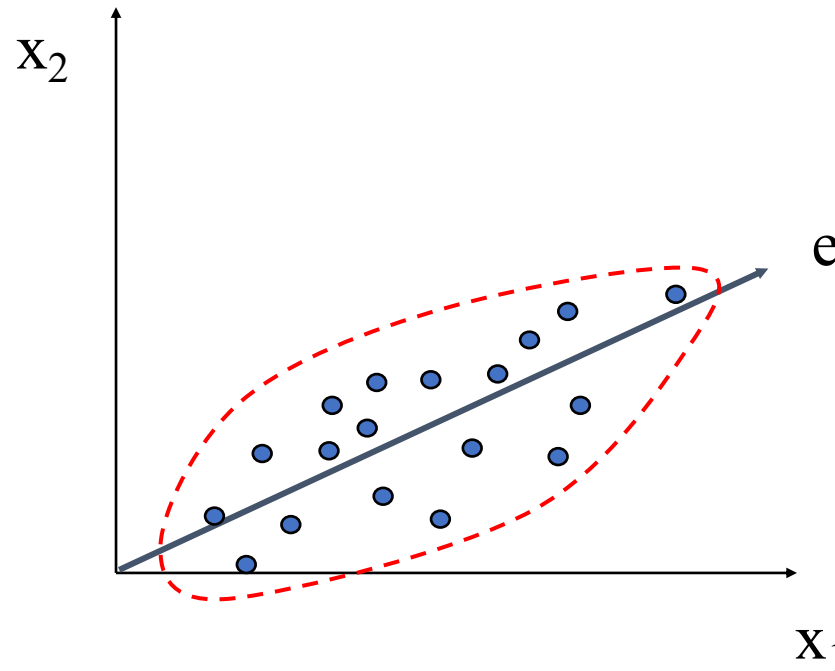
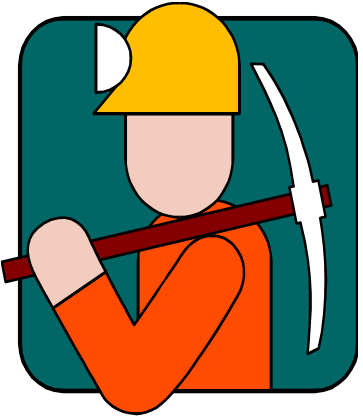


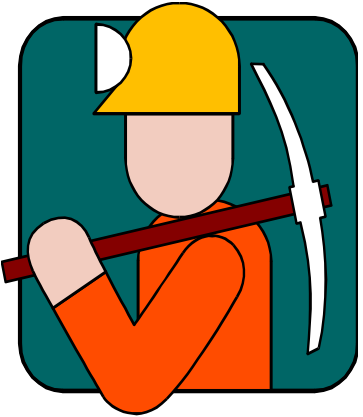
- Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- $S = [2, 2, 0, 2, 3, 5, 4, 4]$ can be transformed to $S_{\wedge} = [2^{3/4}, -1^{1/4}, 1/2, 0, 0, -1, -1, 0]$
- Compression: many small detail coefficients can be replaced by 0's, and only the significant coefficients are retained

Resolution	Averages	Detail Coefficients
8	$[2, 2, 0, 2, 3, 5, 4, 4]$	
4	$[2, 1, 4, 4]$	$[0, -1, -1, 0]$
2	$[1\frac{1}{2}, 4]$	$[\frac{1}{2}, 0]$
1	$[2\frac{3}{4}]$	$[-1\frac{1}{4}]$

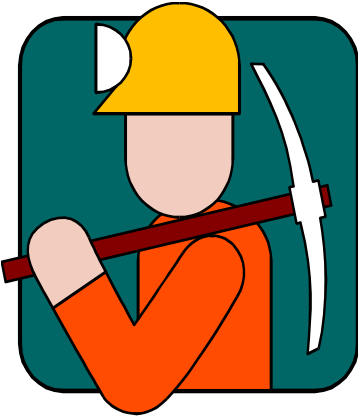


- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space

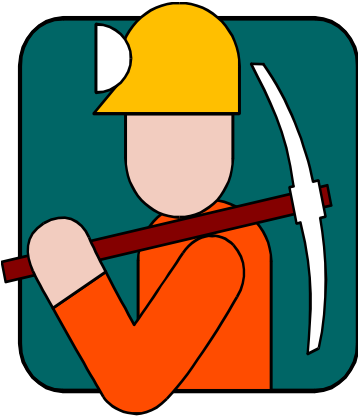




- Given N data vectors from n -dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
 - Normalize input data: Each attribute falls within the same range
 - Compute k orthonormal (unit) vectors, i.e., *principal components*
 - Each input data (vector) 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



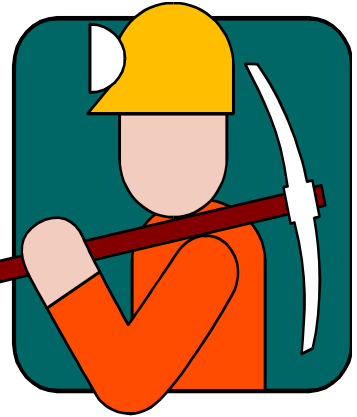
- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** (e.g., regression)
 - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - Ex.: Log-linear models—obtain value at a point in m -D space as the product on appropriate marginal subspaces
- **Non-parametric methods**
 - Do not assume models
 - Major families: histograms, clustering, sampling, ...



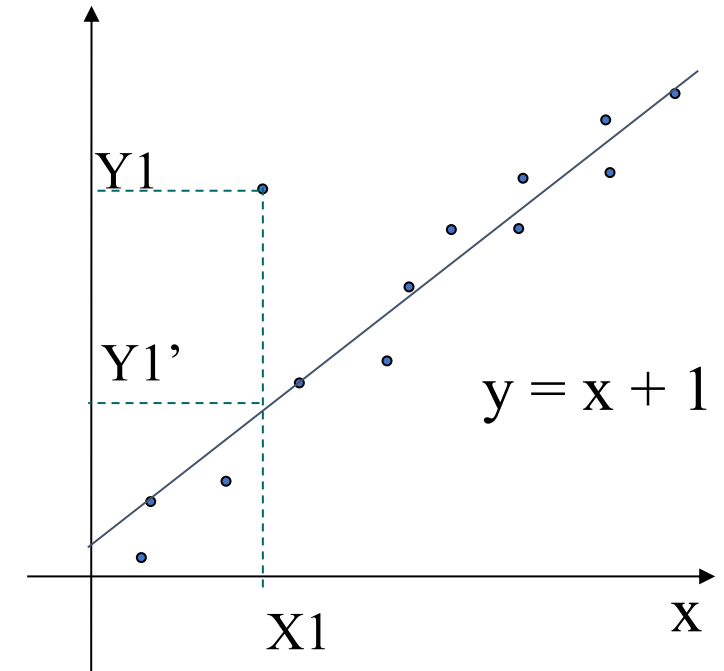
- **Linear regression**
 - Data modeled to fit a straight line
 - Often uses the least-square method to fit the line
- **Multiple regression**
 - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- **Log-linear model**
 - Approximates discrete multidimensional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

