Data Mining and Machine Learning

Bioinspired computational methods

Biological data mining

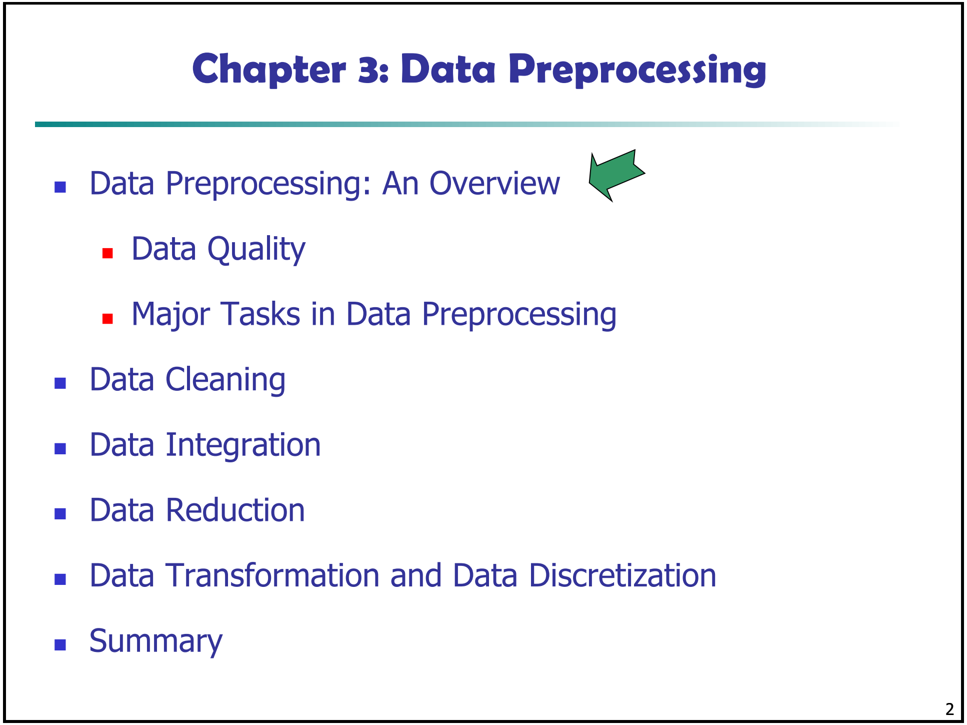
Data Prepocessing

Francesco Marcelloni

Department of Information Engineering
University of Pisa
ITALY

Some slides belong to the collection

Jiawei Han, Micheline Kamber, and Jian Pei
University of Illinois at Urbana-Champaign
Simon Fraser University

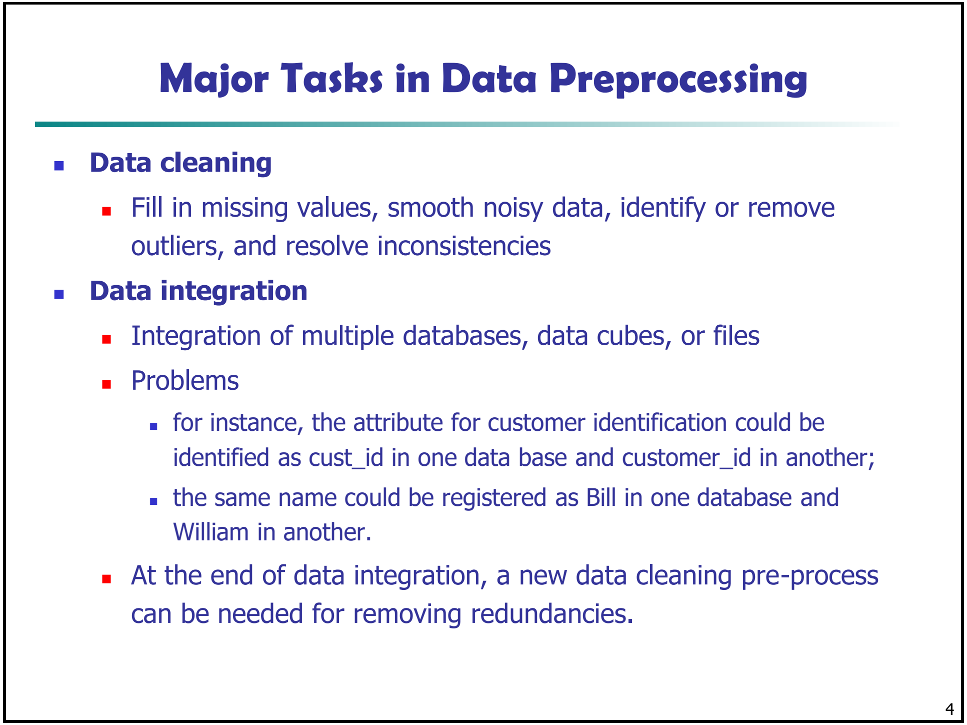


Chapter 3: Data Preprocessing

- 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: 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?

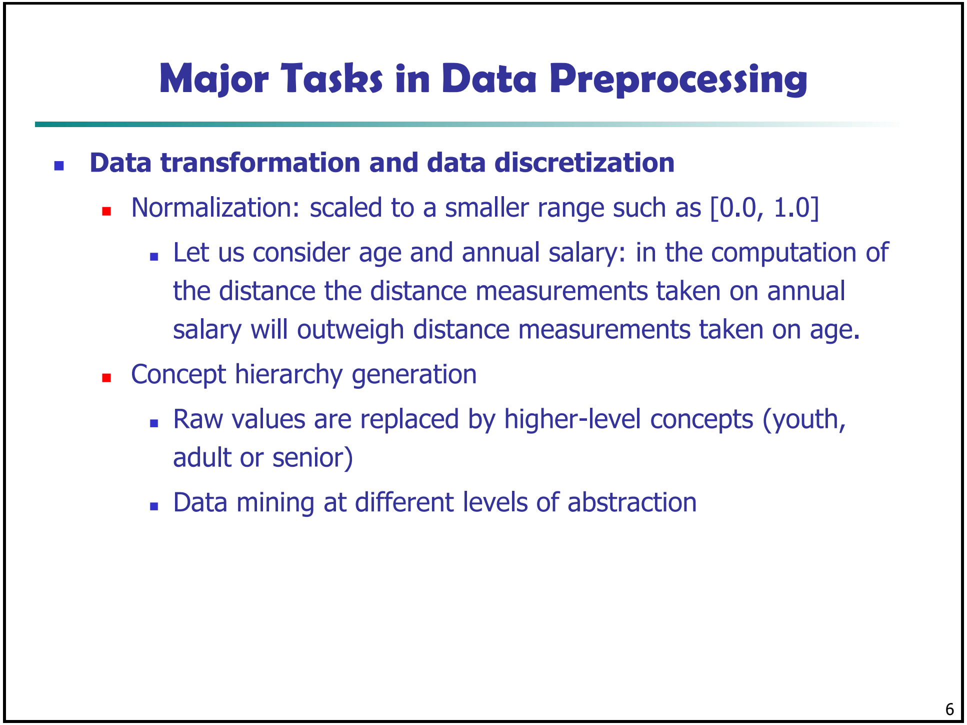


Major Tasks in Data Preprocessing

- **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
  - Problems
    - for instance, the attribute for customer identification could be identified as cust_id in one data base and customer_id in another;
    - the same name could be registered as Bill in one database and William in another.
  - At the end of data integration, a new data cleaning pre-process can be needed for removing redundancies.

Major Tasks in Data Preprocessing

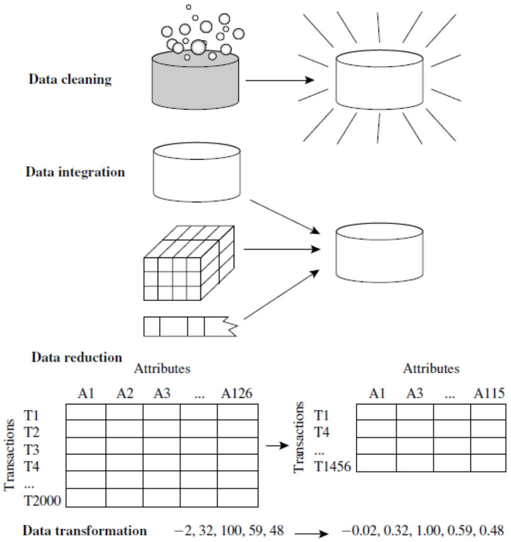
- **Data reduction**
  - **Is there any way to reduce the size of data set without jeopardizing the data mining results?**
  - *Dimensionality reduction*: data encoding schemes are applied to obtain a compressed representation of the original data
    - Compression techniques
    - Attribute subset selection
    - Attribute construction
  - *Numerosity reduction*: smaller representations using parametric models (regression models) or nonparametric models (cluster, sampling, ecc.)
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

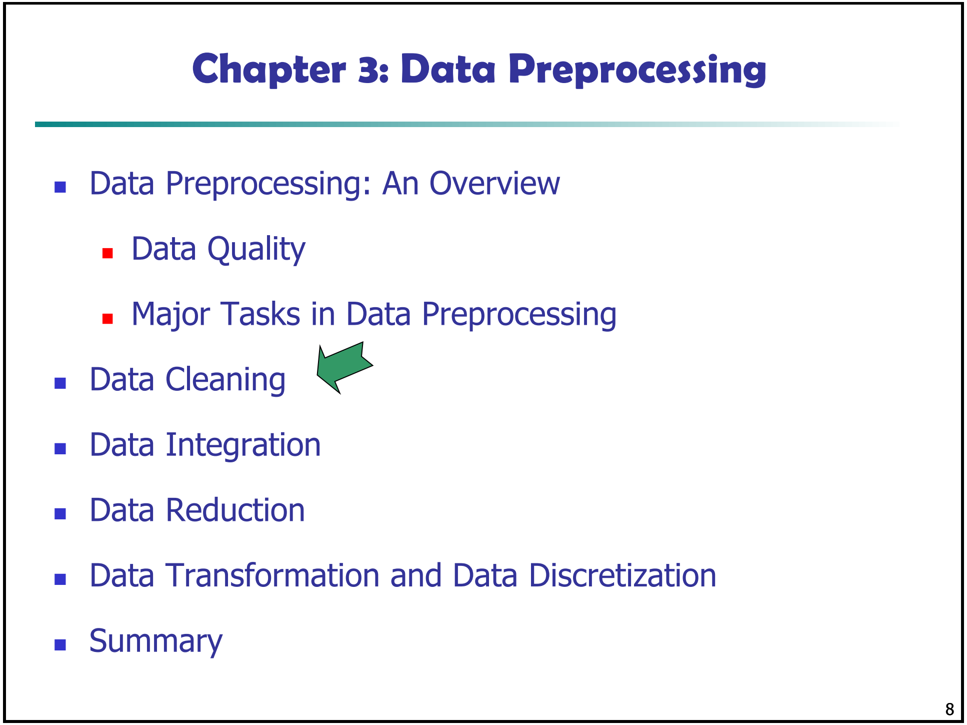


Major Tasks in Data Preprocessing

- **Data transformation and data discretization**
  - Normalization: scaled to a smaller range such as [0.0, 1.0]
    - Let us consider age and annual salary: in the computation of the distance the distance measurements taken on annual salary will outweigh distance measurements taken on age.
  - Concept hierarchy generation
    - Raw values are replaced by higher-level concepts (youth, adult or senior)
    - Data mining at different levels of abstraction

Major Tasks in Data Preprocessing



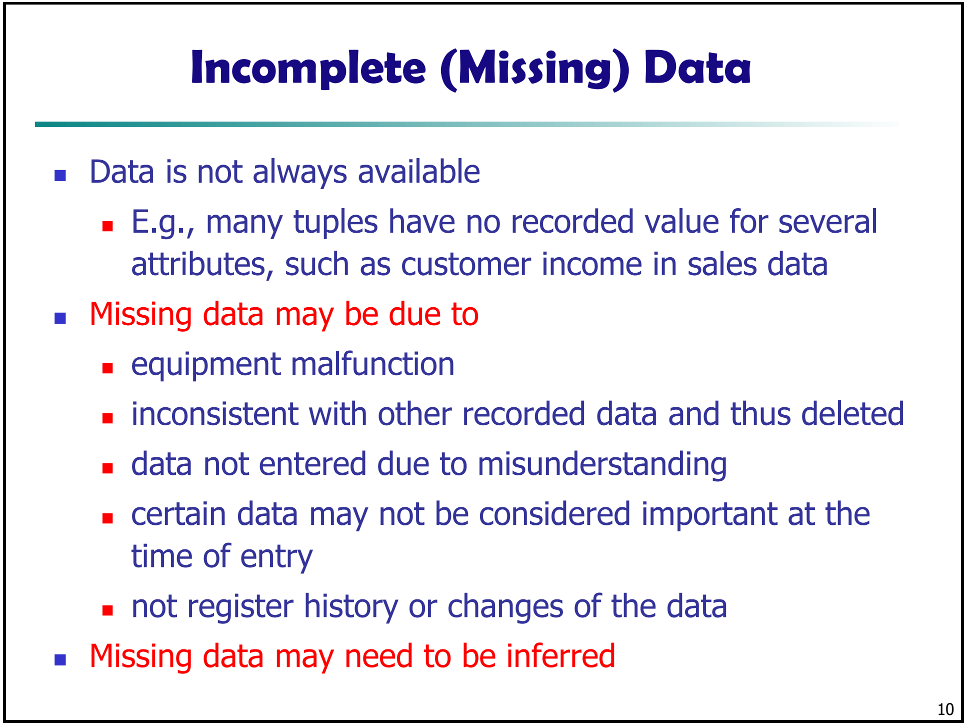


Chapter 3: Data Preprocessing

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

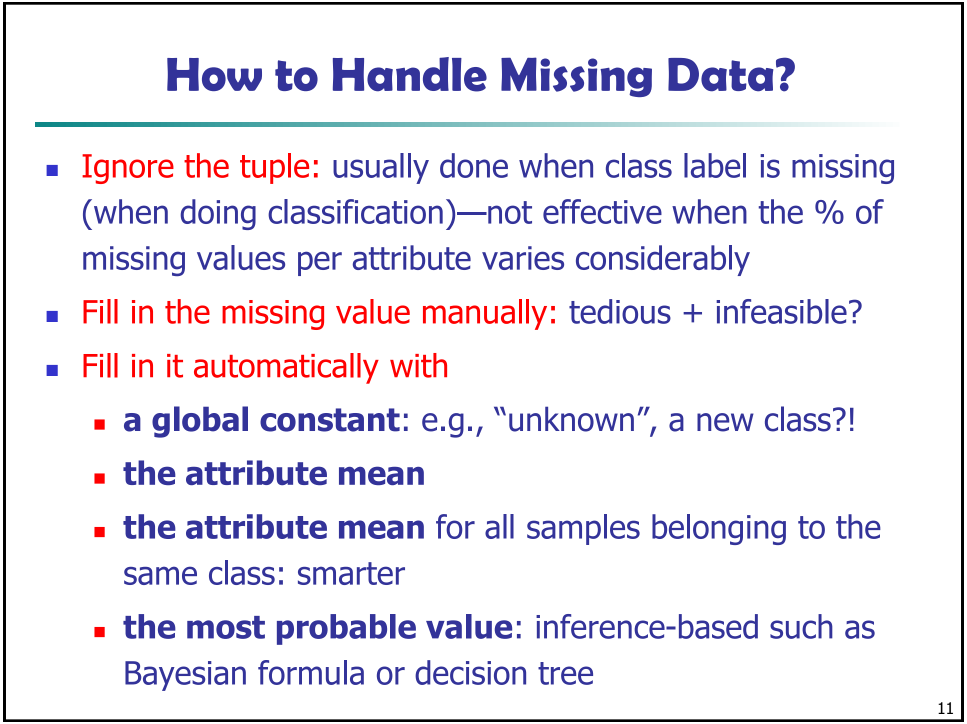
Data Cleaning

- 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?



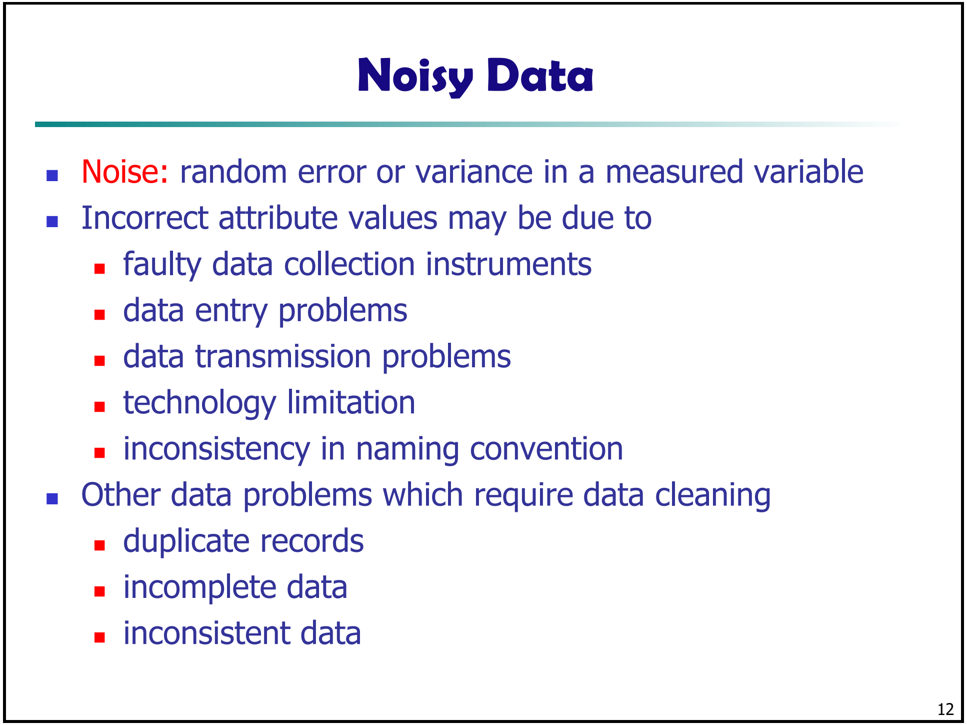
Incomplete (Missing) Data

- 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 register history or changes of the data
- Missing data may need to be inferred



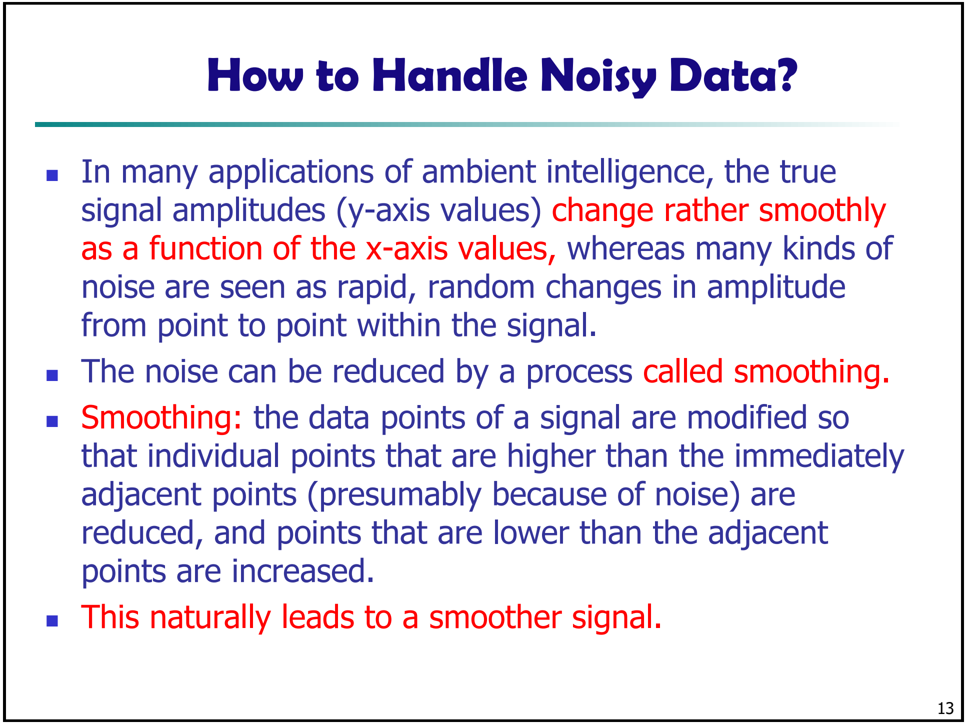
How to Handle Missing Data?

- 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



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?

- In many applications of ambient intelligence, the true signal amplitudes (y-axis values) change rather smoothly as a function of the x-axis values, whereas many kinds of noise are seen as rapid, random changes in amplitude from point to point within the signal.
- The noise can be reduced by a process called smoothing.
- Smoothing: the data points of a signal are modified so that individual points that are higher than the immediately adjacent points (presumably because of noise) are reduced, and points that are lower than the adjacent points are increased.
- This naturally leads to a smoother signal.

Smoothing Algorithms

Rectangular or unweighted sliding-average smooth

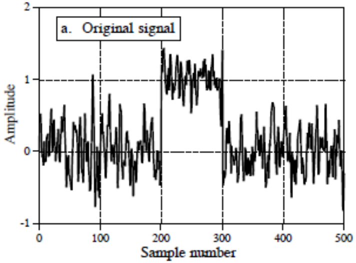
- The simplest smoothing algorithm
- It simply replaces each point in the signal with the average of m adjacent points, where m is a positive integer called the smooth width. For example, for a 3-point smooth (m = 3):

$$S_j = \frac{Y_{j-1} + Y_j + Y_{j+1}}{3}$$

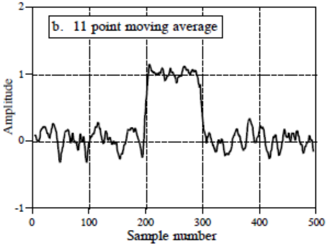
for j = 2 to n-1, where S_j the jth point in the smoothed signal, Y_j the jth point in the original signal, and n is the total number of points in the signal.