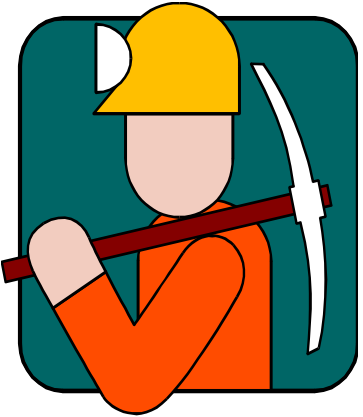


Regression Analysis



- 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

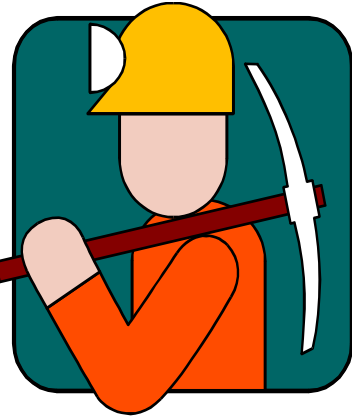




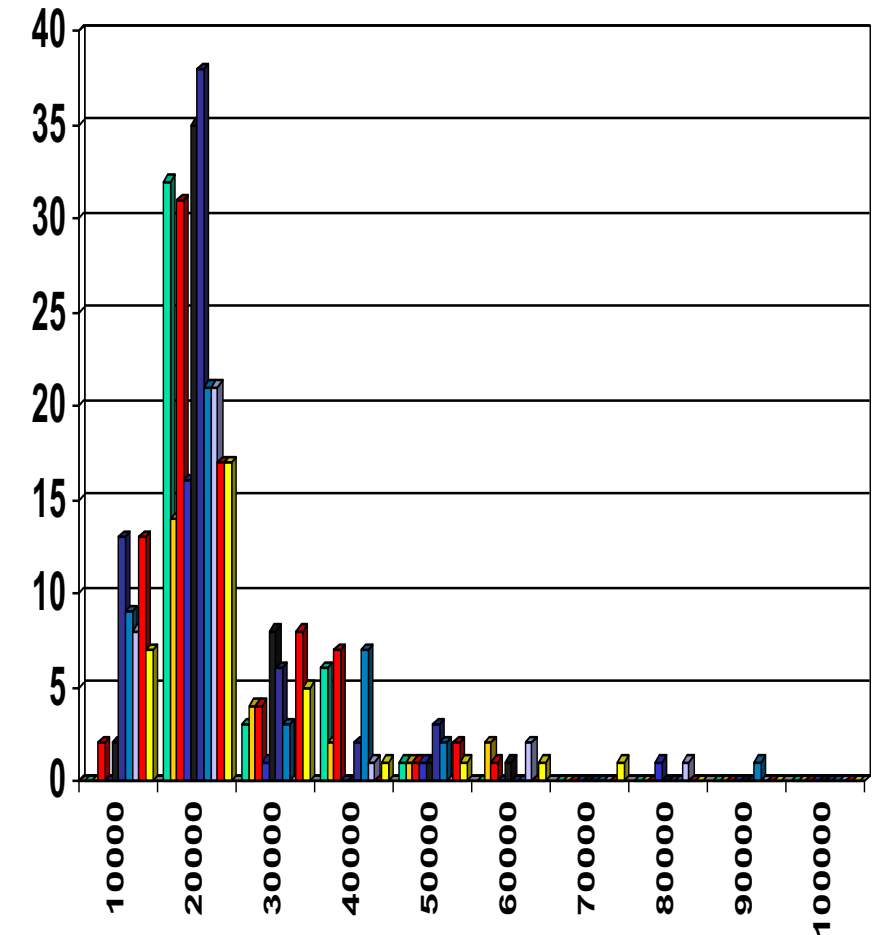
- **Linear regression:** $Y = wX + b$
 - Two regression coefficients, w and b , specify the line and are to be estimated by using the data at hand
 - Using the least squares criterion to the known values of $Y_1, Y_2, \dots, X_1, X_2, \dots$
- **Multiple regression:** $Y = b_0 + b_1 X_1 + b_2 X_2$
 - 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

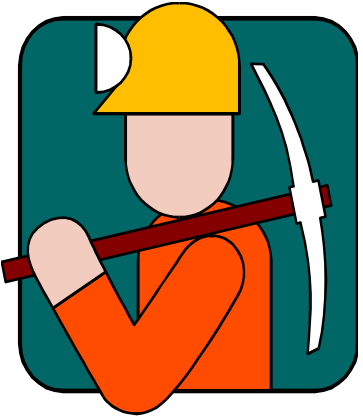


Histogram Analysis

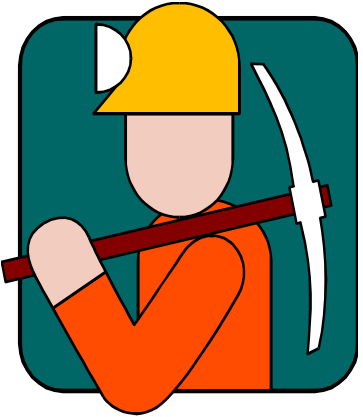


- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
 - Equal-width: equal bucket range
 - Equal-frequency (or equal-depth)