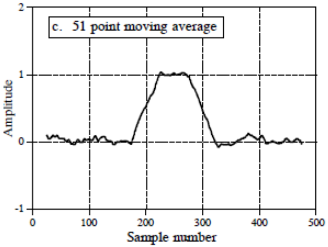
Smoothing Algorithms

Rectangular or unweighted sliding-average smooth



Filtered with 11 and 51 point moving average filters





Smoothing Algorithms

Rectangular or unweighted sliding-average smooth

- If the underlying function is **constant**, or is **changing linearly with time** (increasing or decreasing), then no bias is introduced into the result.
- A bias is introduced, however, if the underlying function has a **nonzero second derivative**. At a local maximum, for example, moving window averaging always reduces the function value.

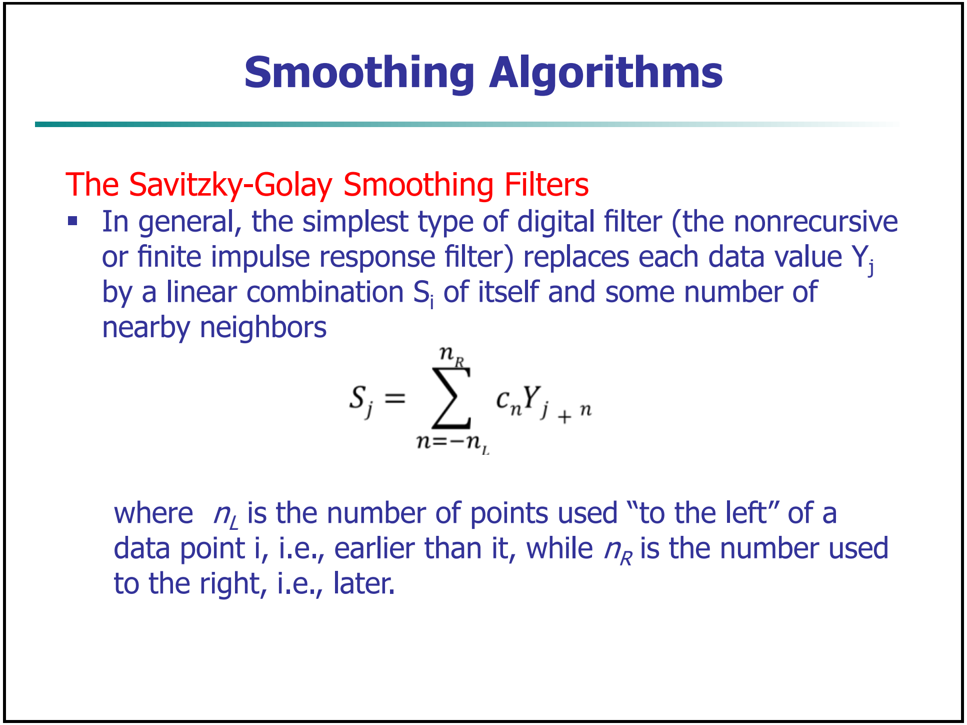
Smoothing Algorithms

Triangular smooth

- Implements a weighted smoothing function. For example, for a 5-point smooth (m = 5):

$$S_j = \frac{Y_{j-2} + 2Y_{j-1} + 3Y_j + 2Y_{j+1} + Y_{j+2}}{9}$$

  for j = 3 to n-2. This smooth is more effective at reducing high-frequency noise in the signal than the simpler rectangular smooth.
- The 5-point triangular smooth above **is equivalent to two passes of a 3-point rectangular smooth**
- The width of the smooth *m* is an odd integer and the smooth coefficients are symmetrically balanced around the central point, which is important point because it preserves the x-axis position of peaks and other features in the signal.



Smoothing Algorithms

The Savitzky-Golay Smoothing Filters

- In general, the simplest type of digital filter (the nonrecursive or finite impulse response filter) replaces each data value Y_j by a linear combination S_i of itself and some number of nearby neighbors

$$S_j = \sum_{n=-n_L}^{n_R} c_n Y_{j+n}$$

where n_L is the number of points used “to the left” of a data point i, i.e., earlier than it, while n_R is the number used to the right, i.e., later.

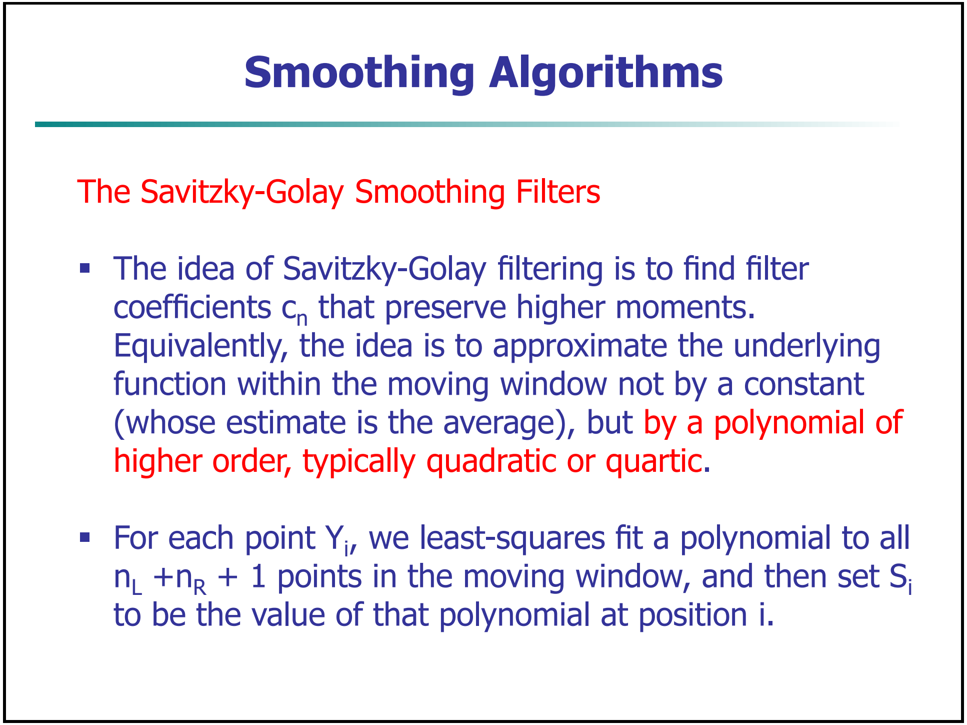
Smoothing Algorithms

The Savitzky-Golay Smoothing Filters

- In the unweighted sliding-average smooth

$$c_n = \frac{1}{(n_L + n_R + 1)}$$

where n_L is the number of points used “to the left” of a data point i, i.e., earlier than it, while n_R is the number used to the right, i.e., later.



Smoothing Algorithms

The Savitzky-Golay Smoothing Filters

- The idea of Savitzky-Golay filtering is to find filter coefficients c_n that preserve higher moments. Equivalently, the idea is to approximate the underlying function within the moving window not by a constant (whose estimate is the average), but by a polynomial of higher order, typically quadratic or quartic.
- For each point Y_i, we least-squares fit a polynomial to all $n_L + n_R + 1$ points in the moving window, and then set S_i to be the value of that polynomial at position i.

Smoothing Algorithms

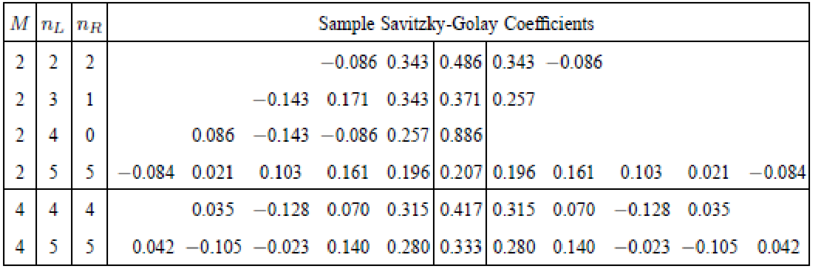
The Savitzky-Golay Smoothing Filters

- All these least-squares fits would be laborious if done as described.
- Luckily, since the process of least-squares fitting involves only a linear matrix inversion, the coefficients of a fitted polynomial are themselves linear in the values of the data.
- Thus, we can do all the fitting in advance, for fictitious data consisting of all zeros except for a single 1, and then do the fits on the real data just by taking linear combinations.

Smoothing Algorithms

The Savitzky-Golay Smoothing Filters

Some example

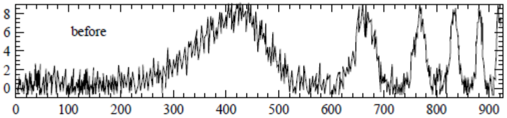


M	n_L	n_R	Sample Savitzky-Golay Coefficients										
2	2	2				−0.086	0.343	0.486	0.343	−0.086			
2	3	1			−0.143	0.171	0.343	0.371	0.257				
2	4	0		0.086	−0.143	−0.086	0.257	0.886					
2	5	5	−0.084	0.021	0.103	0.161	0.196	0.207	0.196	0.161	0.103	0.021	−0.084
4	4	4		0.035	−0.128	0.070	0.315	0.417	0.315	0.070	−0.128	0.035	
4	5	5	0.042	−0.105	−0.023	0.140	0.280	0.333	0.280	0.140	−0.023	−0.105	0.042

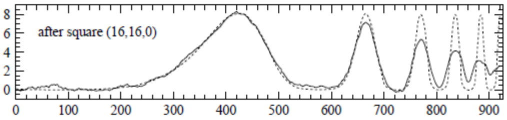
Smoothing Algorithms

The Savitzky-Golay Smoothing Filters

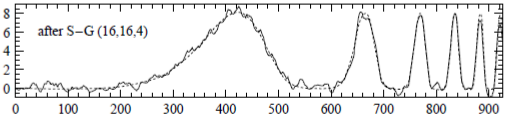
Some example

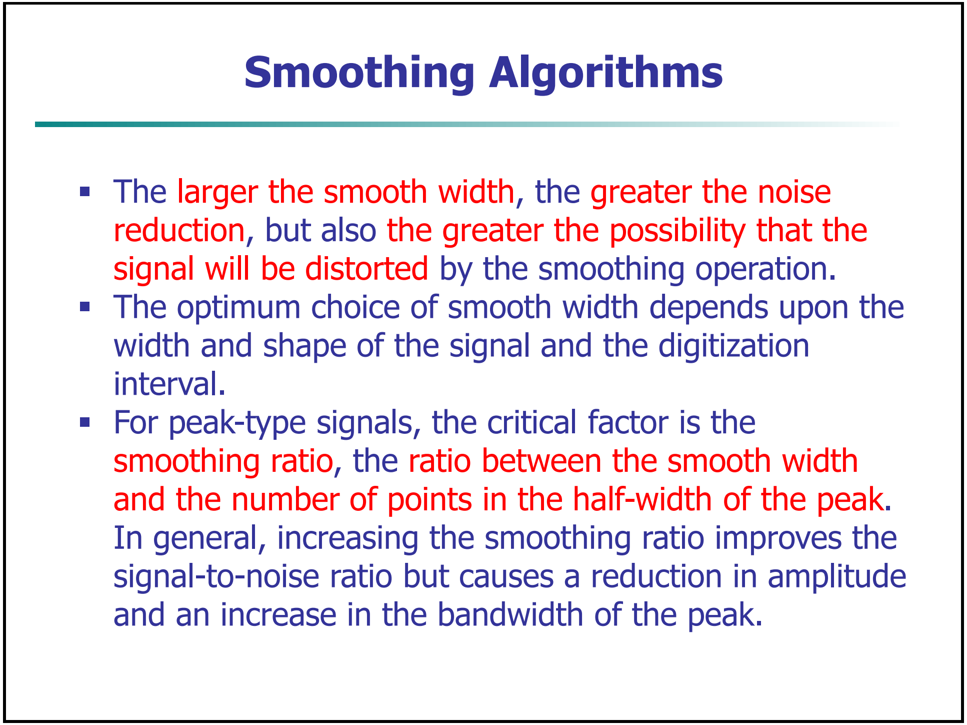


Sliding-average smooth with window of 33 points



Savitzky-Golay of degree 4 with window of 33 points

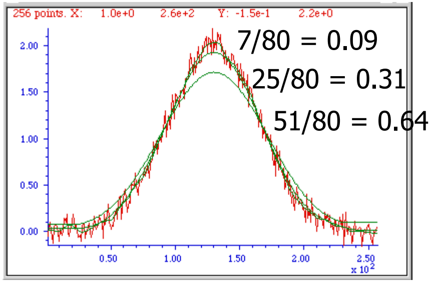




Smoothing Algorithms

- The larger the smooth width, the greater the noise reduction, but also the greater the possibility that the signal will be distorted by the smoothing operation.
- The optimum choice of smooth width depends upon the width and shape of the signal and the digitization interval.
- For peak-type signals, the critical factor is the smoothing ratio, the ratio between the smooth width and the number of points in the half-width of the peak. In general, increasing the smoothing ratio improves the signal-to-noise ratio but causes a reduction in amplitude and an increase in the bandwidth of the peak.

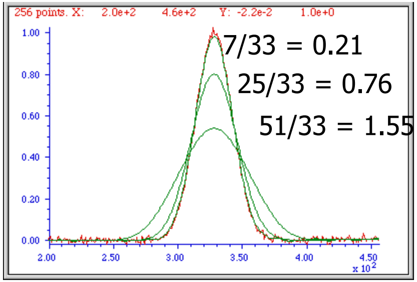
Smoothing Algorithms



7/80 = 0.09

25/80 = 0.31

51/80 = 0.64



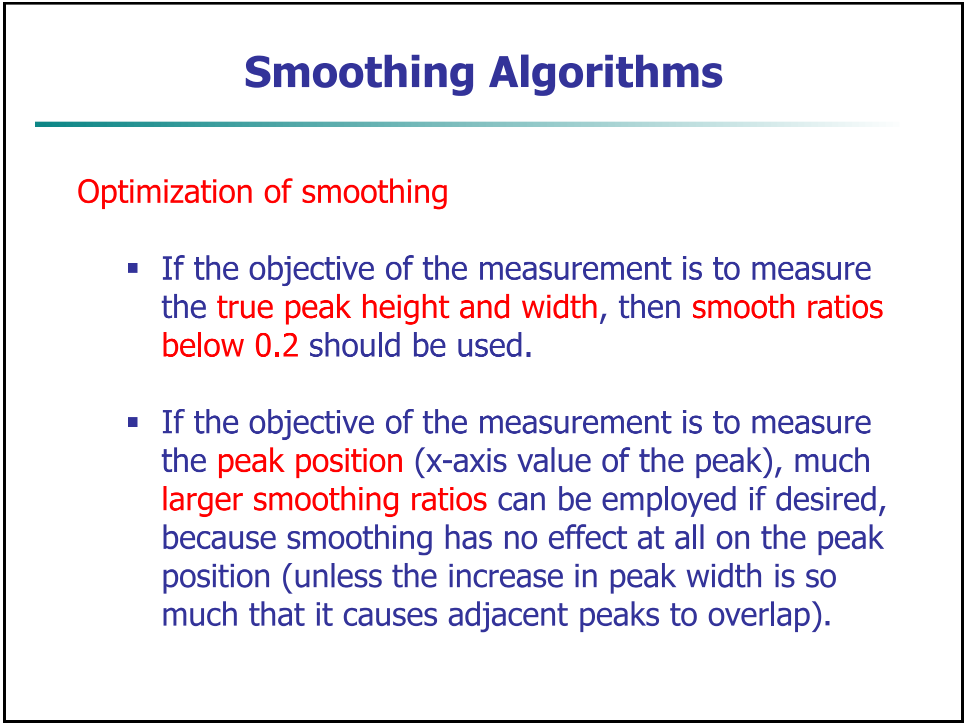
7/33 = 0.21

25/33 = 0.76

51/33 = 1.55

half-width of
the peak 80 pts

half-width of
the peak 33 pts



Smoothing Algorithms

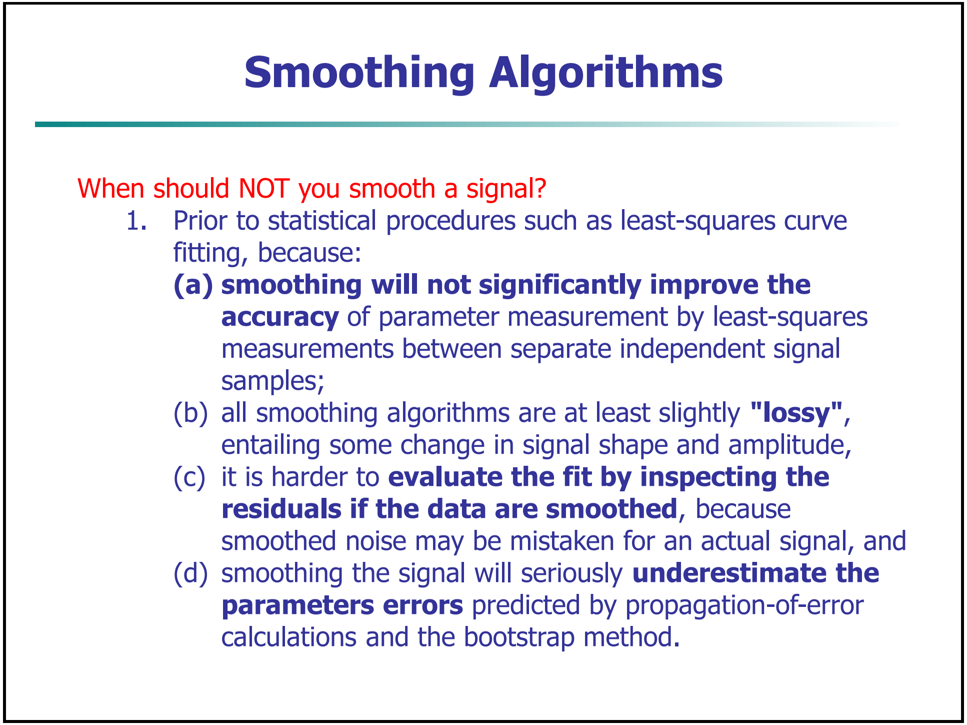
Optimization of smoothing

- If the objective of the measurement is to measure the true peak height and width, then smooth ratios below 0.2 should be used.
- If the objective of the measurement is to measure the peak position (x-axis value of the peak), much larger smoothing ratios can be employed if desired, because smoothing has no effect at all on the peak position (unless the increase in peak width is so much that it causes adjacent peaks to overlap).

Smoothing Algorithms

When should you smooth a signal?

1. for cosmetic reasons, to prepare a nicer-looking graphic of a signal for visual inspection or publication;
2. if the signal will be subsequently processed by an algorithm that would be adversely affected by the presence of too much high-frequency noise in the signal, for example if the location of maxima, minima, or inflection points in the signal is to be automatically determined by detecting zero-crossings in derivatives of the signal.



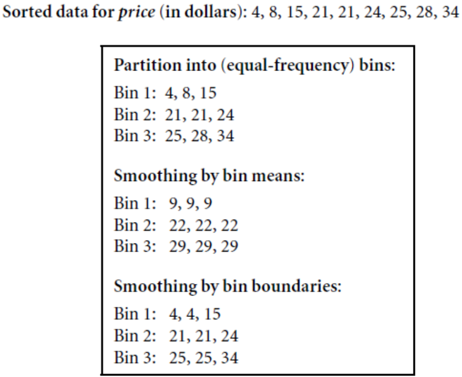
Smoothing Algorithms

When should NOT you smooth a signal?

1. Prior to statistical procedures such as least-squares curve fitting, because:
   (a) smoothing will not significantly improve the accuracy of parameter measurement by least-squares measurements between separate independent signal samples;
   (b) all smoothing algorithms are at least slightly **"lossy"**, entailing some change in signal shape and amplitude,
   (c) it is harder to **evaluate the fit by inspecting the residuals if the data are smoothed**, because smoothed noise may be mistaken for an actual signal, and
   (d) smoothing the signal will seriously **underestimate the parameters errors** predicted by propagation-of-error calculations and the bootstrap method.

How to Handle Noisy Data?

- Binning: smooth a sorted data value by consulting its neighborhood
  - 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.



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

Partition into (equal-frequency) bins:

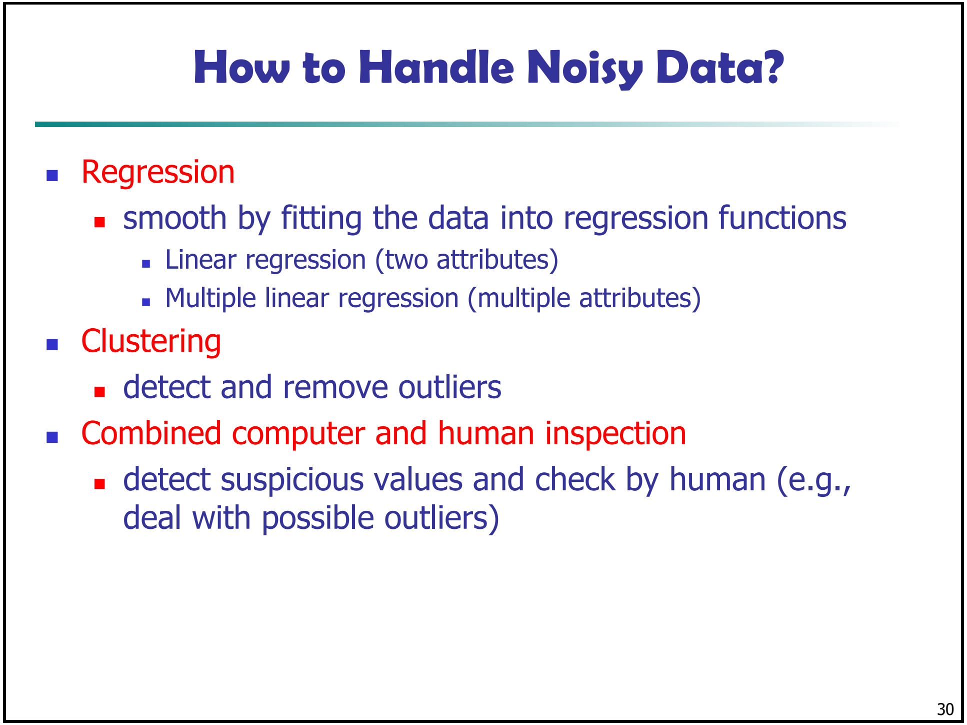
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:

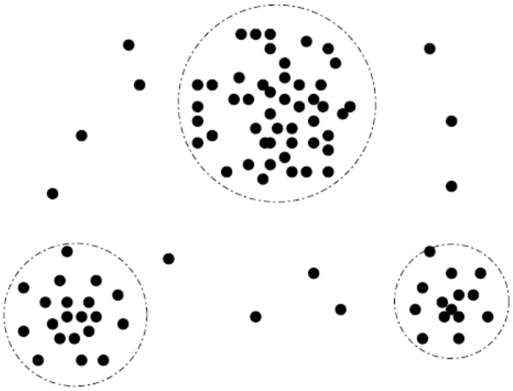
Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

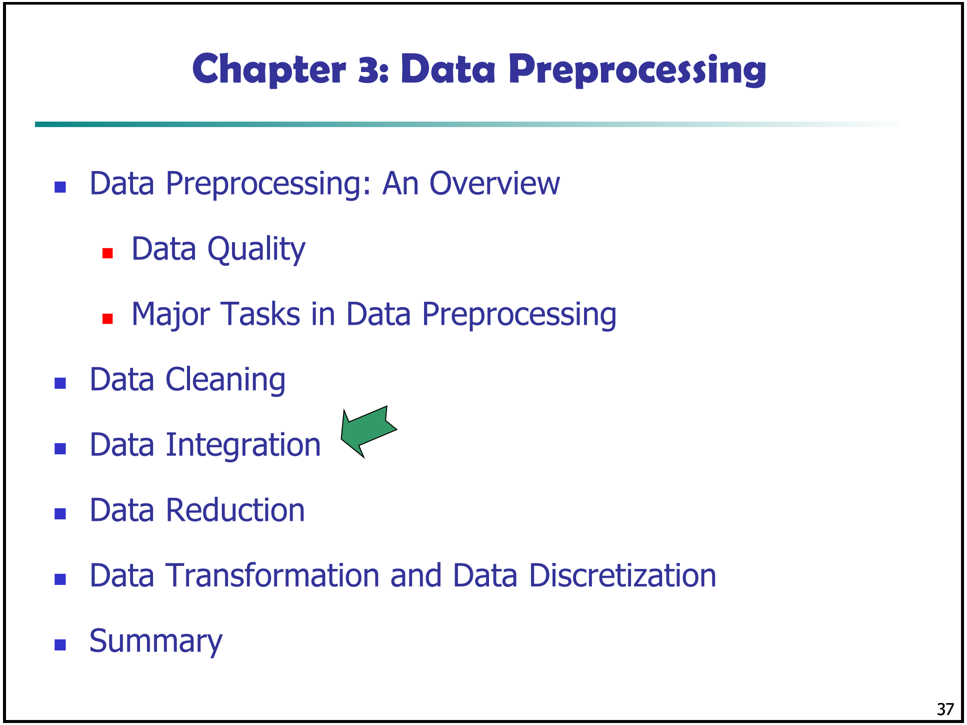


How to Handle Noisy Data?

- Regression
  - smooth by fitting the data into regression functions
    - Linear regression (two attributes)
    - Multiple linear regression (multiple attributes)
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

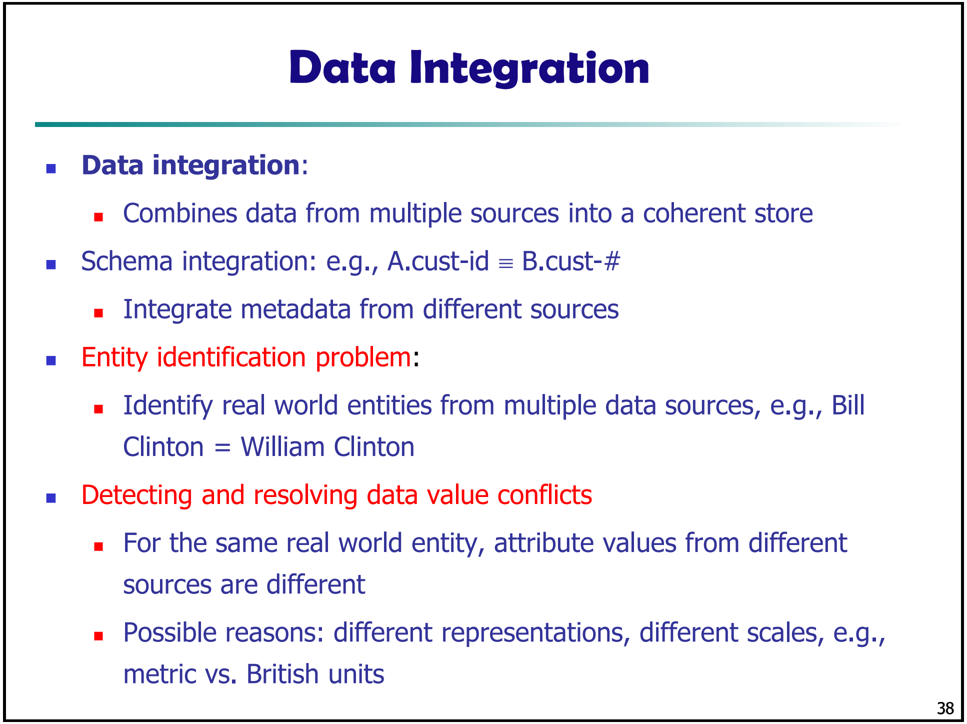
How to Handle Noisy Data?





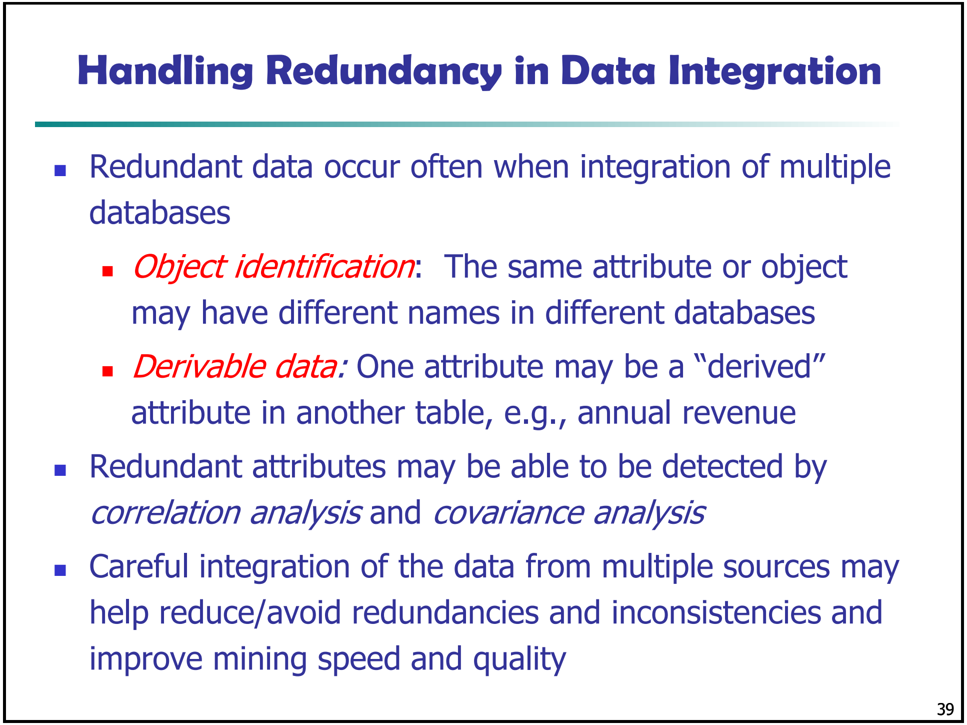
Chapter 3: Data Preprocessing

- 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

- **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
- 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



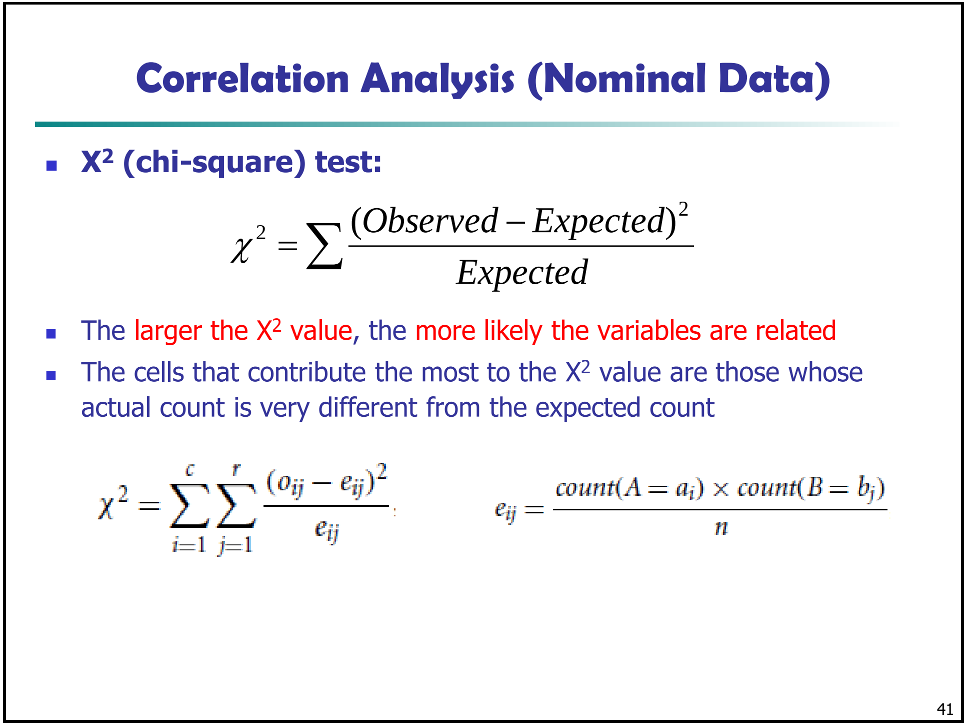
Handling Redundancy in Data Integration

- 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:** The chi-square independence test is a procedure for testing if two categorical variables A and B are related in some population.
- Suppose:
  - A= $\{a_1,\dots, a_c\}$
  - B= $\{b_1,\dots, b_r\}$
- The data tuples can be represented by a contingency table

	b_1	b_2	…	b_r	
a_1	$o_{1,1}$	$o_{1,2}$		$o_{1,c}$	$o_{1,.}$
a_2	$o_{2,1}$	$o_{2,2}$		$o_{2,c}$	$o_{2,.}$
…					
a_c	$o_{r,1}$	$o_{r,2}$		$o_{r,c}$	$o_{r,.}$
	$o_{.,1}$	$o_{.,2}$		$o_{.,r}$	Tot



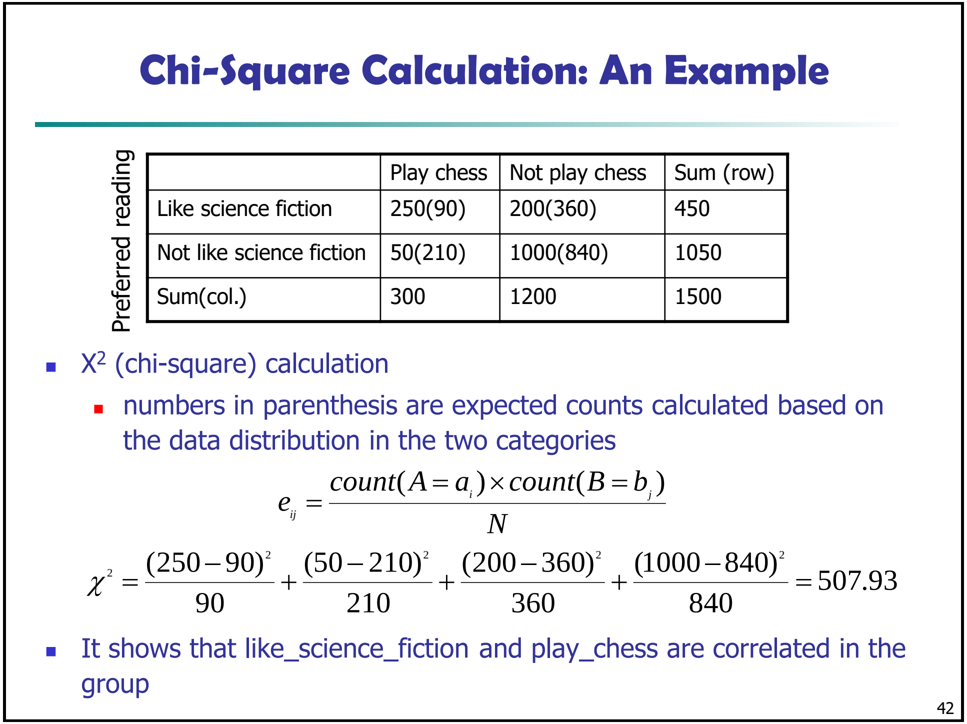
Correlation Analysis (Nominal Data)

- **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

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \qquad e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n}$$



Chi-Square Calculation: An Example

Preferred reading

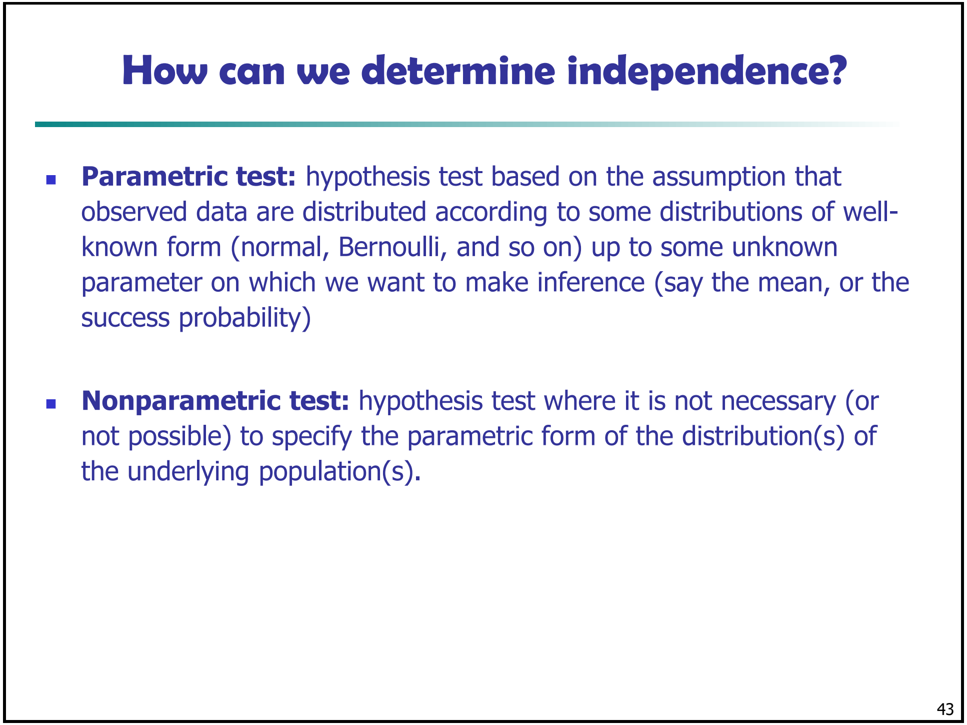
	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

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

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{N}$$

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

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

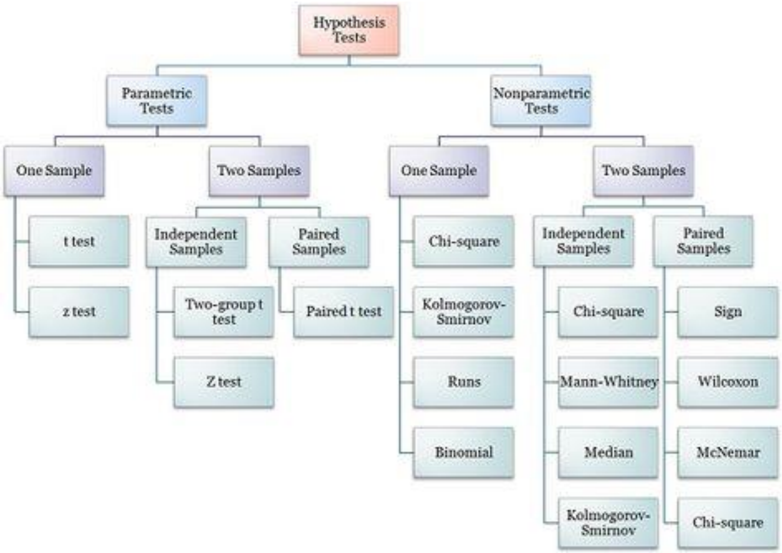


How can we determine independence?

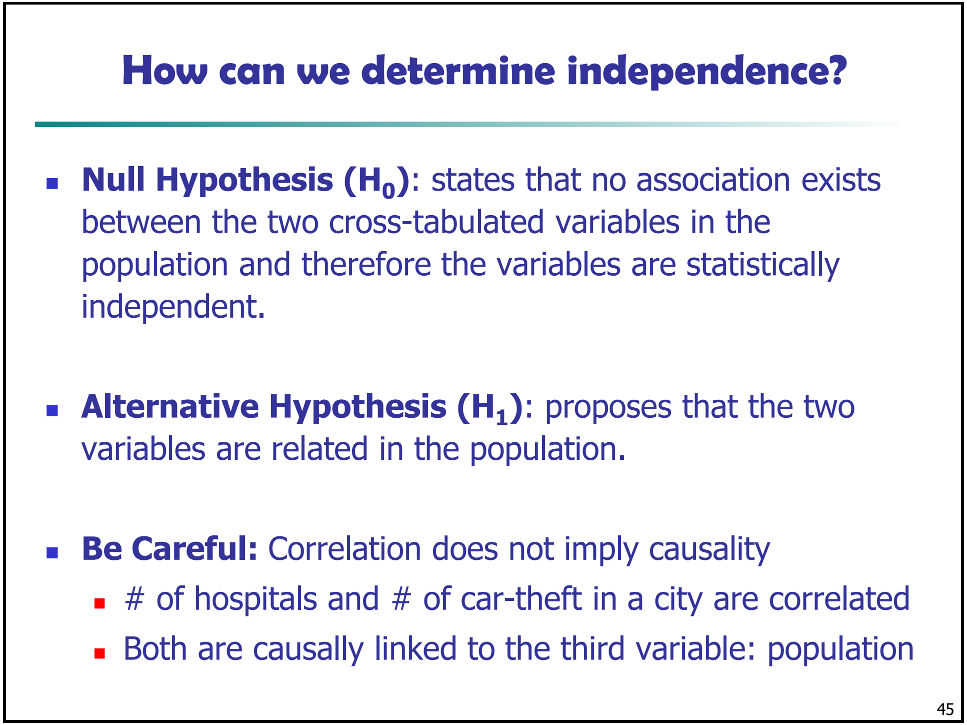
- **Parametric test:** hypothesis test based on the assumption that observed data are distributed according to some distributions of well-known form (normal, Bernoulli, and so on) up to some unknown parameter on which we want to make inference (say the mean, or the success probability)

- **Nonparametric test:** hypothesis test where it is not necessary (or not possible) to specify the parametric form of the distribution(s) of the underlying population(s).

How can we determine independence?



- Hypothesis Tests
  - Parametric Tests
    - One Sample
      - t test
      - z test
    - Two Samples
      - Independent Samples
        - Two-group t test
        - Z test
      - Paired Samples
        - Paired t test
  - Nonparametric Tests
    - One Sample
      - Chi-square
      - Kolmogorov-Smirnov
      - Runs
      - Binomial
    - Two Samples
      - Independent Samples
        - Chi-square
        - Mann-Whitney
        - Median
        - Kolmogorov-Smirnov
      - Paired Samples
        - Sign
        - Wilcoxon
        - McNemar
        - Chi-square

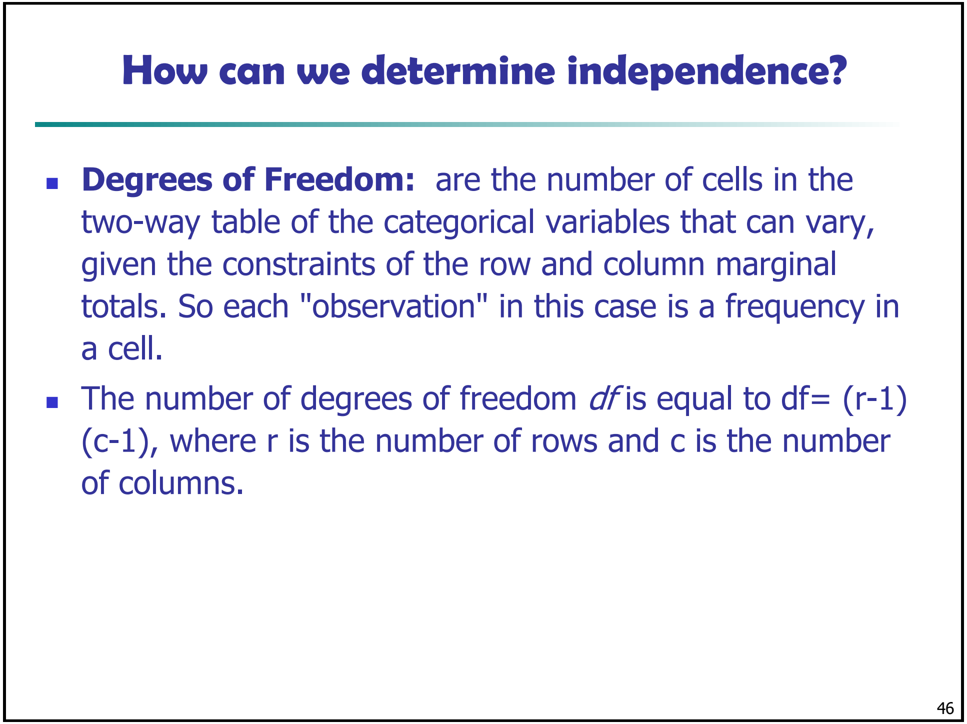


How can we determine independence?