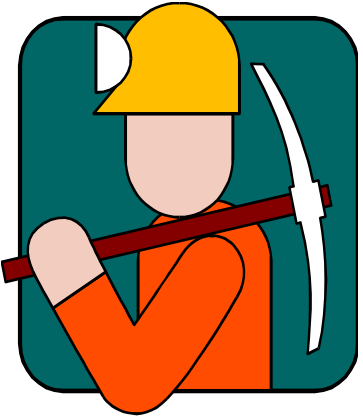




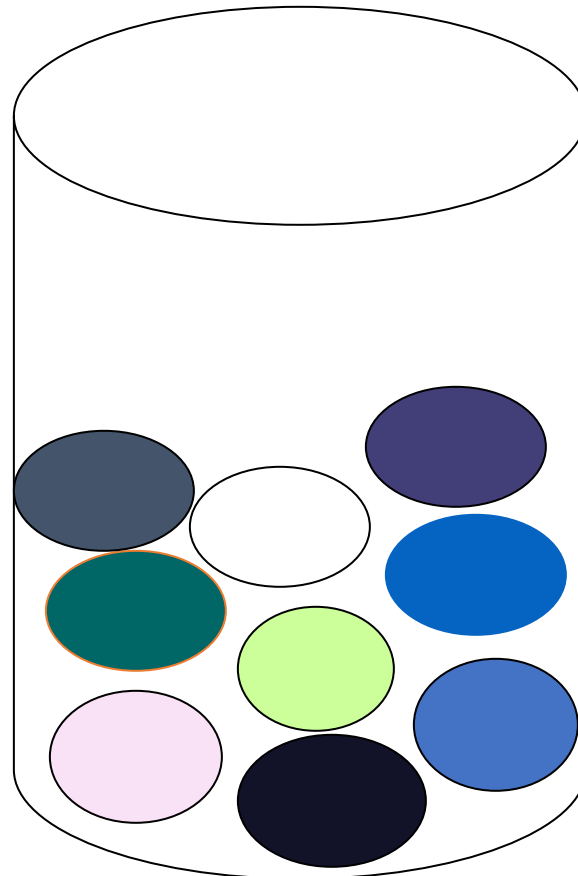
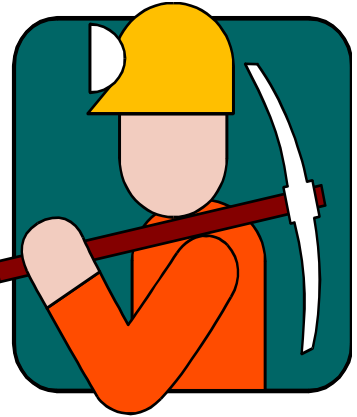
- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are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



- **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)

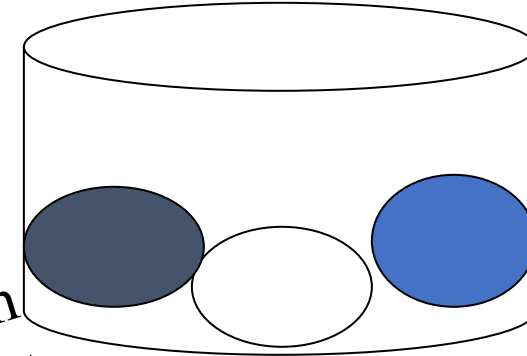


- **Simple random sampling**
 - There is an equal probability of selecting any particular item
- **Sampling without replacement**
 - Once an object is selected, it is removed from the population
- **Sampling with replacement**
 - A selected object is not removed from the population
- **Stratified sampling:**
 - 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

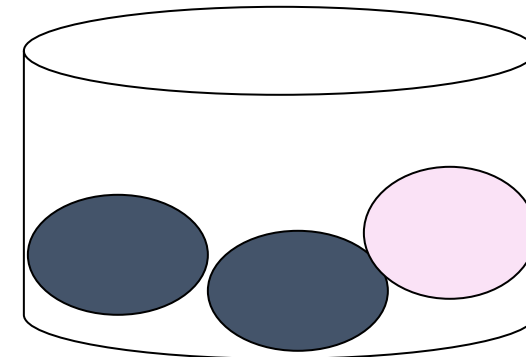


Raw Data

SRSWOR
(simple random
sample without
replacement)

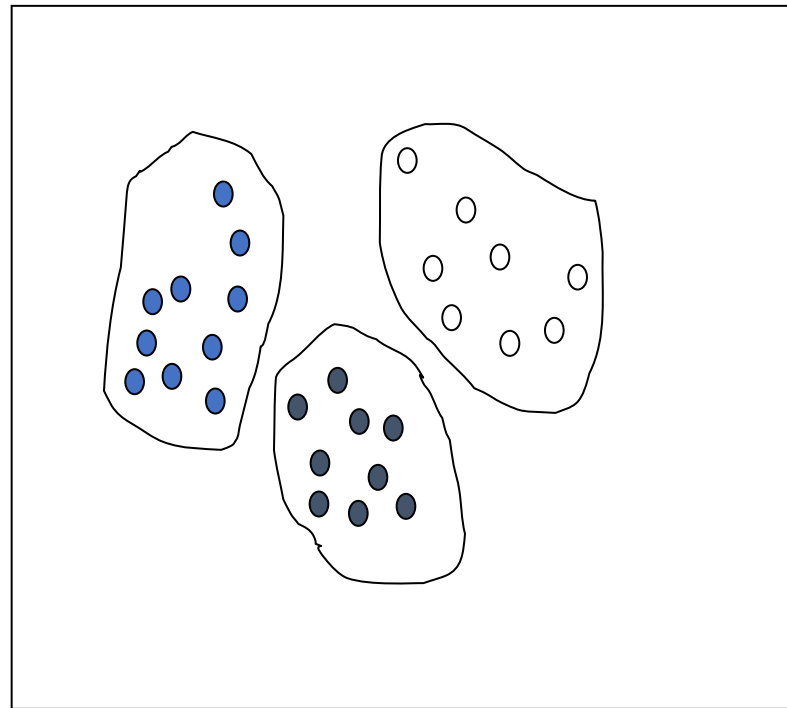


SRSWR

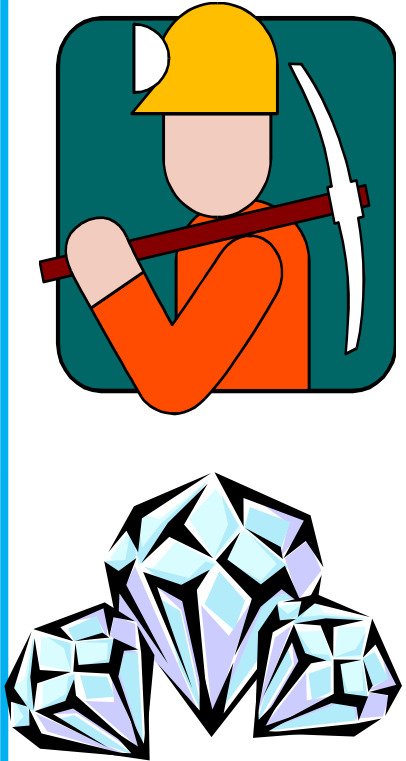
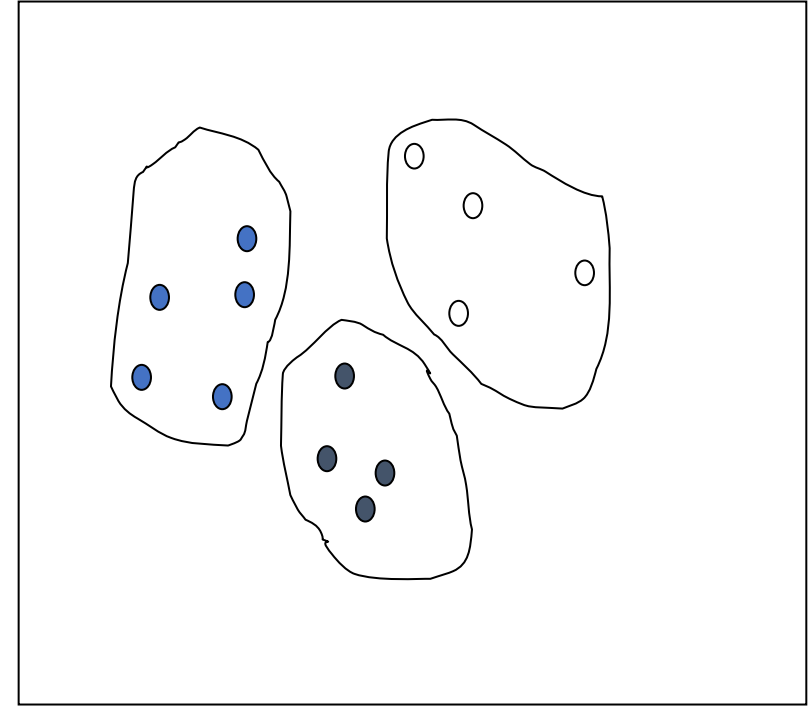


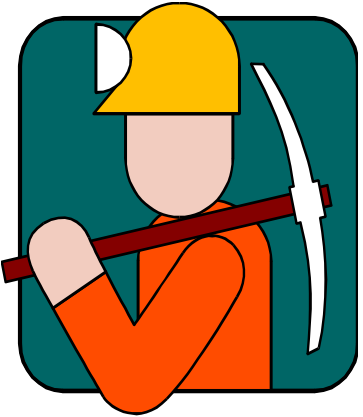


Raw Data

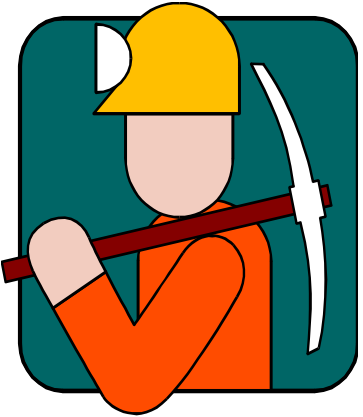


Cluster/Stratified Sample

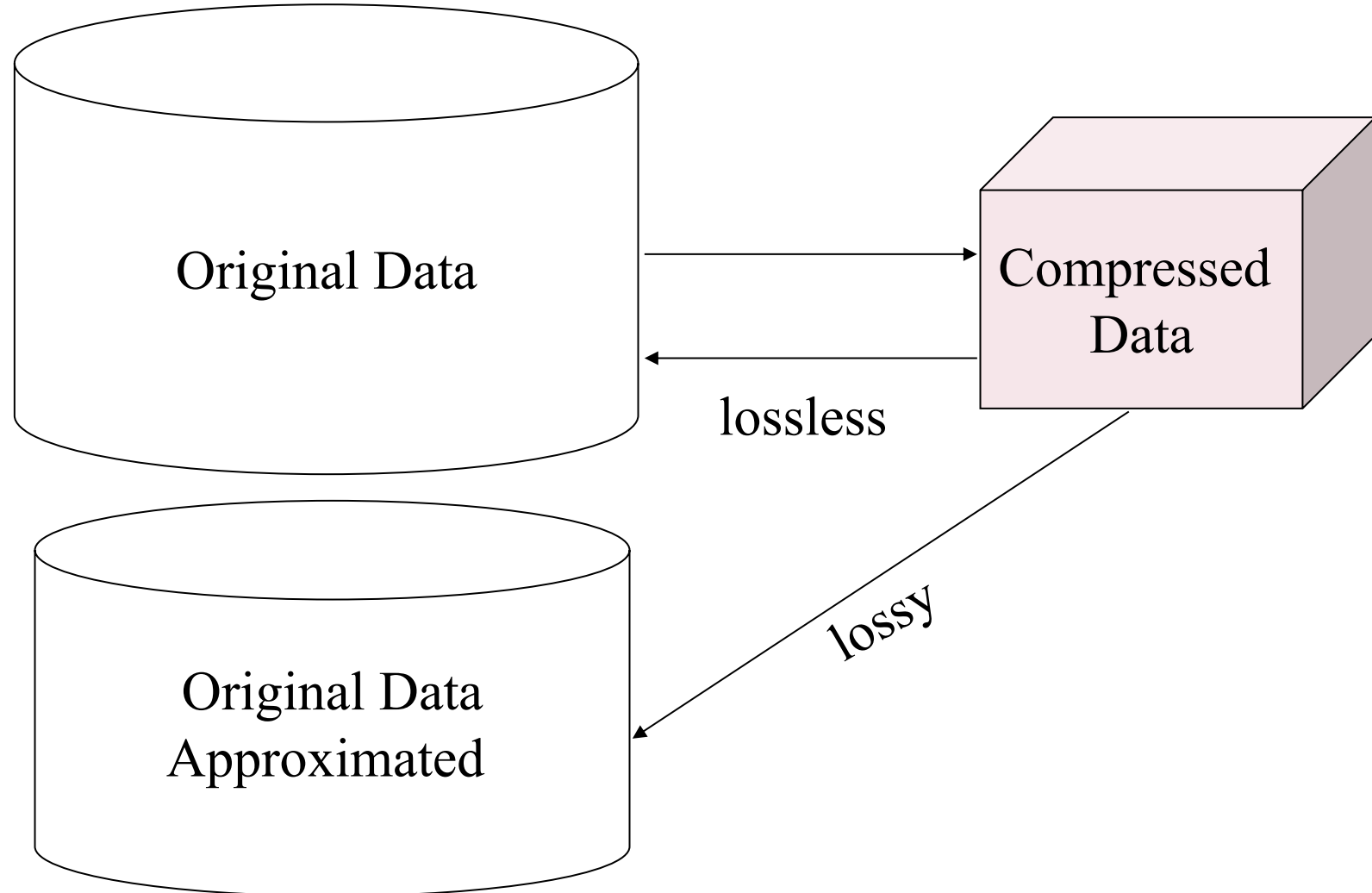
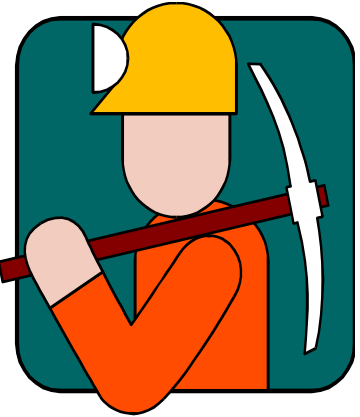