- **Null Hypothesis (H_0)**: states that no association exists between the two cross-tabulated variables in the population and therefore the variables are statistically independent.

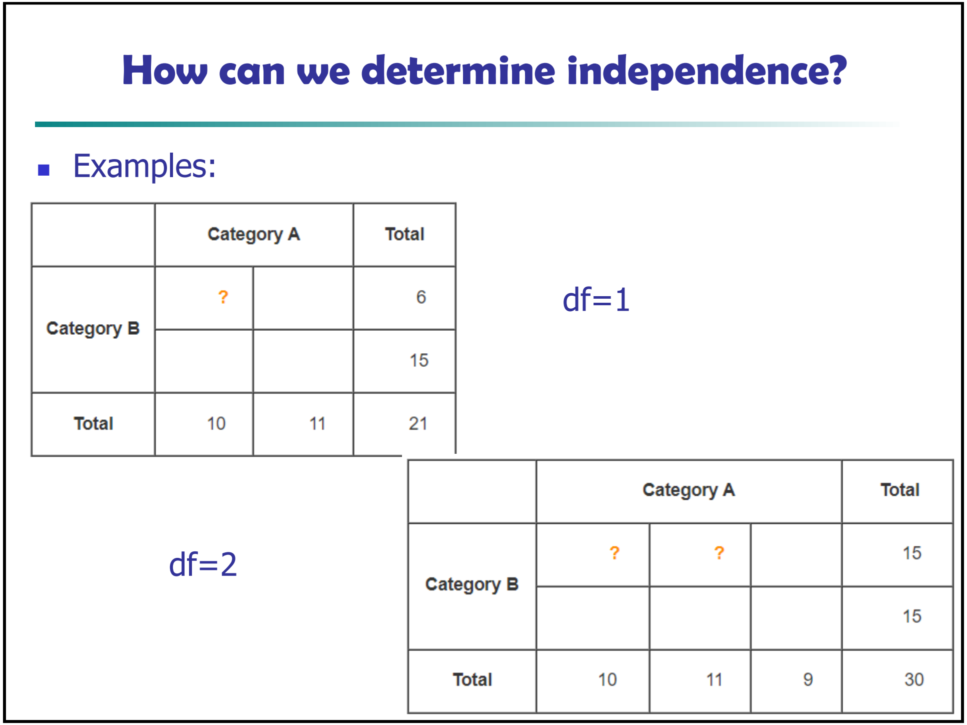
- **Alternative Hypothesis (H_1)**: proposes that the two variables are related in the population.

- **Be Careful:** 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



How can we determine independence?

- **Degrees of Freedom:** are the number of cells in the two-way table of the categorical variables that can vary, given the constraints of the row and column marginal totals. So each "observation" in this case is a frequency in a cell.
- The number of degrees of freedom *df* is equal to df= (r-1)(c-1), where r is the number of rows and c is the number of columns.



How can we determine independence?

- Examples:

	Category A		Total
Category B	?		6
			15
Total	10	11	21

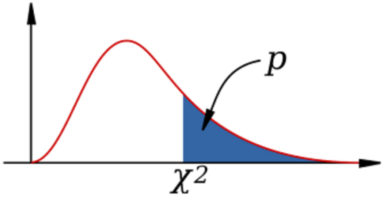
df=1

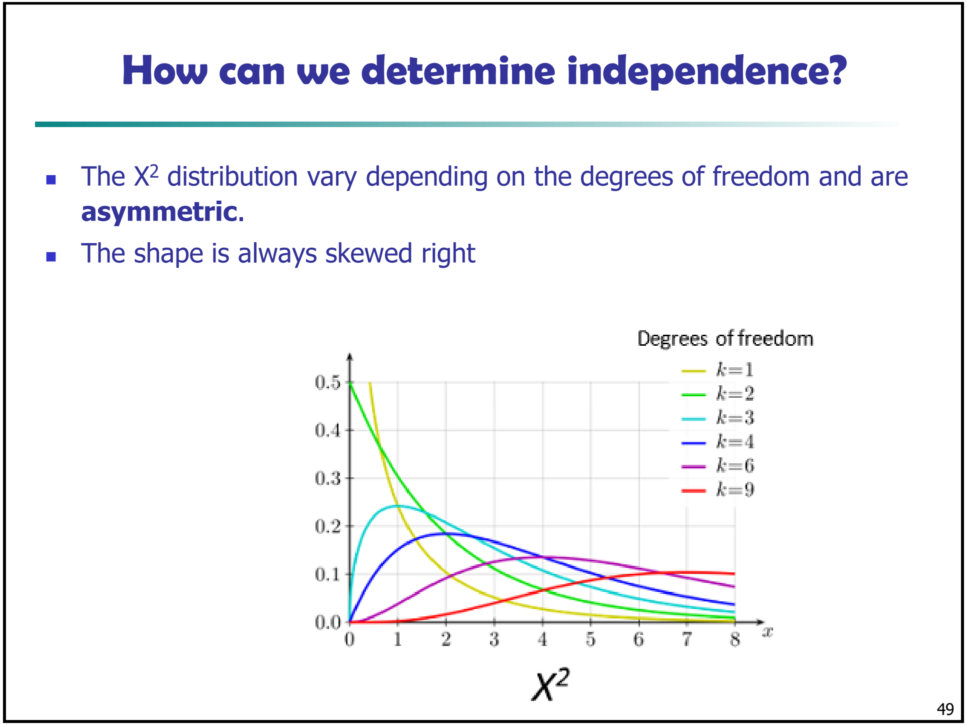
df=2

	Category A			Total
Category B	?	?		15
				15
Total	10	11	9	30

How can we determine independence?

- The X^2 distribution is a type of probability distribution.
- Probability distributions provide the probability of every possible value that may occur.
- Distributions that are cumulative **give the probability of a random variable being less than or equal to a particular value.**
- To find the probability of a particular value, we find the area under the curve before the value. The area that's after the value is called the **p-value**



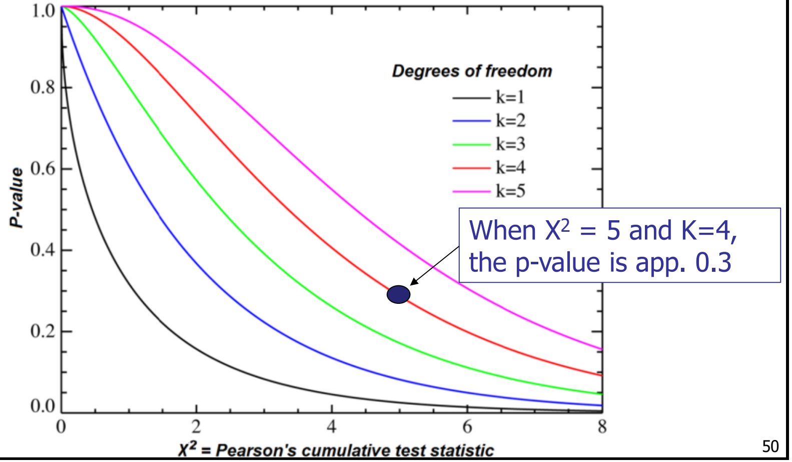


How can we determine independence?

- The X^2 distribution vary depending on the degrees of freedom and are **asymmetric**.
- The shape is always skewed right

How can we determine independence?

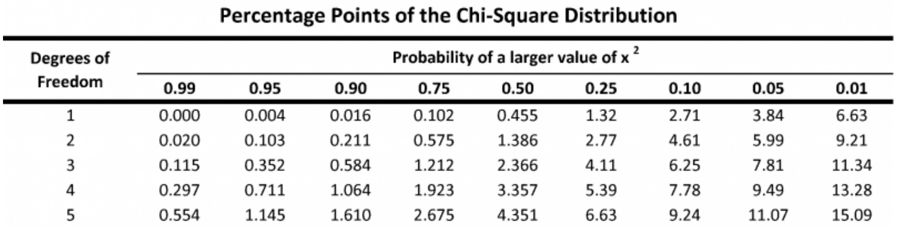
- Many statistical analyses involve using the p-value. However, calculating a portion of the area under the curve can be difficult. Alternatively



When $X^2 = 5$ and K=4, the p-value is app. 0.3

How can we determine independence?

- It's often more efficient to use a X^2 table. In this table, each row represents a different degree of freedom along with several X^2 values. The corresponding p-values are listed at the top of each column
- The null hypothesis is rejected when the probability of a larger value of X^2 is lower than the significance level α.

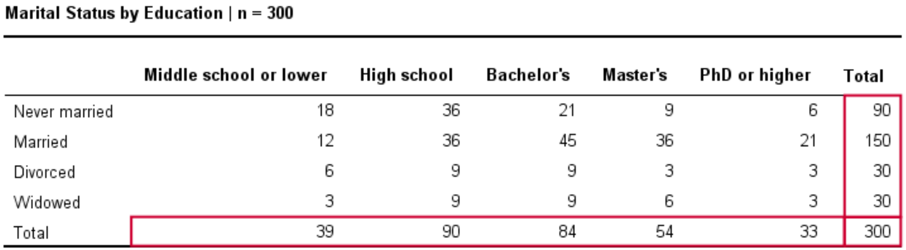


Percentage Points of the Chi-Square Distribution

Degrees of Freedom	Probability of a larger value of x^2								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09

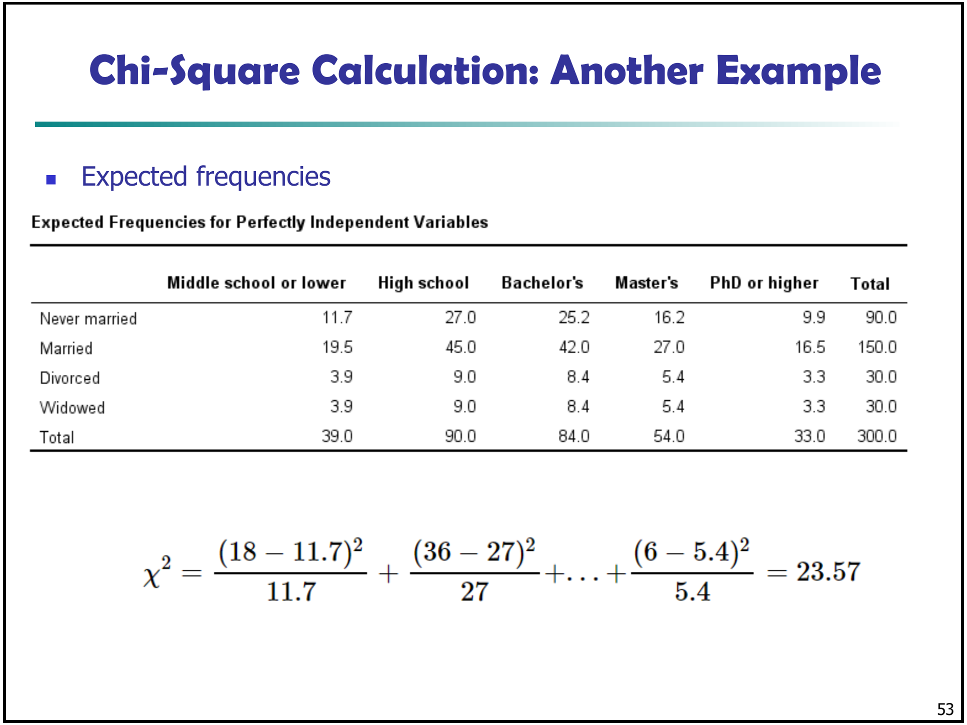
Chi-Square Calculation: Another Example

- A scientist wants to know if education level and marital status are related for all people in some country. He collects data on a simple random sample of n = 300 people



Marital Status by Education | n = 300

	Middle school or lower	High school	Bachelor's	Master's	PhD or higher	Total
Never married	18	36	21	9	6	90
Married	12	36	45	36	21	150
Divorced	6	9	9	3	3	30
Widowed	3	9	9	6	3	30
Total	39	90	84	54	33	300



Chi-Square Calculation: Another Example

- Expected frequencies

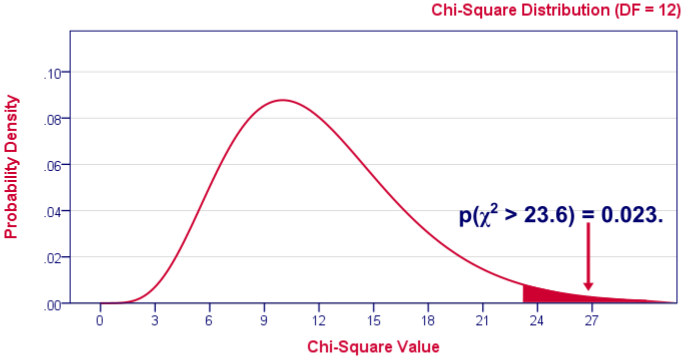
Expected Frequencies for Perfectly Independent Variables

	Middle school or lower	High school	Bachelor's	Master's	PhD or higher	Total
Never married	11.7	27.0	25.2	16.2	9.9	90.0
Married	19.5	45.0	42.0	27.0	16.5	150.0
Divorced	3.9	9.0	8.4	5.4	3.3	30.0
Widowed	3.9	9.0	8.4	5.4	3.3	30.0
Total	39.0	90.0	84.0	54.0	33.0	300.0

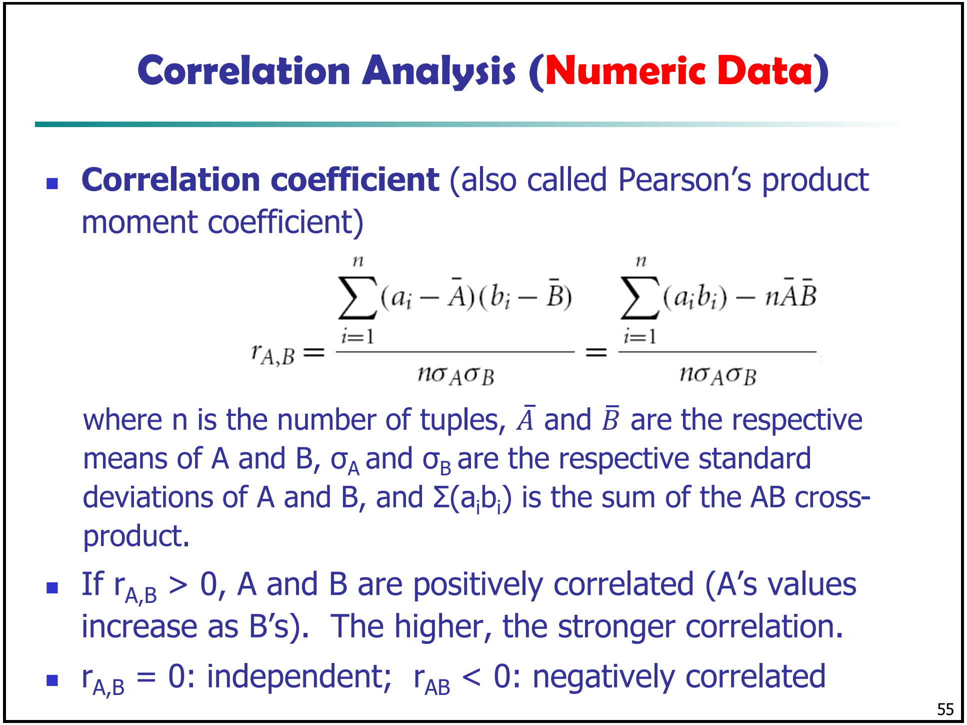
$$\chi^2 = \frac{(18-11.7)^2}{11.7} + \frac{(36-27)^2}{27} + \ldots + \frac{(6-5.4)^2}{5.4} = 23.57$$

Chi-Square Calculation: Another Example

- DF = 12



- Conclusion: marital status and education are related in our population



Correlation Analysis (Numeric Data)

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

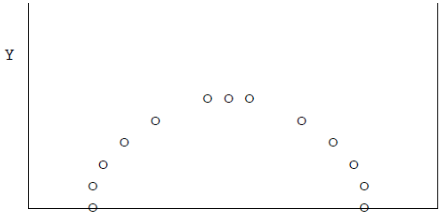
$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B}$$

  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 deviations of A and B, and $\Sigma(a_ib_i)$ is the sum of the AB cross-product.

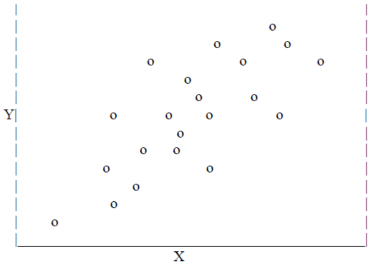
- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

Correlation Analysis (Numeric Data)

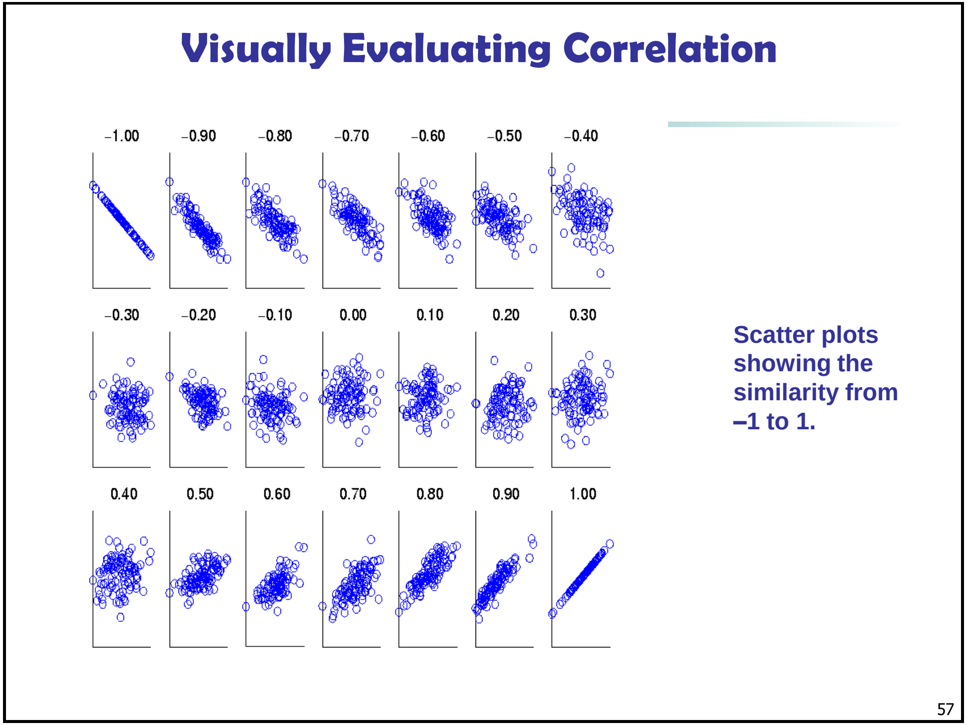
- **Be careful:** The Pearson's product moment coefficient only detects linear relationships



Correlation 0

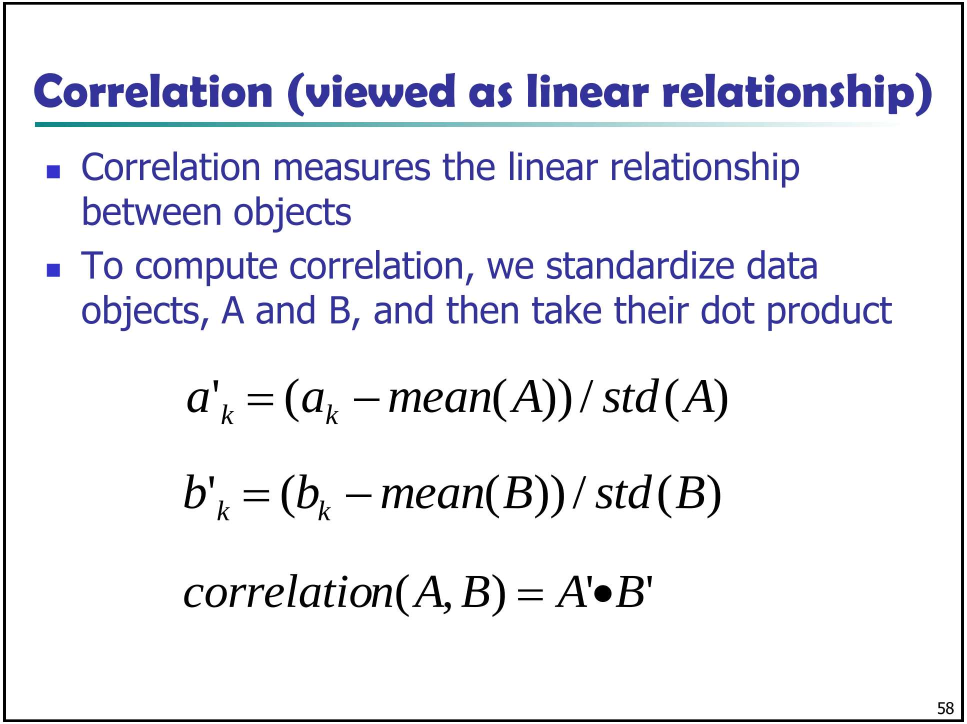


Correlation 0.8



Visually Evaluating Correlation

Scatter plots showing the similarity from –1 to 1.



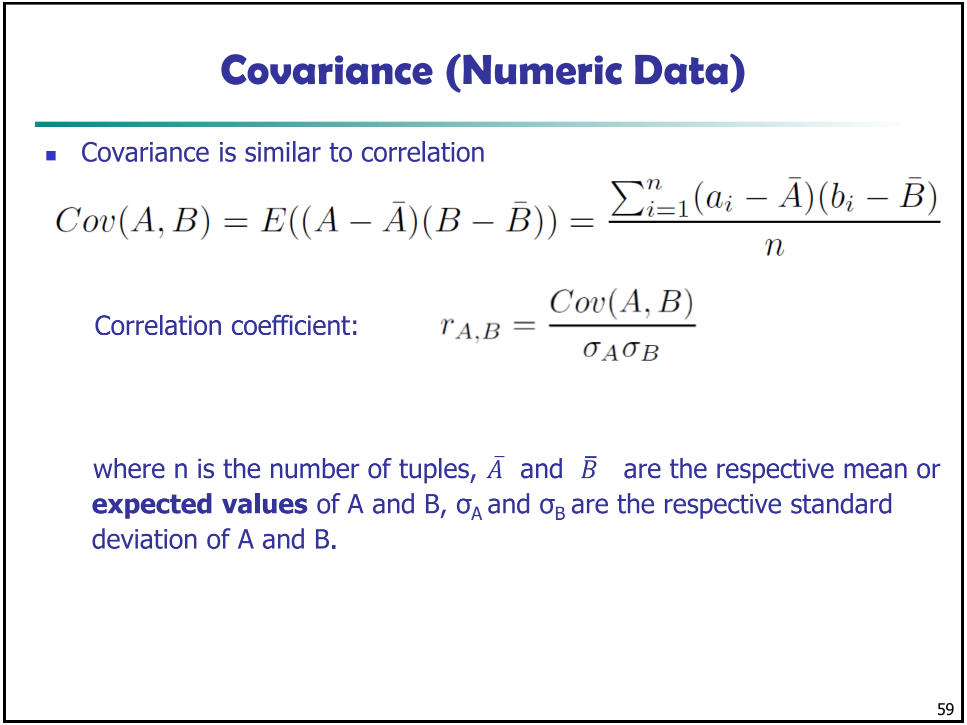
Correlation (viewed as linear relationship)

- 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

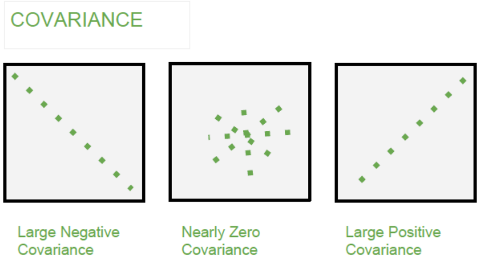
$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

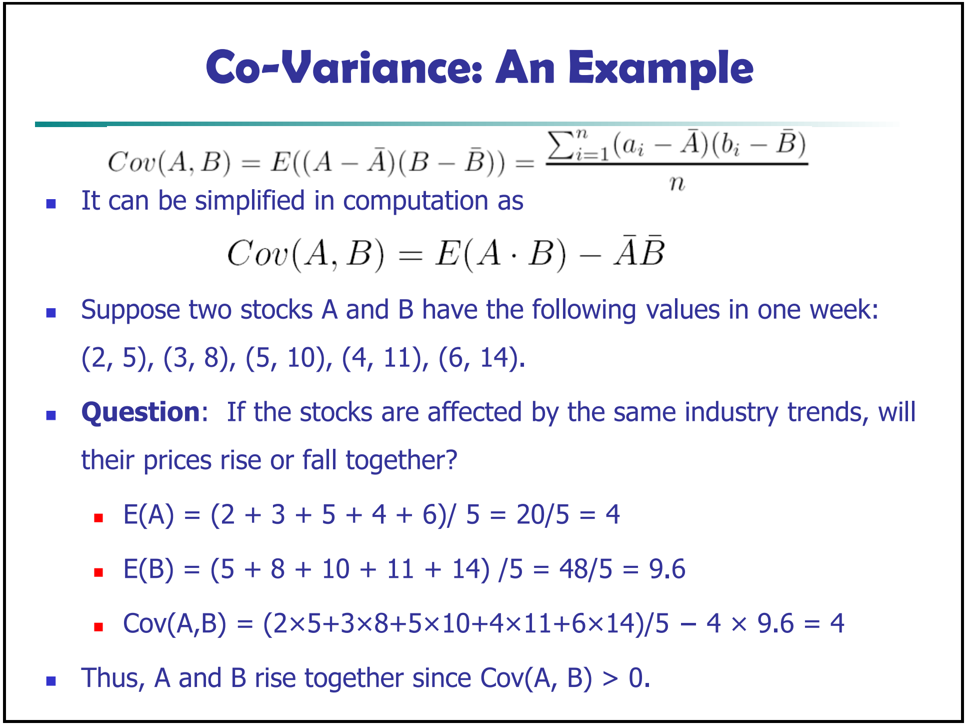
Correlation coefficient: $$r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$$

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

Covariance (Numeric Data)

- **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





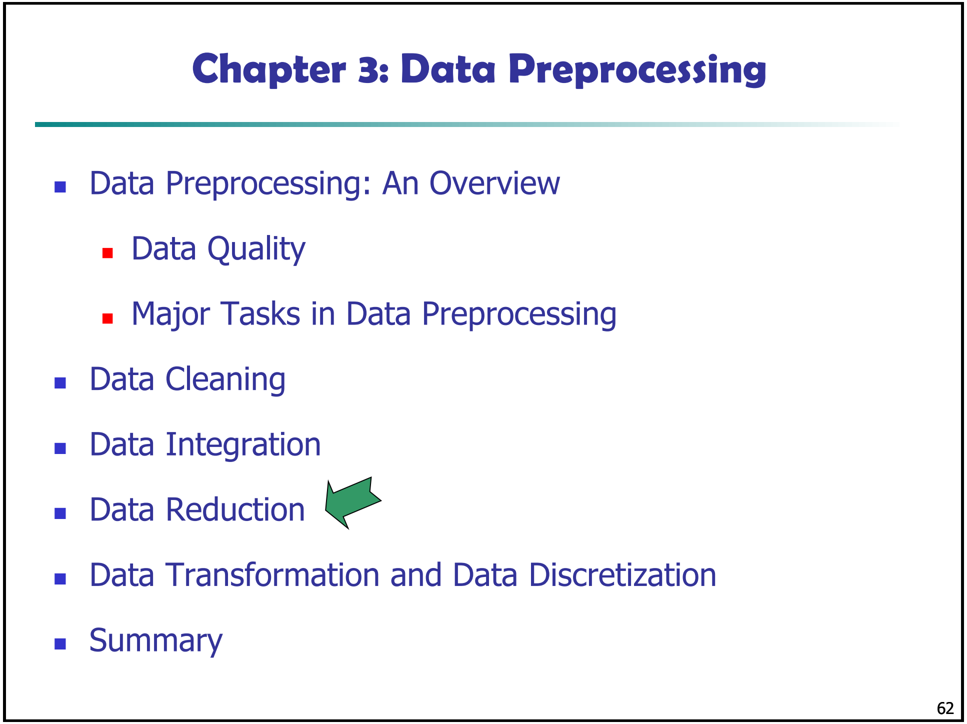
Co-Variance: An Example

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

- It can be simplified in computation as

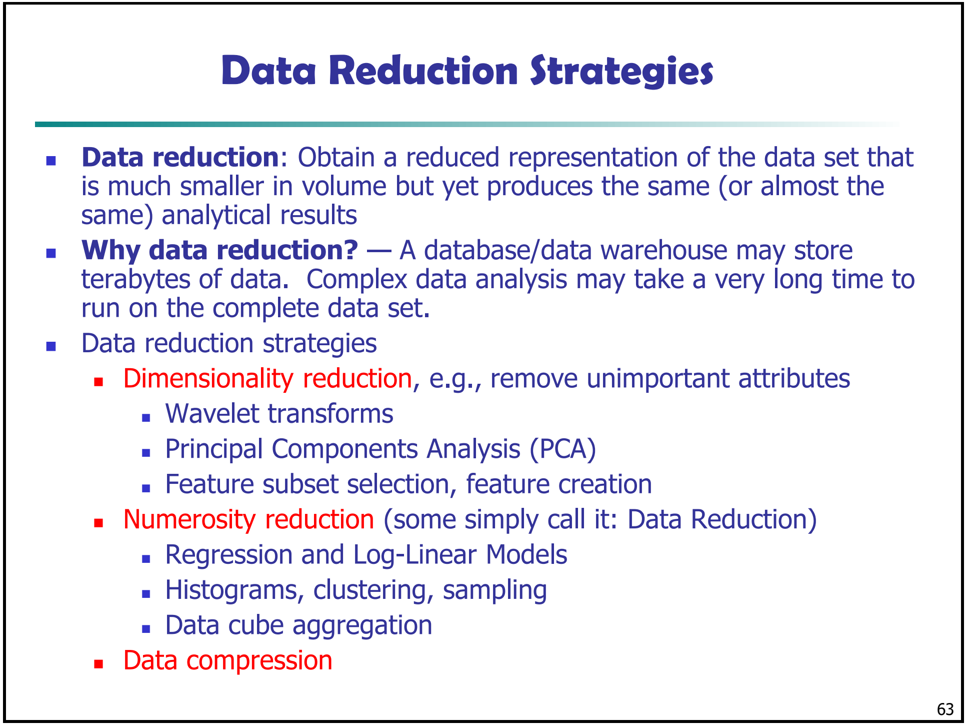
$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}$$

- 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.



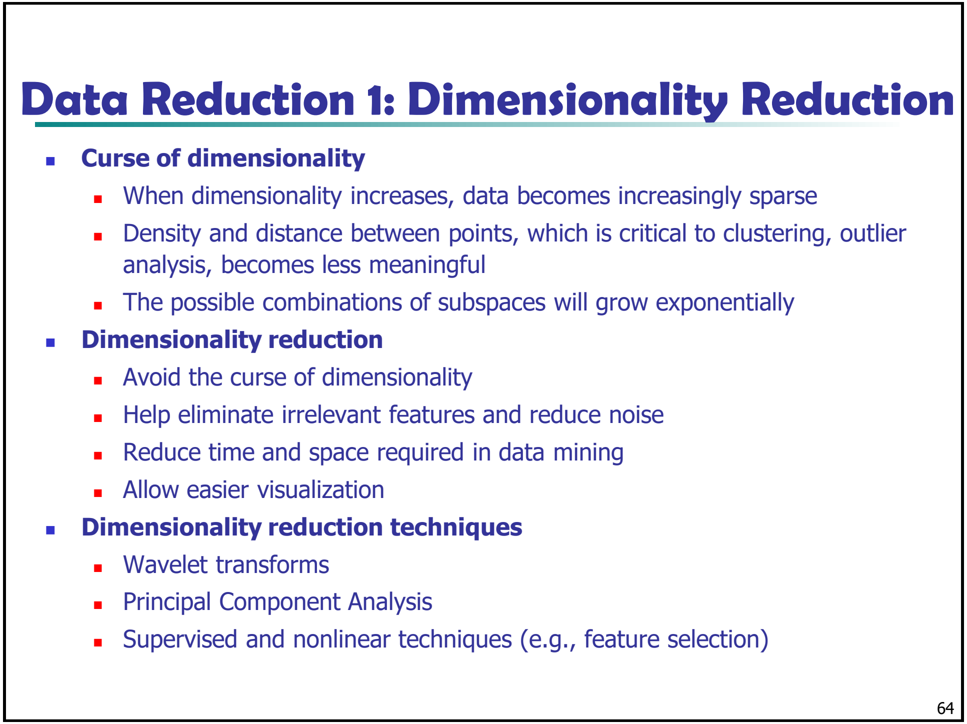
Chapter 3: Data Preprocessing

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



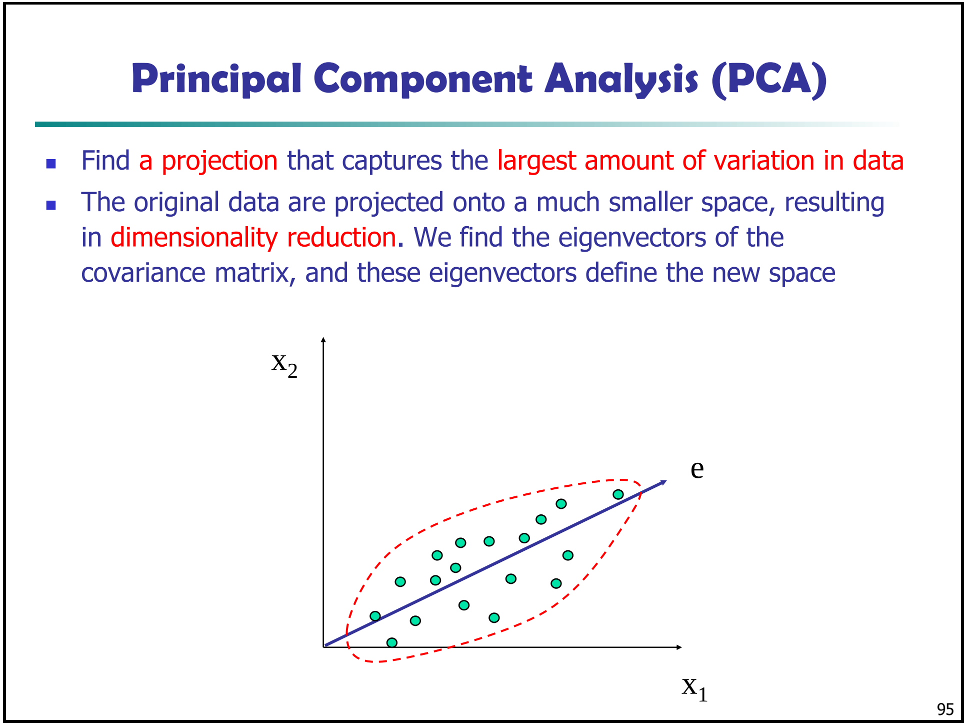
Data Reduction Strategies

- **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



Data Reduction 1: Dimensionality Reduction

- **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)

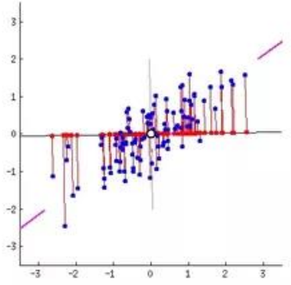


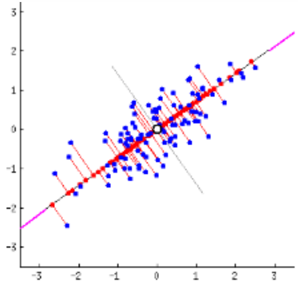
Principal Component Analysis (PCA)

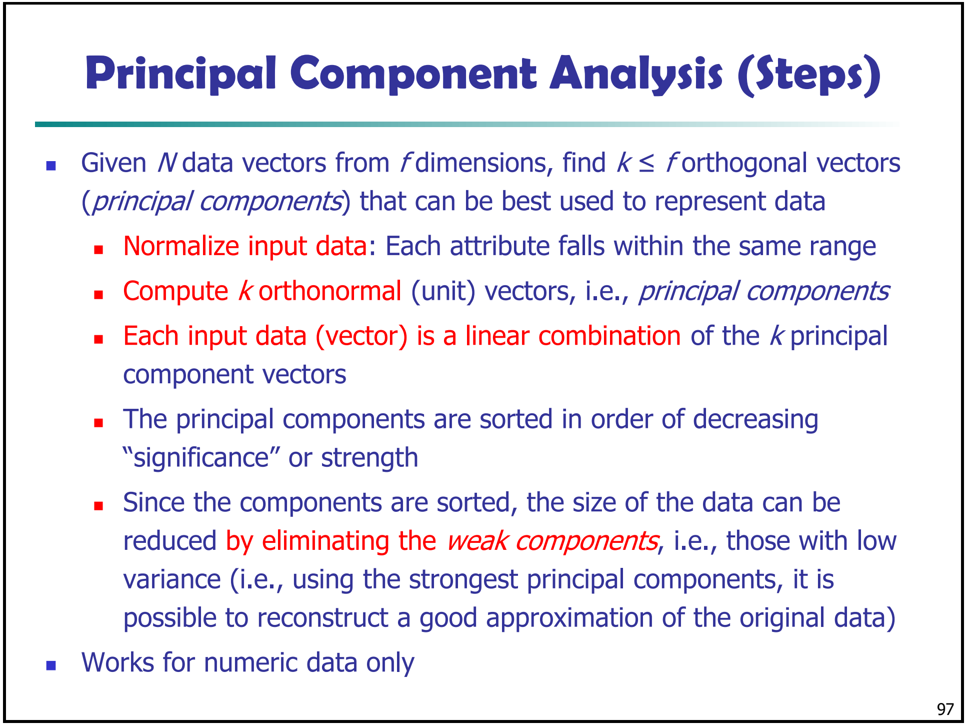
- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space

Principal Component Analysis (PCA)

- The projection error is less than in the original dataset
- Newly projected red points are more widely spread out than in the original dataset, i.e. more variance.







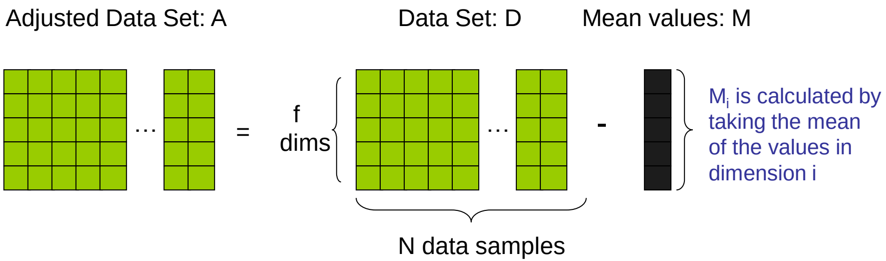
Principal Component Analysis (Steps)

- Given *N* data vectors from *f* dimensions, find *k* ≤ *f* 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

Principal Component Analysis (Steps)

- Given *N* data vectors from *f*-dimensions, find $k \leq f$ orthogonal vectors (*principal components*) that can be best used to represent data

Step #1: Calculate Adjusted Data Set



Adjusted Data Set: A

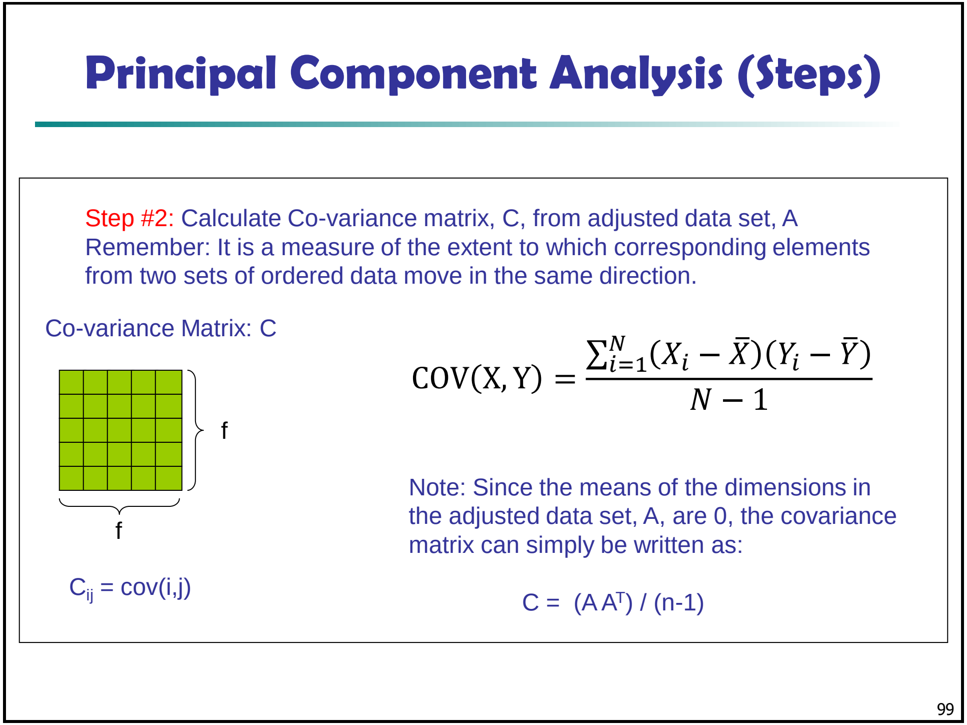
Data Set: D

Mean values: M

… = f dims … -

N data samples

M_i is calculated by taking the mean of the values in dimension i



Principal Component Analysis (Steps)

Step #2: Calculate Co-variance matrix, C, from adjusted data set, A
Remember: It is a measure of the extent to which corresponding elements from two sets of ordered data move in the same direction.