- The lowest level of a data cube (base cuboid)
 - The aggregated data for an **individual entity of interest**
 - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
 - Further reduce the size of data to deal with
- Reference appropriate levels
 - Use the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



- 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

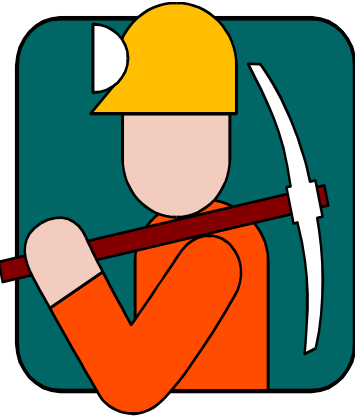




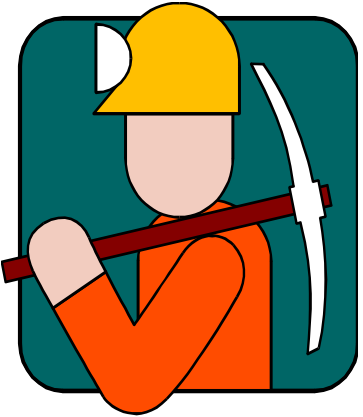
Data Preprocessing

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





- 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



- **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 \$73,600 is mapped to $\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$

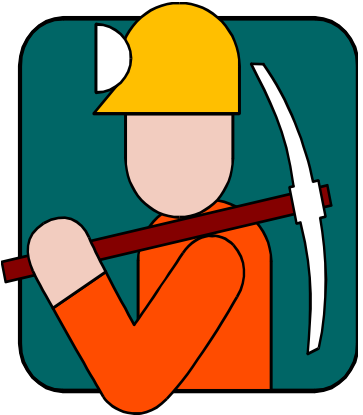
- **Z-score normalization** (μ : mean, σ : standard deviation):

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

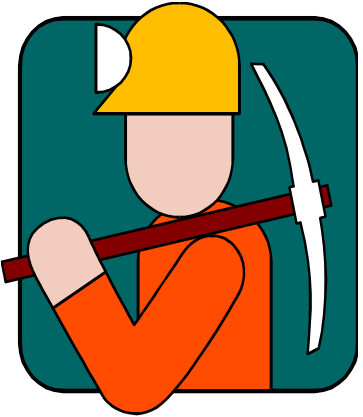
- Ex. Let $\mu = 54,000$, $\sigma = 16,000$. Then $\frac{73,600 - 54,000}{16,000} = 1.225$

- **Normalization by decimal scaling**

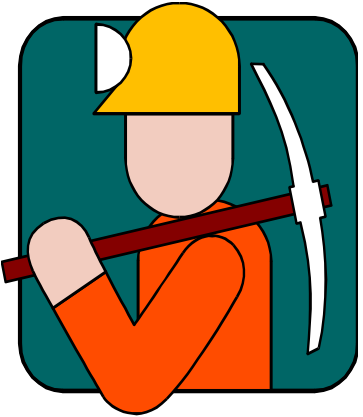
$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(|v'|) < 1$$



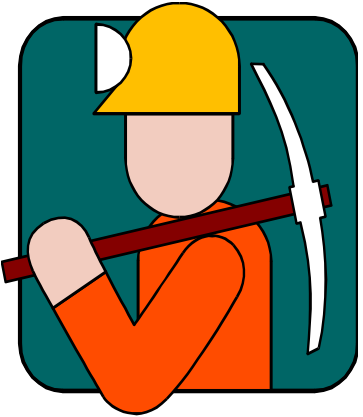
- Three types of attributes
 - Nominal—values from an unordered set, e.g., color, profession
 - Ordinal—values from an ordered set, e.g., military or academic rank
 - Numeric—real numbers, e.g., integer or real numbers
- **Discretization:** Divide the range of a continuous attribute into intervals
 - 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



- 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., χ^2) analysis (unsupervised, bottom-up merge)



- **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



❑ 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**:

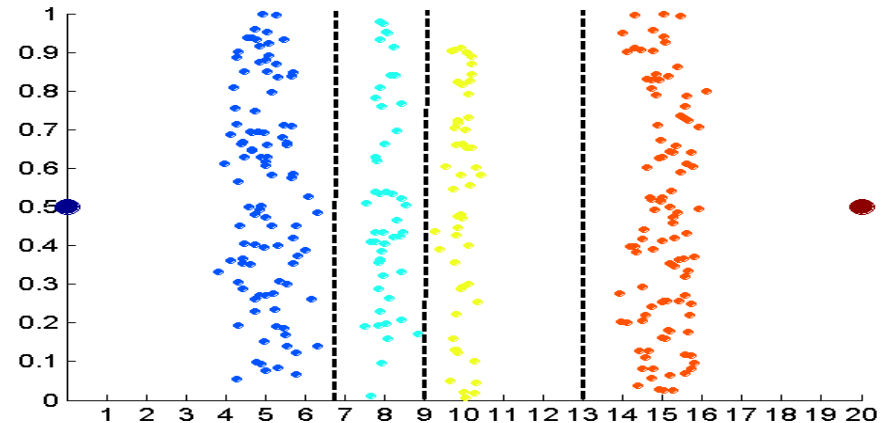
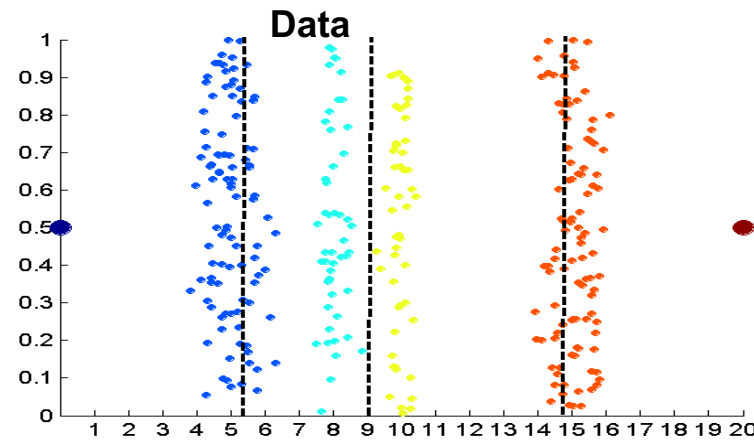
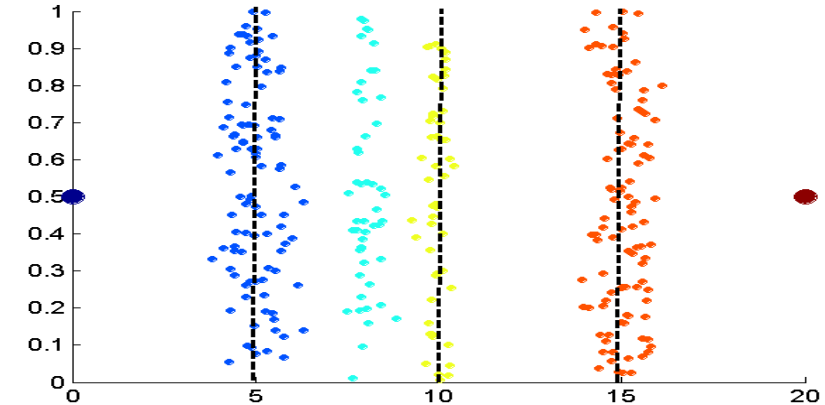
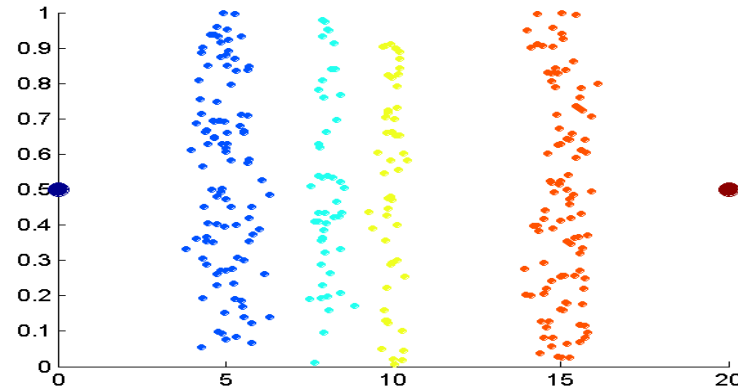
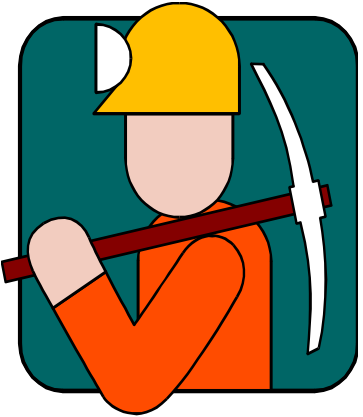
- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

* Smoothing by **bin boundaries**:

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

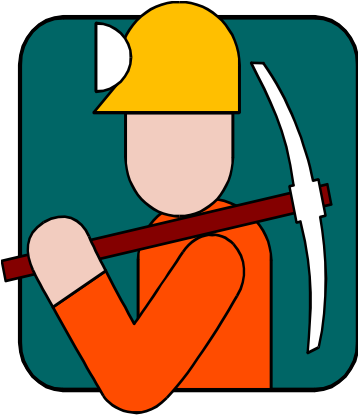


Discretization Without Using Class Labels (Binning vs. Clustering)

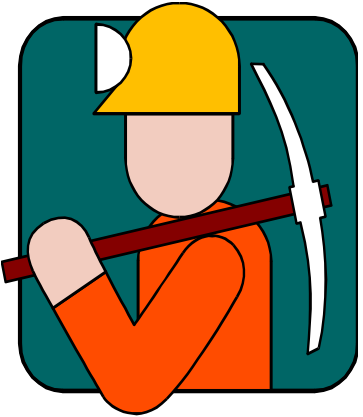


Equal frequency (binning)

K-means clustering leads to better results



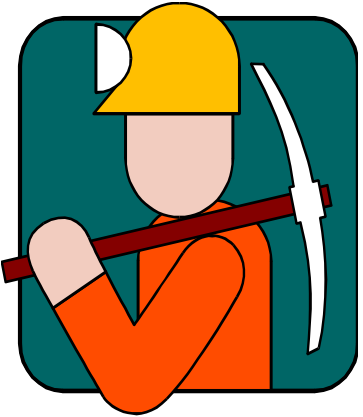
- **Classification** (e.g., decision tree analysis)
 - Supervised: Given class labels, e.g., cancerous vs. benign
 - Using *entropy* to determine split point (discretization point)
 - Top-down, recursive split
 - Details to be covered in Chapter 7
- **Correlation analysis** (e.g., Chi-merge: χ^2 -based discretization)
 - Supervised: use class information
 - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
 - Merge performed recursively, until a predefined stopping condition



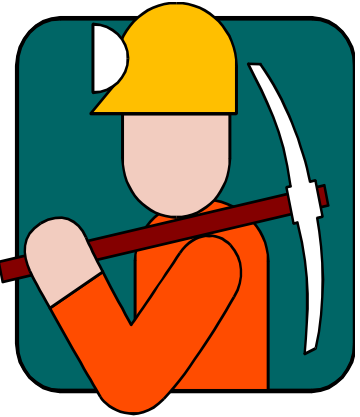
- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively 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.