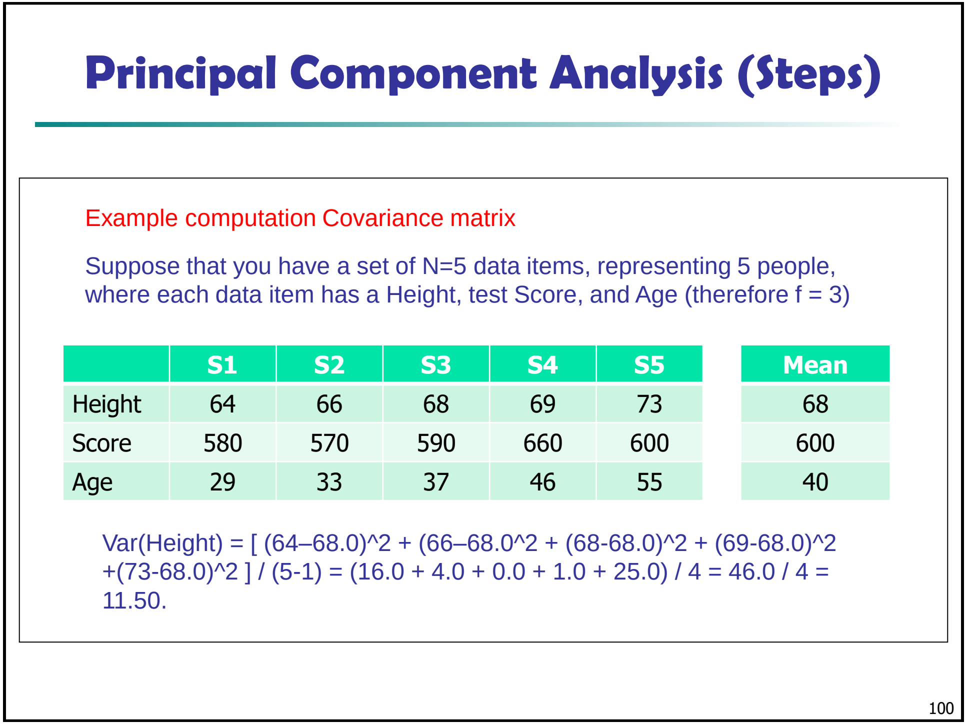
Co-variance Matrix: C

C_{ij} = cov(i,j)

$$\mathrm{COV}(X,Y) = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{N-1}$$

Note: Since the means of the dimensions in the adjusted data set, A, are 0, the covariance matrix can simply be written as:

$$C = (A\,A^T) / (n-1)$$



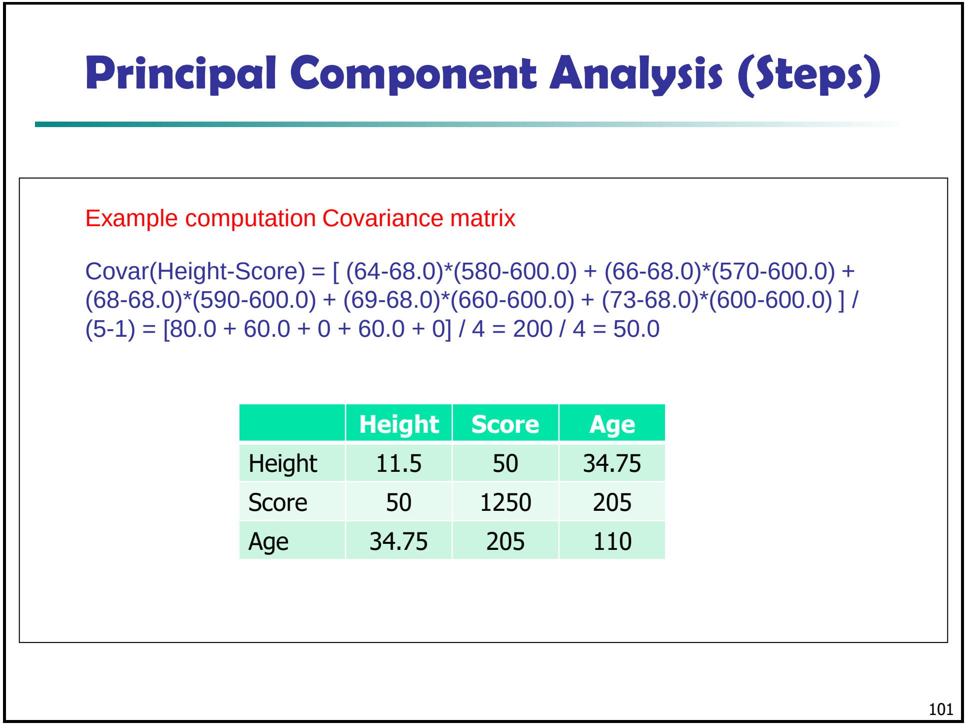
Principal Component Analysis (Steps)

Example computation Covariance matrix

Suppose that you have a set of N=5 data items, representing 5 people, where each data item has a Height, test Score, and Age (therefore f = 3)

	S1	S2	S3	S4	S5		Mean
Height	64	66	68	69	73		68
Score	580	570	590	660	600		600
Age	29	33	37	46	55		40

Var(Height) = [(64–68.0)^2 + (66–68.0^2 + (68-68.0)^2 + (69-68.0)^2 +(73-68.0)^2] / (5-1) = (16.0 + 4.0 + 0.0 + 1.0 + 25.0) / 4 = 46.0 / 4 = 11.50.



Principal Component Analysis (Steps)

Example computation Covariance matrix

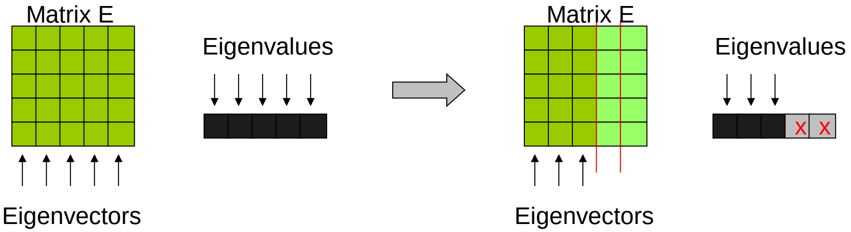
Covar(Height-Score) = [(64-68.0)*(580-600.0) + (66-68.0)*(570-600.0) + (68-68.0)*(590-600.0) + (69-68.0)*(660-600.0) + (73-68.0)*(600-600.0)] / (5-1) = [80.0 + 60.0 + 0 + 60.0 + 0] / 4 = 200 / 4 = 50.0

	Height	Score	Age
Height	11.5	50	34.75
Score	50	1250	205
Age	34.75	205	110

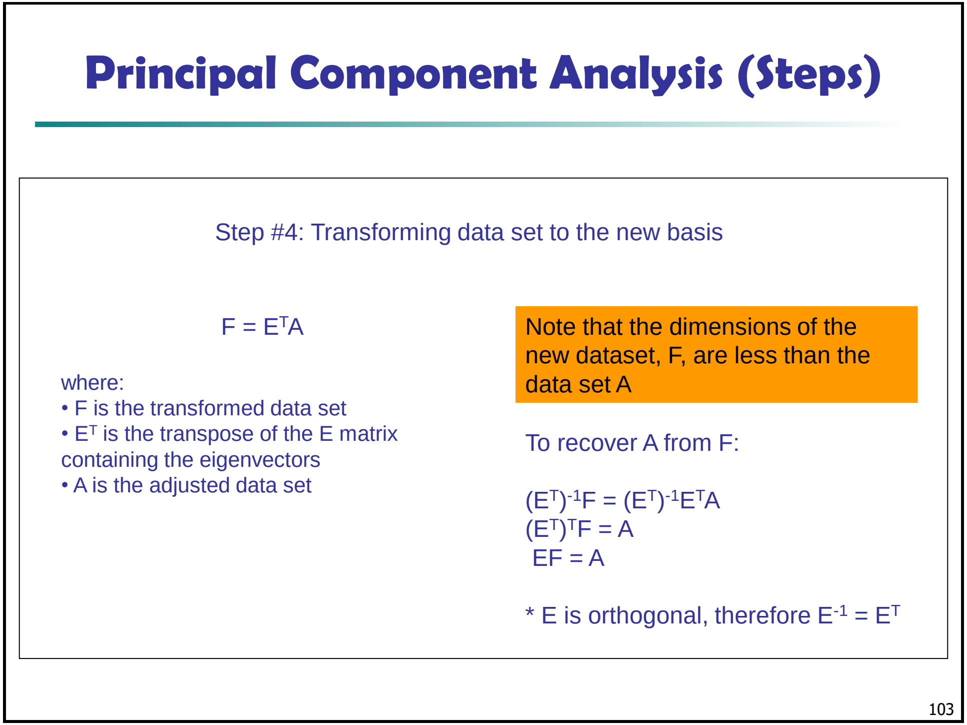
101

Principal Component Analysis (Steps)

Step #3: Calculate eigenvectors and eigenvalues of C



If some eigenvalues are 0 or very small, we can essentially discard those eigenvalues and the corresponding eigenvectors, hence reducing the dimensionality of the new basis.



Principal Component Analysis (Steps)

Step #4: Transforming data set to the new basis

$F = E^{T}A$

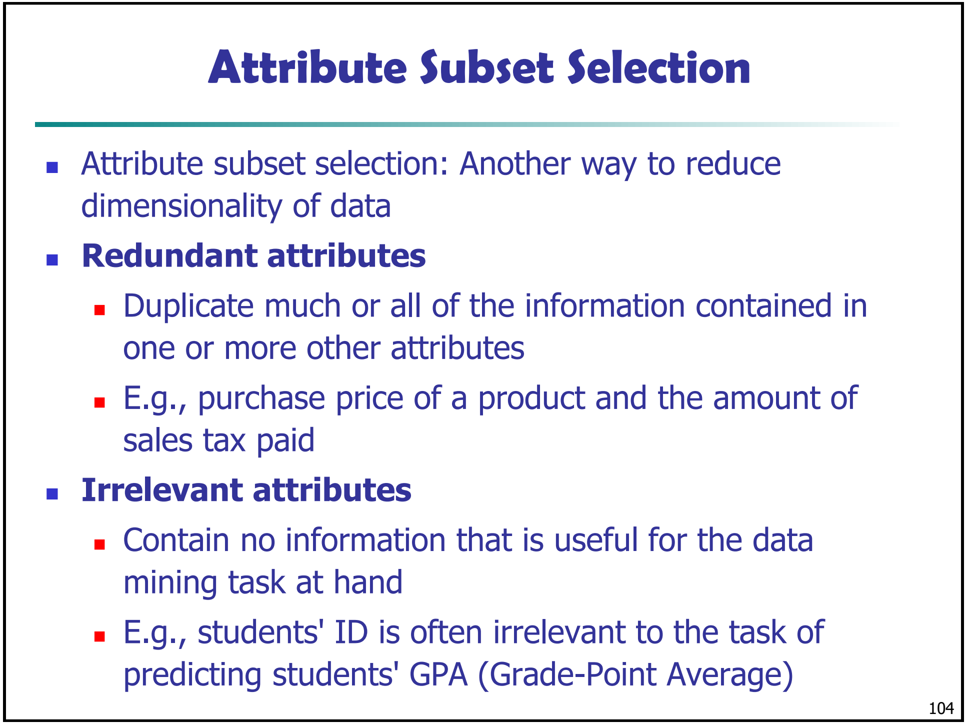
where:
- F is the transformed data set
- E^{T} is the transpose of the E matrix containing the eigenvectors
- A is the adjusted data set

Note that the dimensions of the new dataset, F, are less than the data set A

To recover A from F:

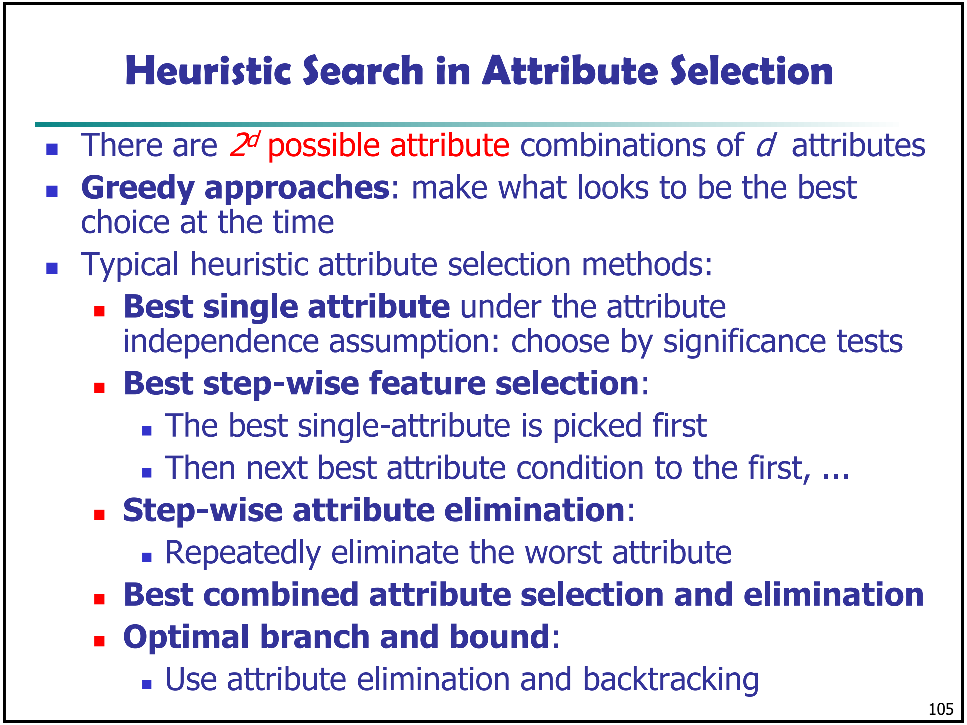
$(E^{T})^{-1}F = (E^{T})^{-1}E^{T}A$
$(E^{T})^{T}F = A$
$EF = A$

* E is orthogonal, therefore $E^{-1} = E^{T}$



Attribute Subset Selection

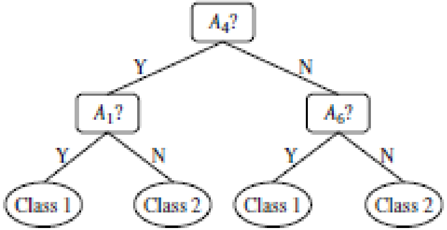
- Attribute subset selection: Another way to reduce dimensionality of data
- **Redundant attributes**
  - Duplicate much or all of the information contained in one or more other attributes
  - E.g., purchase price of a product and the amount of sales tax paid
- **Irrelevant attributes**
  - Contain no 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)

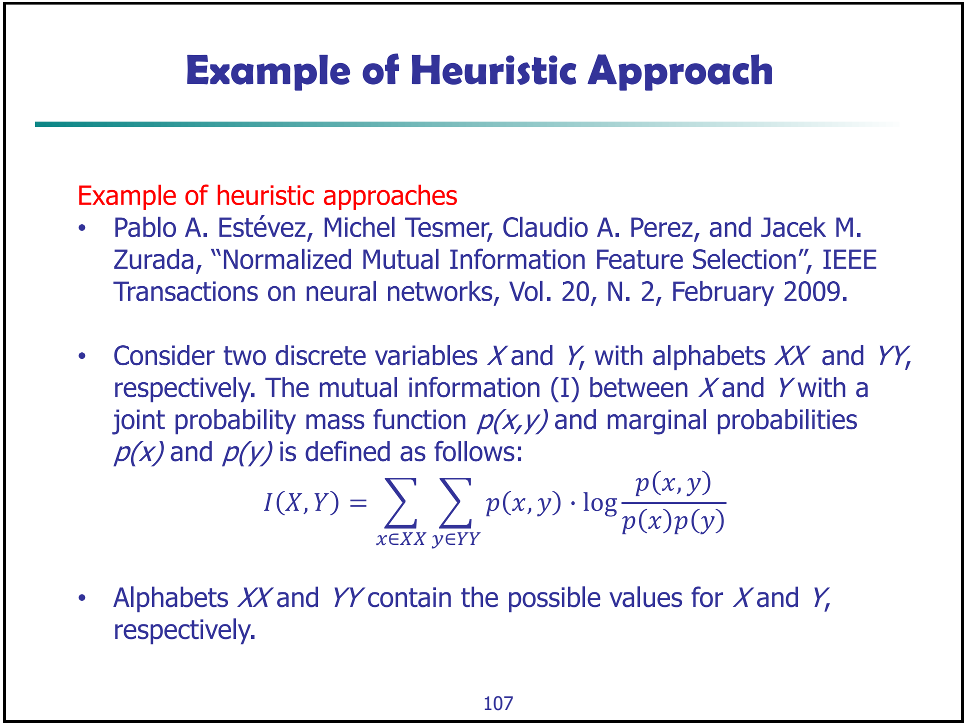


Heuristic Search in Attribute Selection

- There are 2^d possible attribute combinations of d attributes
- **Greedy approaches**: make what looks to be the best choice at the time
- Typical heuristic attribute selection methods:
  - **Best single attribute** under the attribute independence assumption: choose by significance tests
  - **Best step-wise feature selection**:
    - The best single-attribute is picked first
    - Then next best attribute condition to the first, ...
  - **Step-wise attribute elimination**:
    - Repeatedly eliminate the worst attribute
  - **Best combined attribute selection and elimination**
  - **Optimal branch and bound**:
    - Use attribute elimination and backtracking

Heuristic Search in Attribute Selection

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  A_4? Y N A_1? A_6? Y N Y N Class 1 Class 2 Class 1 Class 2 => Reduced attribute set: $\{A_1, A_4, A_6\}$



Example of Heuristic Approach

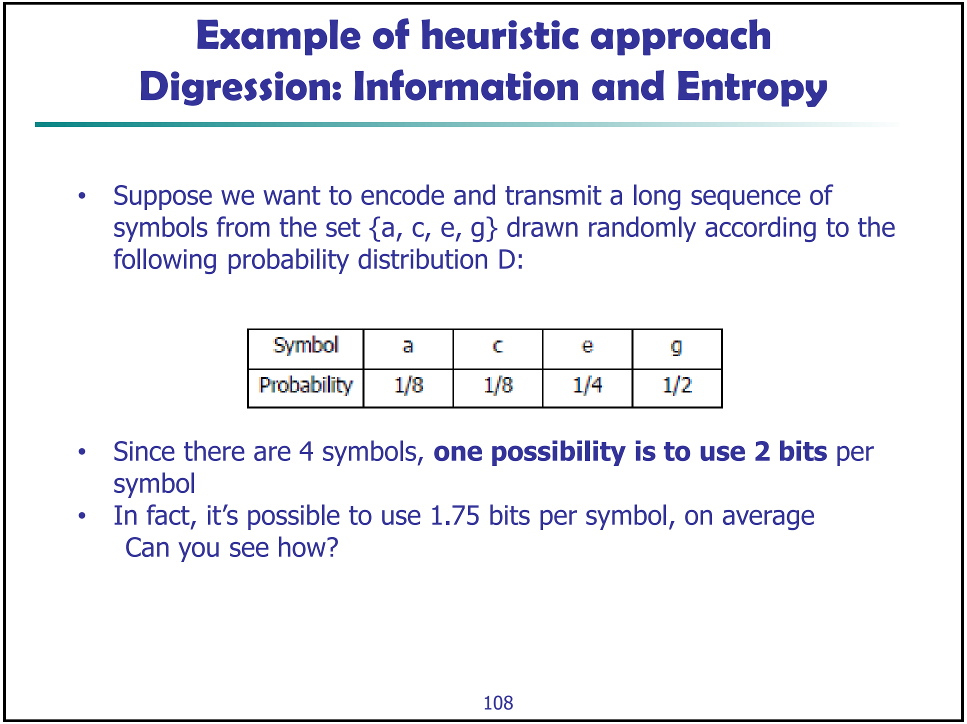
Example of heuristic approaches

- Pablo A. Estévez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada, “Normalized Mutual Information Feature Selection”, IEEE Transactions on neural networks, Vol. 20, N. 2, February 2009.

- Consider two discrete variables *X* and *Y*, with alphabets *XX* and *YY*, respectively. The mutual information (I) between *X* and *Y* with a joint probability mass function *p(x,y)* and marginal probabilities *p(x)* and *p(y)* is defined as follows:

$$I(X,Y) = \sum_{x \in XX} \sum_{y \in YY} p(x,y) \cdot \log \frac{p(x,y)}{p(x)p(y)}$$

- Alphabets *XX* and *YY* contain the possible values for *X* and *Y*, respectively.

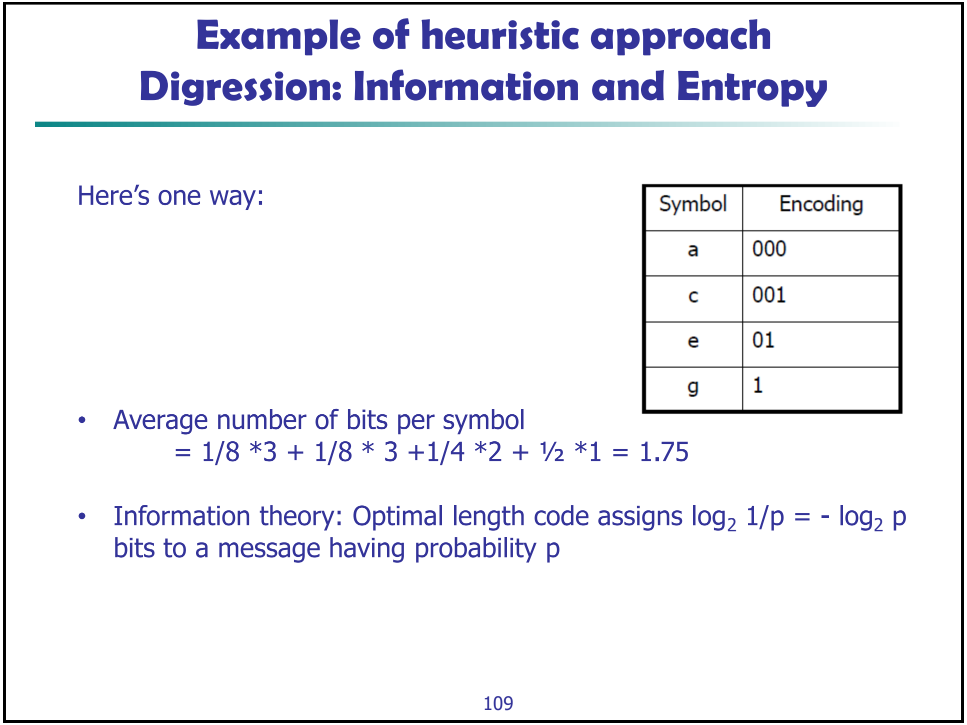


Example of heuristic approach
Digression: Information and Entropy

- Suppose we want to encode and transmit a long sequence of symbols from the set {a, c, e, g} drawn randomly according to the following probability distribution D:

Symbol	a	c	e	g
Probability	1/8	1/8	1/4	1/2

- Since there are 4 symbols, **one possibility is to use 2 bits** per symbol
- In fact, it's possible to use 1.75 bits per symbol, on average
  Can you see how?