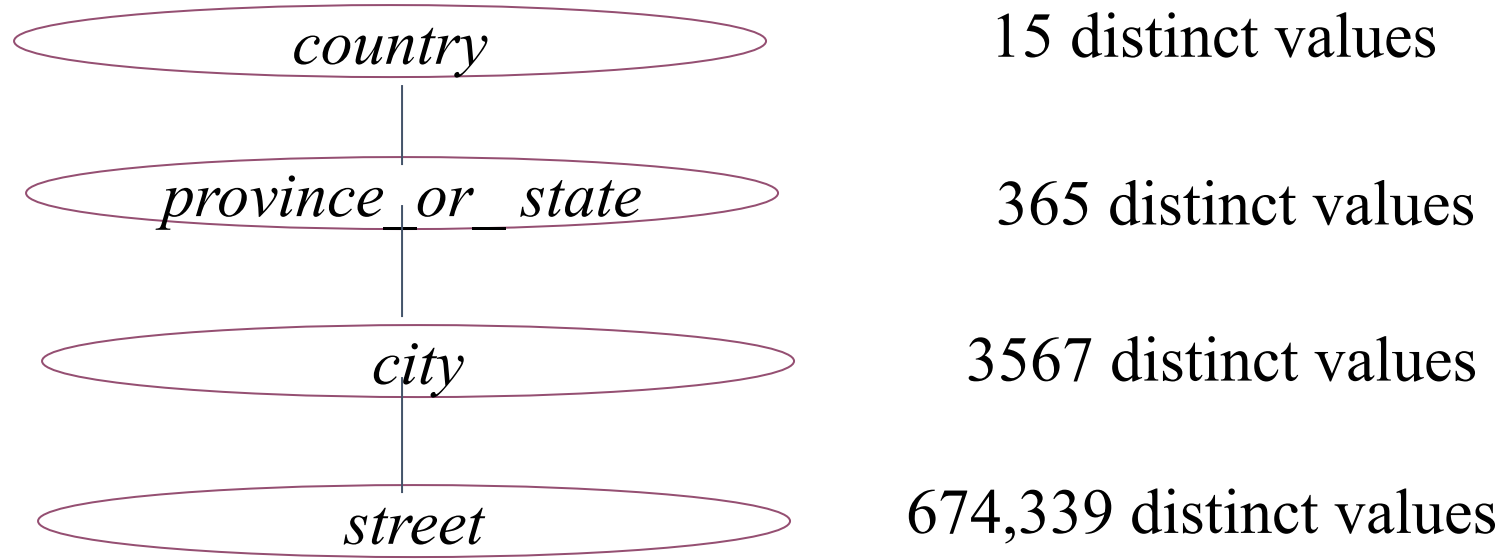
Concept Hierarchy Generation for Nominal Data



- 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: $\{street, city, state, country\}$



- 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

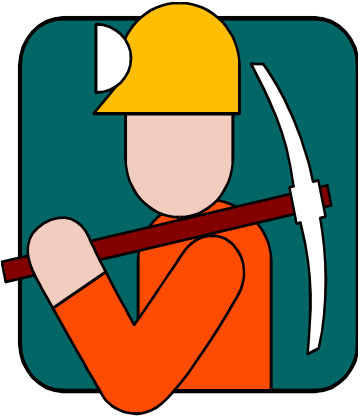




Data Preprocessing

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





- **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



Thank You

Data Preprocessing - Practice



Data Preprocessing - Practice

1 - Data Loading and Inspection

```
In [2]: import pandas as pd
```

```
In [3]: # Loading Data from a csv file  
ufo_data = pd.read_csv('http://bit.ly/uforeports')
```

```
In [3]: # reading tsv files with read_csv, but by using the sep parameter  
chips_data = pd.read_csv('http://bit.ly/chiporders', sep = '\t')
```

Data Preprocessing - Practice

1.1 Displaying Data Frames

```
In [5]: # This is to observe the first n rows of the data  
chips_data.head(5)
```

```
Out [5]:
```

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	\$2.39
1	1	1	Izze	[Clementine]	\$3.39
2	1	1	Nantucket Nectar	[Apple]	\$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	\$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	\$16.98

```
In [6]: # This is to observe the last n rows of the data  
chips_data.tail(5)
```

```
Out [6]:
```

	order_id	quantity	item_name	choice_description	item_price
4617	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour ...	\$11.75
4618	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...	\$11.75
4619	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	\$11.25
4620	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettu...	\$8.75
4621	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	\$8.75

```
In [7]: # This is useful for exploring diverse parts of the dataset.  
chips_data.sample(5)
```

```
Out [7]:
```

	order_id	quantity	item_name	choice_description	item_price
829	343	1	Barbacoa Crispy Tacos	[Roasted Chili Corn Salsa, [Fajita Vegetables,...	\$9.25
3421	1373	1	Carnitas Burrito	[Fresh Tomato Salsa, [Rice, Pinto Beans, Chees...	\$9.25
3315	1329	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Rice, Lettuce]]	\$8.75
1329	543	1	Chicken Bowl	[Tomatillo Red Chili Salsa, [Black Beans, Chee...	\$8.75
2510	997	2	Chicken Bowl	[Roasted Chili Corn Salsa, [Rice, Black Beans,...	\$22.50

Data Preprocessing - Practice

1.2 Data Inspection and Exploration

shape of the data

```
In [8]: chips_data.shape
```

```
Out[8]: (4622, 5)
```

Info of the Data

```
In [9]: chips_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4622 entries, 0 to 4621
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   order_id              4622 non-null   int64
1   quantity              4622 non-null   int64
2   item_name             4622 non-null   object
3   choice_description     3376 non-null   object
4   item_price            4622 non-null   object
dtypes: int64(2), object(3)
memory usage: 180.7+ KB
```