Example of heuristic approach
Digression: Information and Entropy

Here’s one way:

Symbol	Encoding
a	000
c	001
e	01
g	1

- Average number of bits per symbol
  = 1/8 *3 + 1/8 * 3 +1/4 *2 + ½ *1 = 1.75

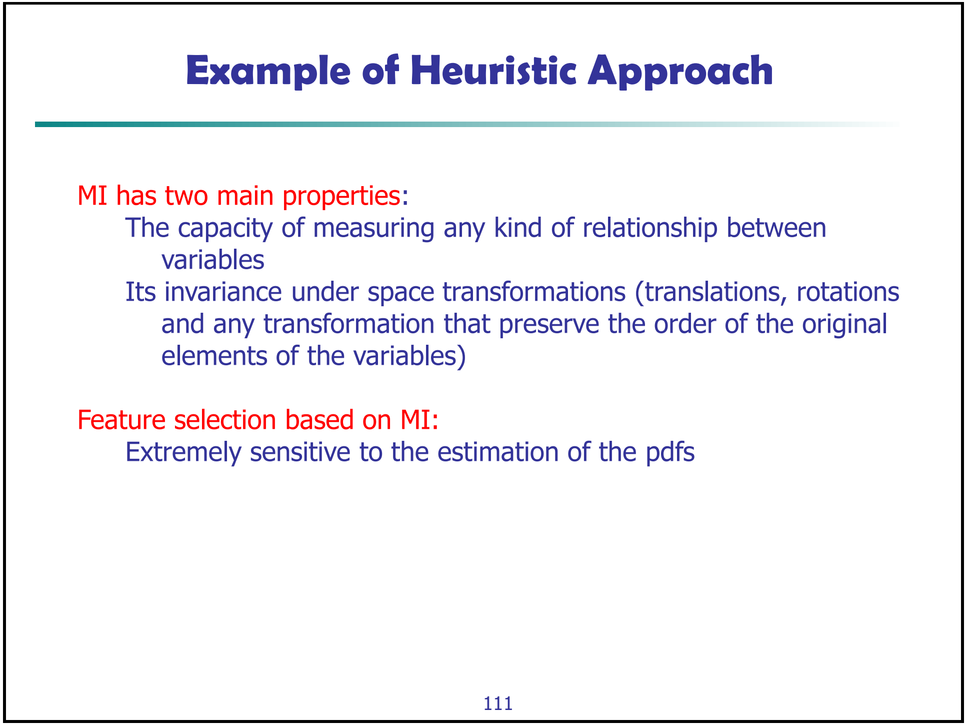
- Information theory: Optimal length code assigns $\log_2 1/p = - \log_2 p$ bits to a message having probability p

Example of heuristic approach
Digression: Information and Entropy

- Given a distribution D over a finite set, where <p_1, p_2, ..., p_n> are the corresponding probabilities, define the entropy of D by

$$H(D) = -\sum_i p_i \, log_2 \, p_i$$

For example, the entropy of the distribution we just examined, <⅛, ⅛, ¼, ½>, is 1.75 (bits)
Also called information
In general, entropy is higher the closer the distribution is to being uniform



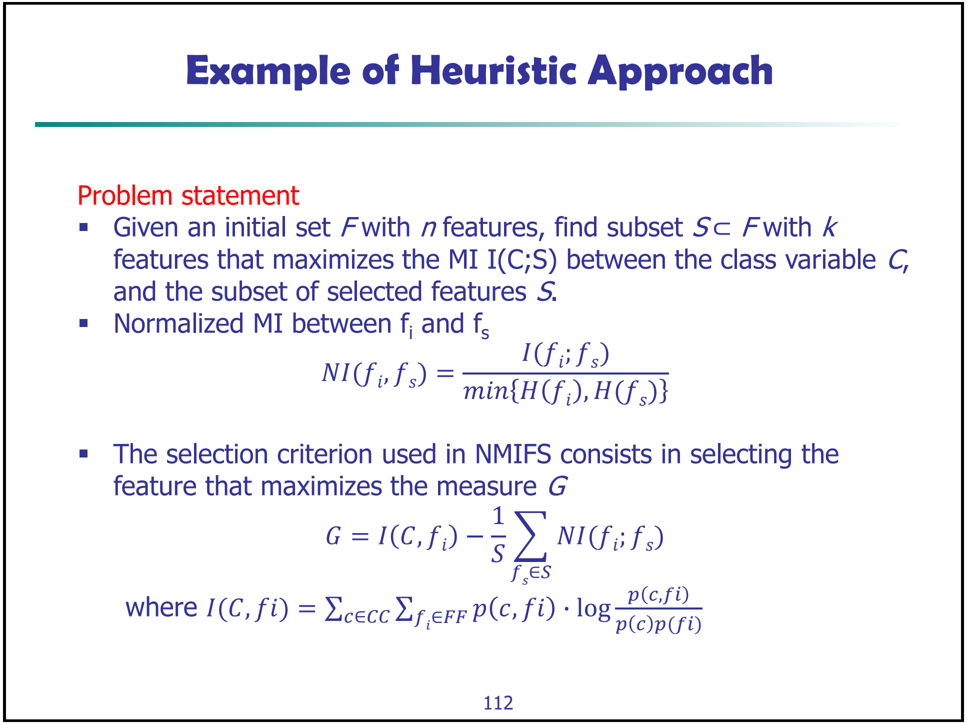
Example of Heuristic Approach

MI has two main properties:

- The capacity of measuring any kind of relationship between variables
- Its invariance under space transformations (translations, rotations and any transformation that preserve the order of the original elements of the variables)

Feature selection based on MI:

- Extremely sensitive to the estimation of the pdfs



Example of Heuristic Approach

Problem statement

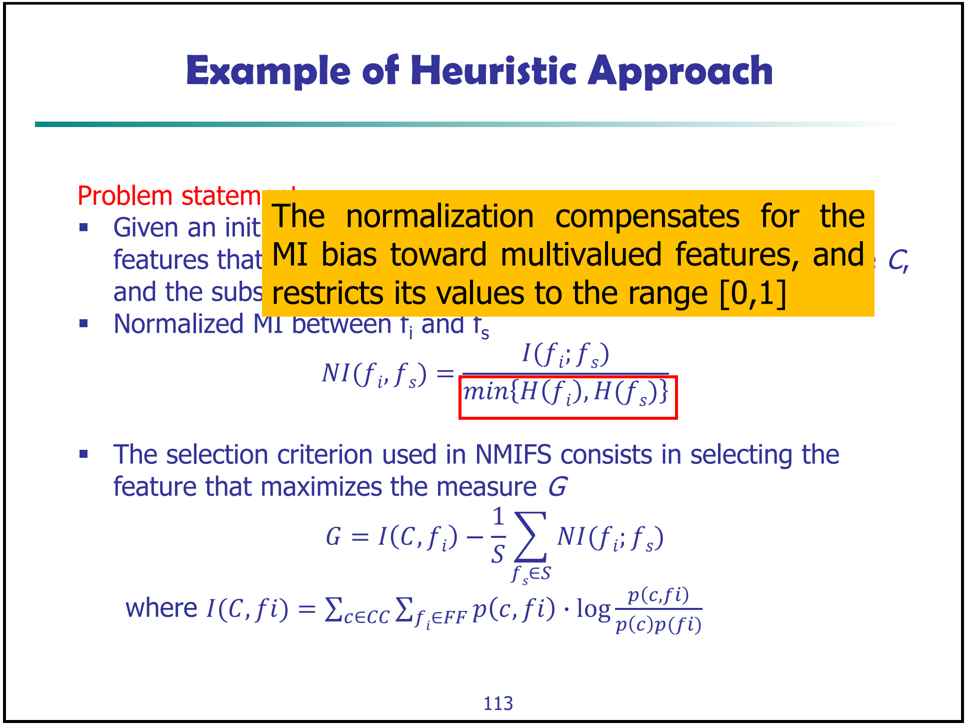
- Given an initial set *F* with *n* features, find subset *S* ⊂ *F* with *k* features that maximizes the MI I(C;S) between the class variable *C*, and the subset of selected features *S*.
- Normalized MI between f_i and f_s

$$NI(f_i, f_s) = \frac{I(f_i; f_s)}{min\{H(f_i), H(f_s)\}}$$

- The selection criterion used in NMIFS consists in selecting the feature that maximizes the measure *G*

$$G = I(C, f_i) - \frac{1}{S}\sum_{f_s \in S} NI(f_i; f_s)$$

where $I(C, fi) = \sum_{c\in CC}\sum_{f_i \in FF} p(c, fi) \cdot \log\frac{p(c,fi)}{p(c)p(fi)}$



Example of Heuristic Approach

Problem statem

- Given an init
  features that
  and the subs

The normalization compensates for the MI bias toward multivalued features, and restricts its values to the range [0,1]

C,

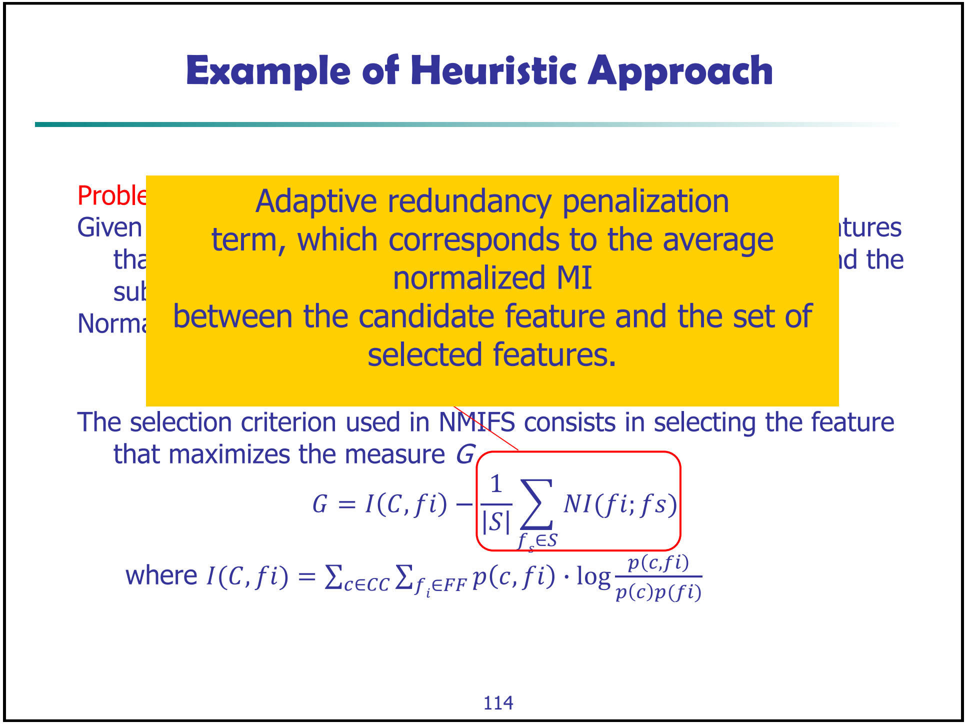
- Normalized MI between f_i and f_s

$$NI(f_i, f_s) = \frac{I(f_i; f_s)}{\boxed{min\{H(f_i), H(f_s)\}}}$$

- The selection criterion used in NMIFS consists in selecting the feature that maximizes the measure *G*

$$G = I(C, f_i) - \frac{1}{S}\sum_{f_s \in S} NI(f_i; f_s)$$

where $I(C, fi) = \sum_{c \in CC} \sum_{f_i \in FF} p(c, fi) \cdot \log \frac{p(c,fi)}{p(c)p(fi)}$



Example of Heuristic Approach

Proble

Given … tures

tha … nd the

sub

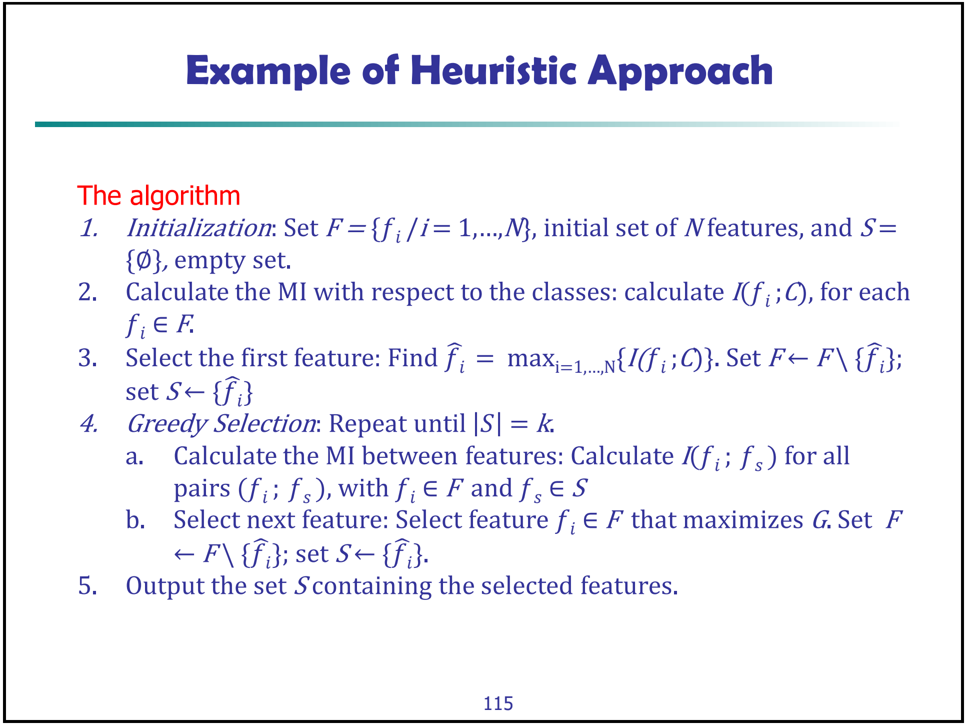
Norma

Adaptive redundancy penalization term, which corresponds to the average normalized MI between the candidate feature and the set of selected features.

The selection criterion used in NMIFS consists in selecting the feature that maximizes the measure G

$$G = I(C, fi) - \boxed{\frac{1}{|S|}\sum_{f_s \in S} NI(fi; fs)}$$

where $I(C, fi) = \sum_{c \in CC} \sum_{f_i \in FF} p(c, fi) \cdot \log \frac{p(c, fi)}{p(c)p(fi)}$



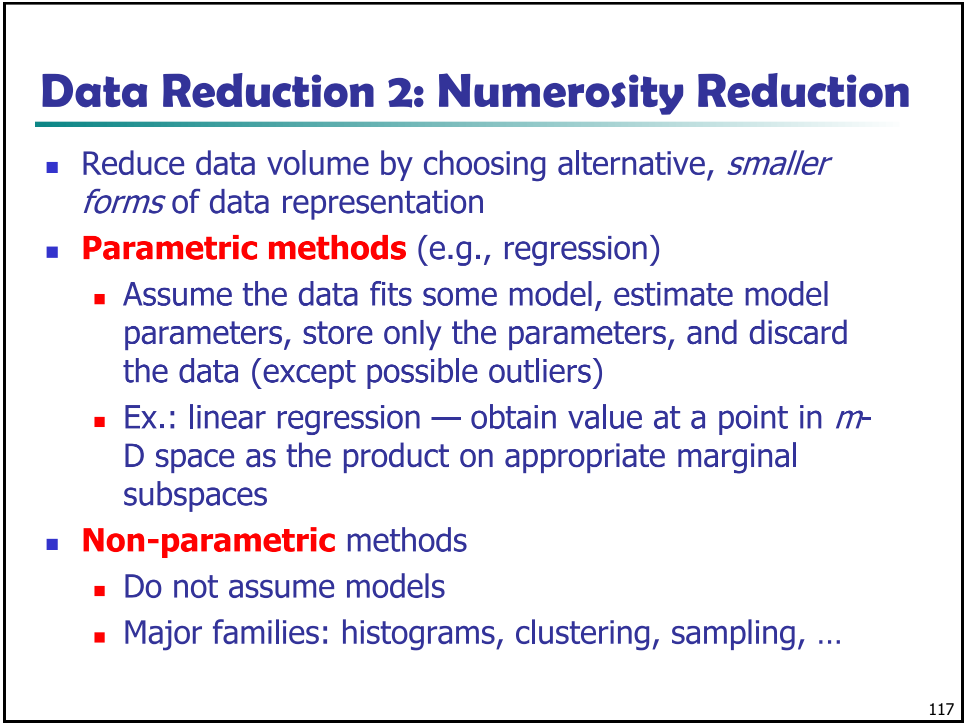
Example of Heuristic Approach

The algorithm

1. *Initialization*: Set $F = \{f_i / i = 1,…,N\}$, initial set of N features, and $S = \{\emptyset\}$, empty set.
2. Calculate the MI with respect to the classes: calculate $I(f_i ; C)$, for each $f_i \in F$.
3. Select the first feature: Find $\widehat{f}_i = \max_{i=1,…,N}\{I(f_i ; C)\}$. Set $F \leftarrow F \setminus \{\widehat{f}_i\}$; set $S \leftarrow \{\widehat{f}_i\}$
4. *Greedy Selection*: Repeat until $|S| = k$.
   a. Calculate the MI between features: Calculate $I(f_i ; f_s)$ for all pairs $(f_i ; f_s)$, with $f_i \in F$ and $f_s \in S$
   b. Select next feature: Select feature $f_i \in F$ that maximizes G. Set $F \leftarrow F \setminus \{\widehat{f}_i\}$; set $S \leftarrow \{\widehat{f}_i\}$.
5. Output the set S containing the selected features.

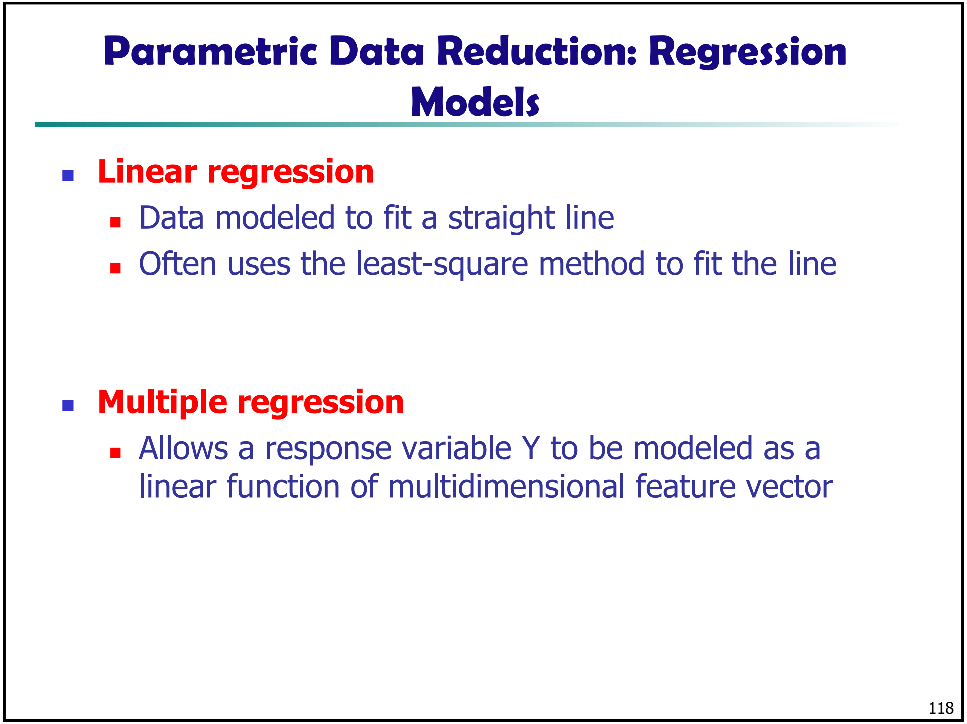
Attribute Creation (Feature Generation)

- 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
    - Data discretization



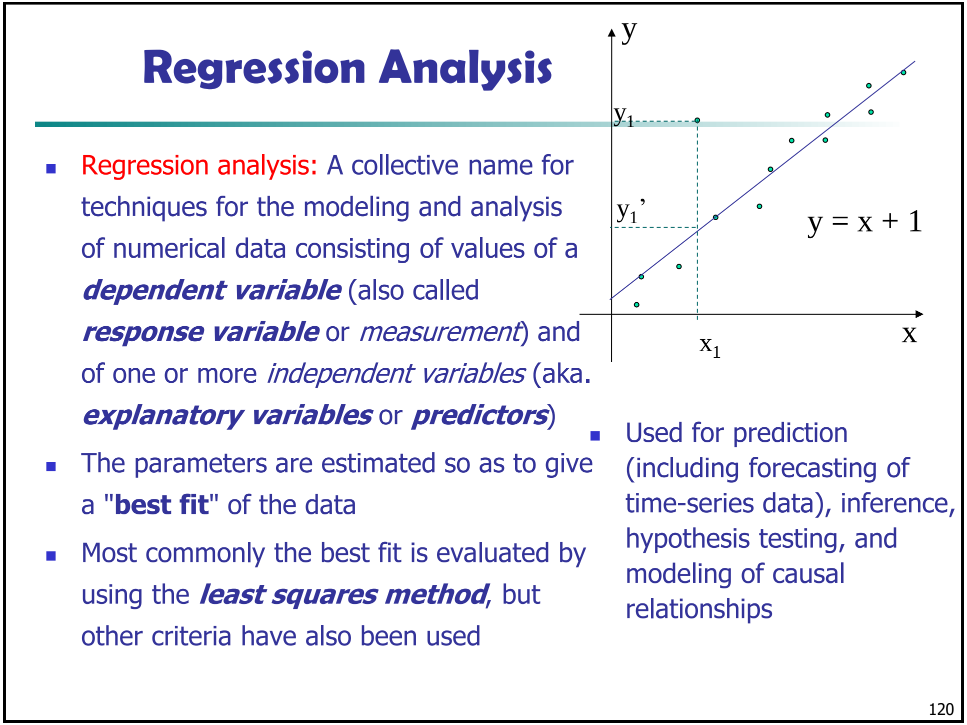
Data Reduction 2: Numerosity Reduction

- 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.: linear regression — 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, …