Data Preprocessing - Practice

Basic Statistics of the data

```
In [10]: chips_data.describe()
```

```
Out[10]:
```

	order_id	quantity
count	4622.000000	4622.000000
mean	927.254868	1.075725
std	528.890796	0.410186
min	1.000000	1.000000
25%	477.250000	1.000000
50%	926.000000	1.000000
75%	1393.000000	1.000000
max	1834.000000	15.000000

```
In [13]:  #(include='object') will describe about the object data.  
 #Similarly, you can use (include='all') if youwant to see all at once.  
 #chips_data.describe(include = 'object')  
chips_data.describe(include = 'object')
```

```
Out[13]:
```

	item_name	choice_description	item_price
count	4622	3376	4622
unique	50	1043	78
top	Chicken Bowl	[Diet Coke]	\$8.75
freq	726	134	730

```
In [12]: chips_data.describe(include = 'all')
```

```
Out[12]:
```

	order_id	quantity	item_name	choice_description	item_price
count	4622.000000	4622.000000	4622	3376	4622
unique	NaN	NaN	50	1043	78
top	NaN	NaN	Chicken Bowl	[Diet Coke]	\$8.75
freq	NaN	NaN	726	134	730
mean	927.254868	1.075725	NaN	NaN	NaN
std	528.890796	0.410186	NaN	NaN	NaN
min	1.000000	1.000000	NaN	NaN	NaN
25%	477.250000	1.000000	NaN	NaN	NaN
50%	926.000000	1.000000	NaN	NaN	NaN
75%	1393.000000	1.000000	NaN	NaN	NaN
max	1834.000000	15.000000	NaN	NaN	NaN

Data Preprocessing - Practice

Accessing Columns

```
In [14]: chips_data.item_name
```

```
Out[14]: 0          Chips and Fresh Tomato Salsa
          1                      Izze
          2          Nantucket Nectar
          3    Chips and Tomatillo-Green Chili Salsa
          4          Chicken Bowl
          ...
          4617         Steak Burrito
          4618         Steak Burrito
          4619         Chicken Salad Bowl
          4620         Chicken Salad Bowl
          4621         Chicken Salad Bowl
          Name: item_name, Length: 4622, dtype: object
```

```
In [15]: # Accessing a single column
          chips_data['item_name']
```

```
Out[15]: 0          Chips and Fresh Tomato Salsa
          1                      Izze
          2          Nantucket Nectar
          3    Chips and Tomatillo-Green Chili Salsa
          4          Chicken Bowl
          ...
          4617         Steak Burrito
          4618         Steak Burrito
          4619         Chicken Salad Bowl
          4620         Chicken Salad Bowl
          4621         Chicken Salad Bowl
          Name: item_name, Length: 4622, dtype: object
```

Data Preprocessing - Practice

Finding no.of unique values from each column

```
In [16]: chips_data.nunique()
```

```
Out[16]: order_id          1834
         quantity          9
         item_name         50
         choice_description 1043
         item_price         78
         dtype: int64
```

Finding the count of unique values from each column

```
In [17]: # As i only want to see top 5, gave head(5)
         chips_data['item_name'].value_counts().head(5)
```

```
Out[17]: Chicken Bowl          726
         Chicken Burrito       553
         Chips and Guacamole   479
         Steak Burrito         368
         Canned Soft Drink     301
         Name: item_name, dtype: int64
```

Finding stats for a specific column ¶

```
In [18]: chips_data.item_name.mode()
```

```
Out[18]: 0    Chicken Bowl
         Name: item_name, dtype: object
```

```
In [19]: chips_data.quantity.mean()
```

```
Out[19]: 1.0757247944612722
```

Data Preprocessing - Practice

3 - Data Cleaning

3.1 Handling Missing Data

```
In [35]: chips_data.choice_description.isna()
```

```
Out [35]: 0      True
1     False
2     False
3      True
4     False
...
4617  False
4619  False
4620  False
4621  False
Name: choice_description, Length: 4622, dtype: bool
```

click to expand output; double click to hide output

```
In [36]: #Applying the .isna() method for a column
#will return the boolean list with True for the indices where there is a missing value.
#And passing this list to a dataframe
#will return the rows where that column values are null.
```

```
In [37]: # Rows where the choice_description column value is missing.
chips_data[chips_data.choice_description.isna()].head(5)
```

```
Out [37]:
```

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	\$2.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	\$2.39
6	3	1	Side of Chips	NaN	\$1.69
10	5	1	Chips and Guacamole	NaN	\$4.45
14	7	1	Chips and Guacamole	NaN	\$4.45

Data Preprocessing - Practice

To Find the count of missing values in each column

```
In [38]: chips_data.isna().sum()
```

```
Out[38]: order_id          0
         quantity         0
         item_name         0
         choice_description 1246
         item_price         0
         dtype: int64
```

```
In [39]: # Similarly if you want to check the missing data count for one column
         chips_data.choice_description.isna().sum()
```

```
Out[39]: 1246
```

Data Preprocessing - Practice

Filling the missing Data using fillna

```
In [40]: ufo_data.isna().sum()
```

```
Out[40]: City                25  
Colors Reported          15359  
Shape Reported           2644  
State                     0  
Time                      0  
dtype: int64
```

```
In [41]: #As for Categorical data we can fill the data with a value if we have any,  
#if not we can prefer the mode.  
#Mean imputation is often used when the missing values are numerical and  
#the distribution of the variable is approximately normal.  
#Median imputation is preferred when the distribution is skewed,  
#as the median is less sensitive to outliers than the mean
```

```
In [43]: most_repeated=ufo_data['Shape Reported'].mode()[0]  
most_repeated
```

```
Out[43]: 'LIGHT'
```

```
In [44]: # using inplace=True will automatically save the data, Here we are filling the missing values  
ufo_data['Shape Reported'].fillna(most_repeated,inplace=True)
```

```
In [45]: ufo_data.isna().sum()
```

```
Out[45]: City                25  
Colors Reported          15359  
Shape Reported           0  
State                     0  
Time                      0  
dtype: int64
```

Data Preprocessing - Practice

Drop the duplicated rows using .drop_duplicates

```
In [11]: chips_data.duplicated().sum()
```

```
Out[11]: 59
```

```
In [12]: # As we know there are only 59 duplicated rows and we choose to drop them, then w  
chips_data.drop_duplicates().head(3)
```

```
Out[12]:
```

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	\$2.39
1	1	1	Izze	[Clementine]	\$3.39
2	1	1	Nantucket Nectar	[Apple]	\$3.39

```
In [13]: # It is advisable to check the shape after dropping before saving it  
chips_data.drop_duplicates().shape
```

```
Out[13]: (4563, 5)
```

```
In [14]: # If you are satisfied with the resulting shape, you can save it.  
chips_data.drop_duplicates(inplace=True)
```