Parametric Data Reduction: Regression Models

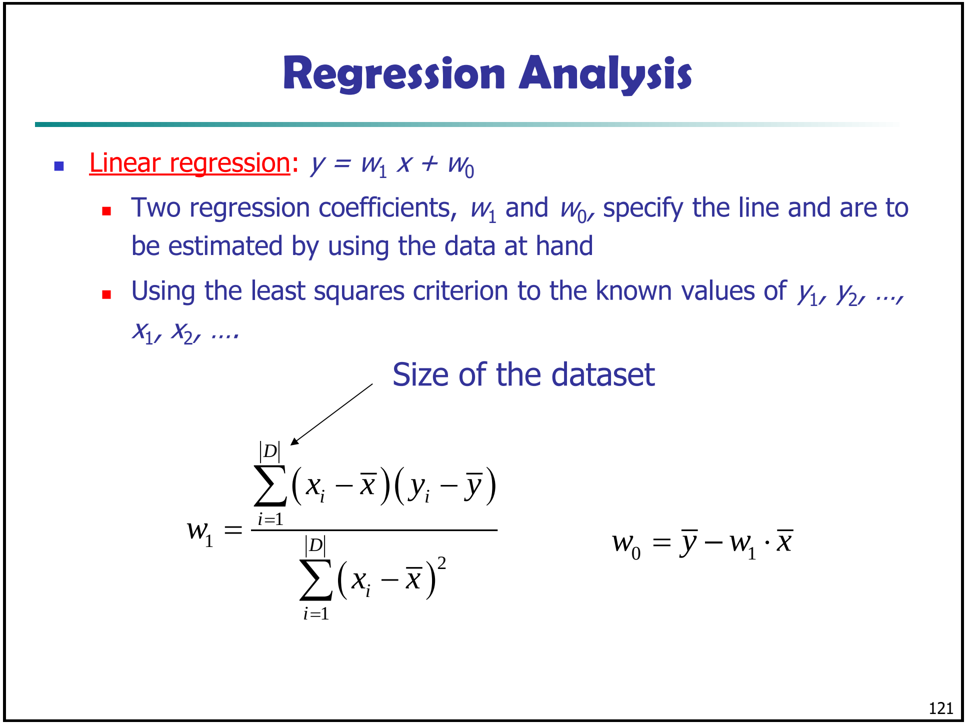
- **Linear regression**
  - Data modeled to fit a straight line
  - Often uses the least-square method to fit the line
- **Multiple regression**
  - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector



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

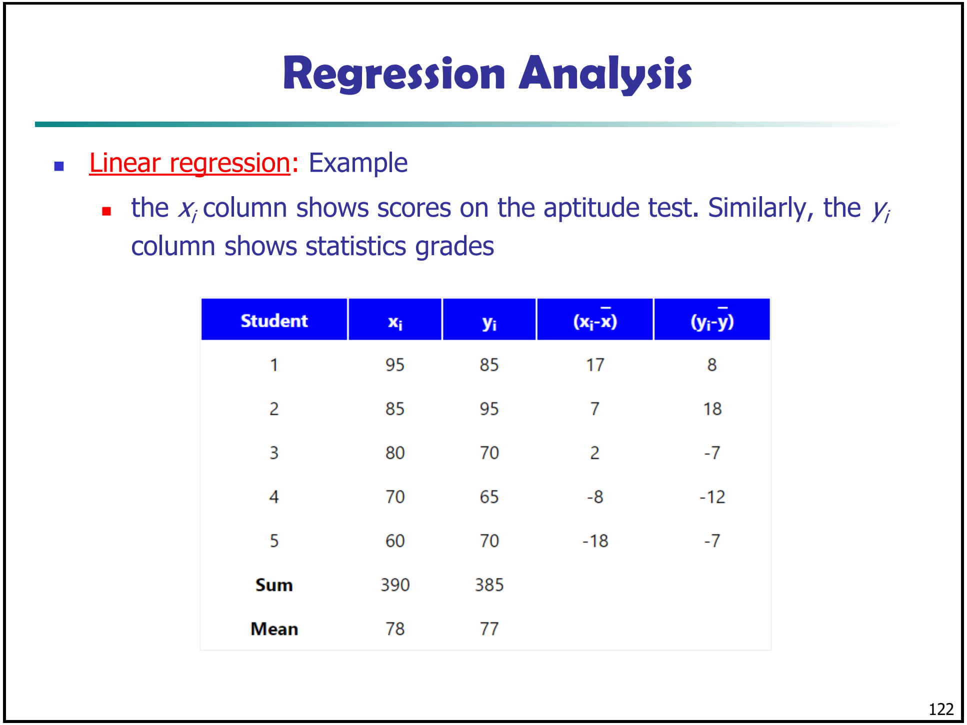


Regression Analysis

- Linear regression: $y = w_1\ x + w_0$
  - Two regression coefficients, w_1 and w_0, specify the line and are to be estimated by using the data at hand
  - Using the least squares criterion to the known values of $y_1, y_2, \ldots, x_1, x_2, \ldots.$

Size of the dataset

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2} \qquad w_0 = \bar{y} - w_1 \cdot \bar{x}$$



Regression Analysis

- Linear regression: Example
  - the x_i column shows scores on the aptitude test. Similarly, the y_i column shows statistics grades

Student	x_i	y_i	$(x_i-\bar{x})$	$(y_i-\bar{y})$
1	95	85	17	8
2	85	95	7	18
3	80	70	2	-7
4	70	65	-8	-12
5	60	70	-18	-7
Sum	390	385		
Mean	78	77		

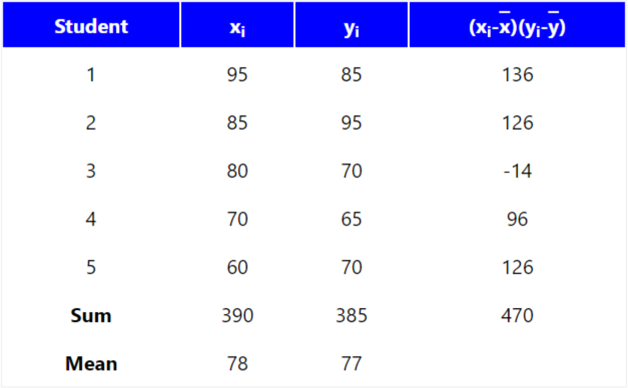
Regression Analysis

- Linear regression: Example
  - Computation of the squares of the deviation scores

Student	x_i	y_i	$(x_i-\bar{x})^2$	$(y_i-\bar{y})^2$
1	95	85	289	64
2	85	95	49	324
3	80	70	4	49
4	70	65	64	144
5	60	70	324	49
Sum	390	385	730	630
Mean	78	77		

Regression Analysis

- Linear regression: Example
  - Computation of the product of the deviation scores



Student	x_i	y_i	$(x_i-\bar{x})(y_i-\bar{y})$
1	95	85	136
2	85	95	126
3	80	70	-14
4	70	65	96
5	60	70	126
Sum	390	385	470
Mean	78	77	

Regression Analysis

- Linear regression: Example
  - Computation of w_1

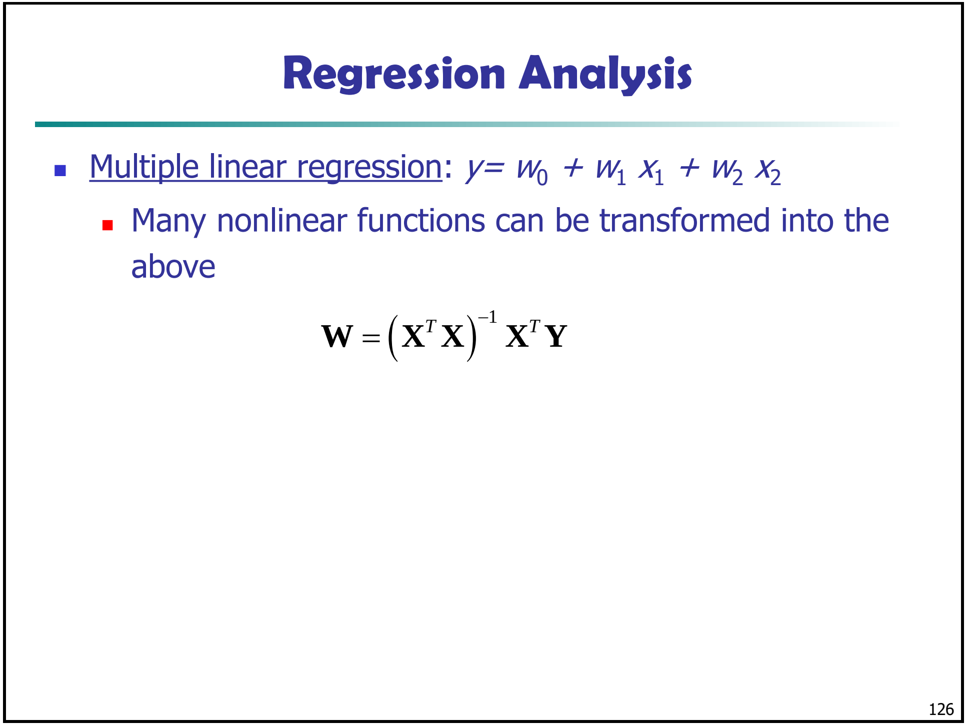
    $w_1 = 470/730 = 0.644$

  - Computation of w_0

    $w_0 = 77 - 0.644 \cdot 78 = 26.768$

  - Linear regression equation

    $y = 0.644x + 26.768$



Regression Analysis

- Multiple linear regression: $y = w_0 + w_1 x_1 + w_2 x_2$
  - Many nonlinear functions can be transformed into the above

$$\mathbf{W} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{Y}$$

Non-linear Regression Analysis

- Polynomial regression: $y = w_0 + w_1\, x + w_2\, x^2 + w_3\, x^3$
  - To convert this equation to linear form, we apply the following transformation
    - $x_1=x$
    - $x_2=x^2$
    - $x_3=x^3$

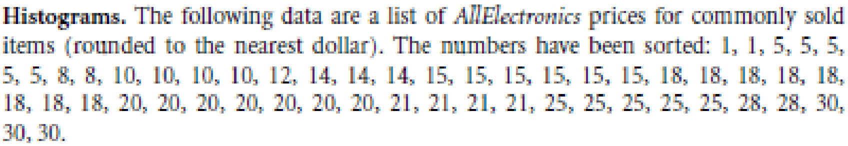
  Thus, the equation becomes

  $y= w_0 + w_1\, x_1 + w_2\, x_2 + w_3\, x_3$

  which can be solved by using the methods for multiple regression analysis.

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)



Histograms. The following data are a list of *AllElectronics* prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

Histogram Analysis

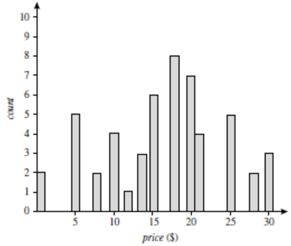


Figure 3.7 A histogram for *price* using singleton buckets—each bucket represents one price–value/ frequency pair.

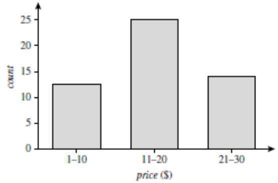
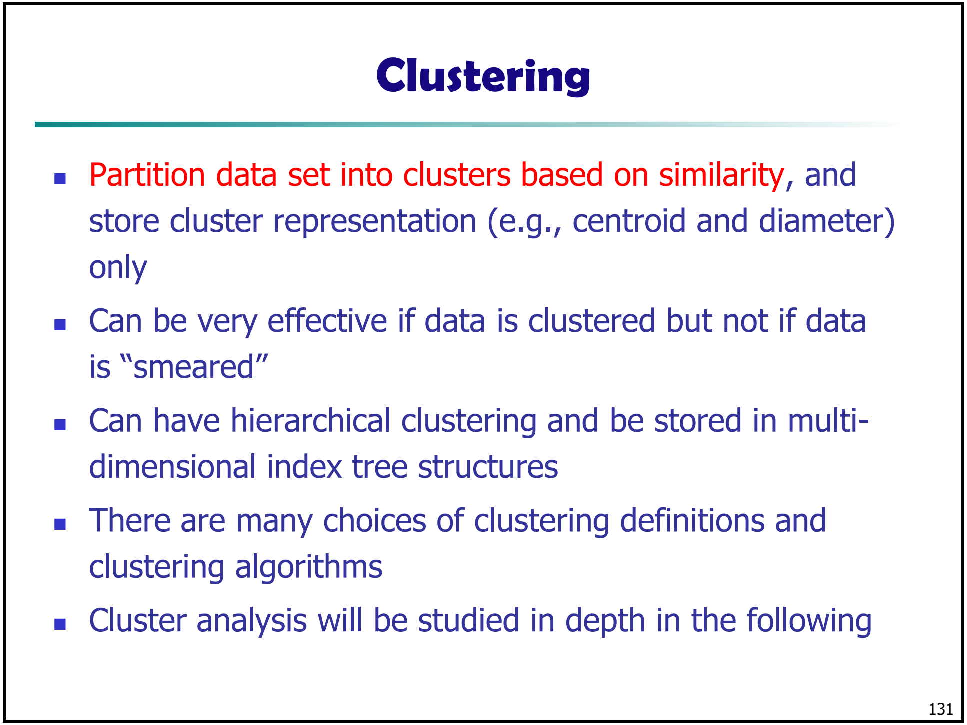


Figure 3.8 An equal-width histogram for *price*, where values are aggregated so that each bucket has a uniform width of $10.

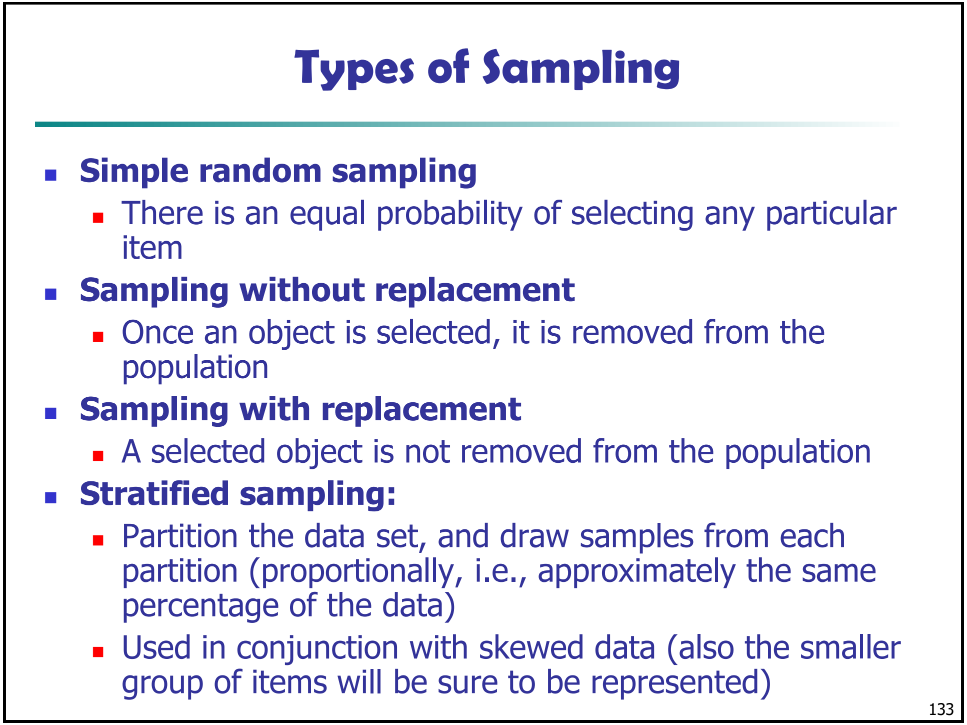


Clustering

- 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 the following

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)

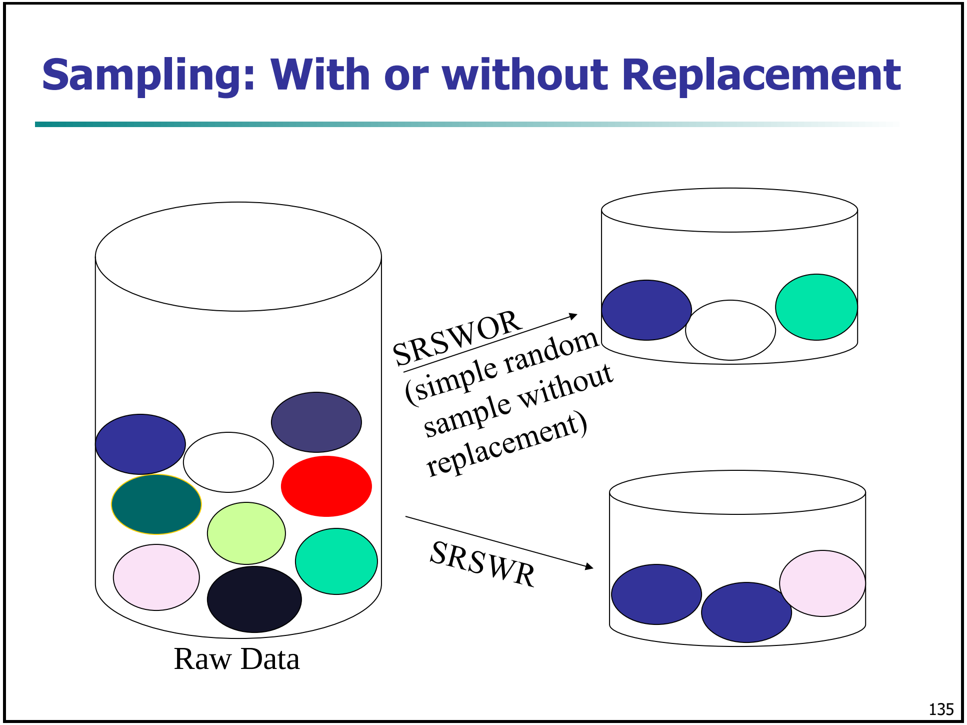


Types of Sampling

- **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 (also the smaller group of items will be sure to be represented)

Types of Sampling

- **Sampling with and without replacement. What is the difference**
  - When we sample with replacement, the two sample values are independent.
    - what we get on the first one doesn't affect what we get on the second.
    - The covariance between the two is zero.
  - In sampling without replacement, the two sample values aren't independent.
    - what we got on for the first one affects what we can get for the second one.
    - The covariance between the two is $\frac{-\sigma^2}{N-1}$ from a population with variance σ^2



Sampling: With or without Replacement

SRSWOR (simple random sample without replacement)

SRSWR

Raw Data

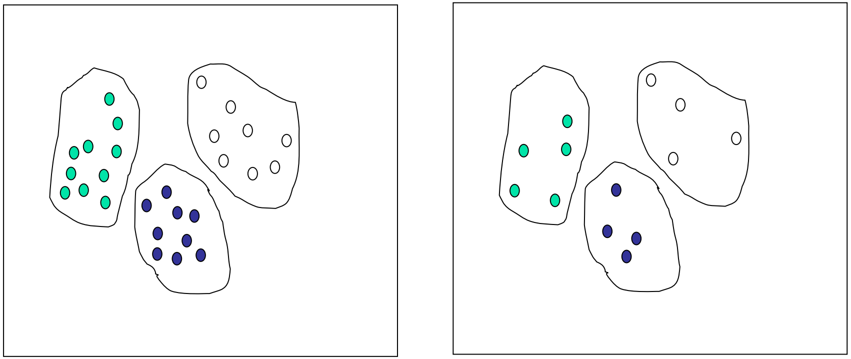
135

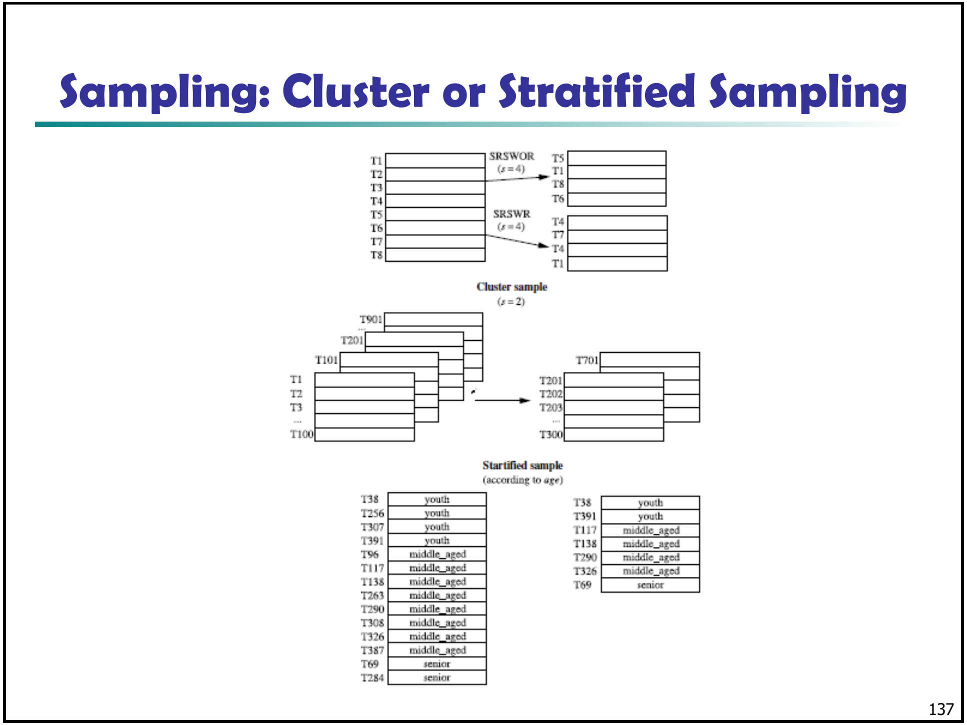
135

Sampling: Cluster or Stratified Sampling

Raw Data

Cluster/Stratified Sample





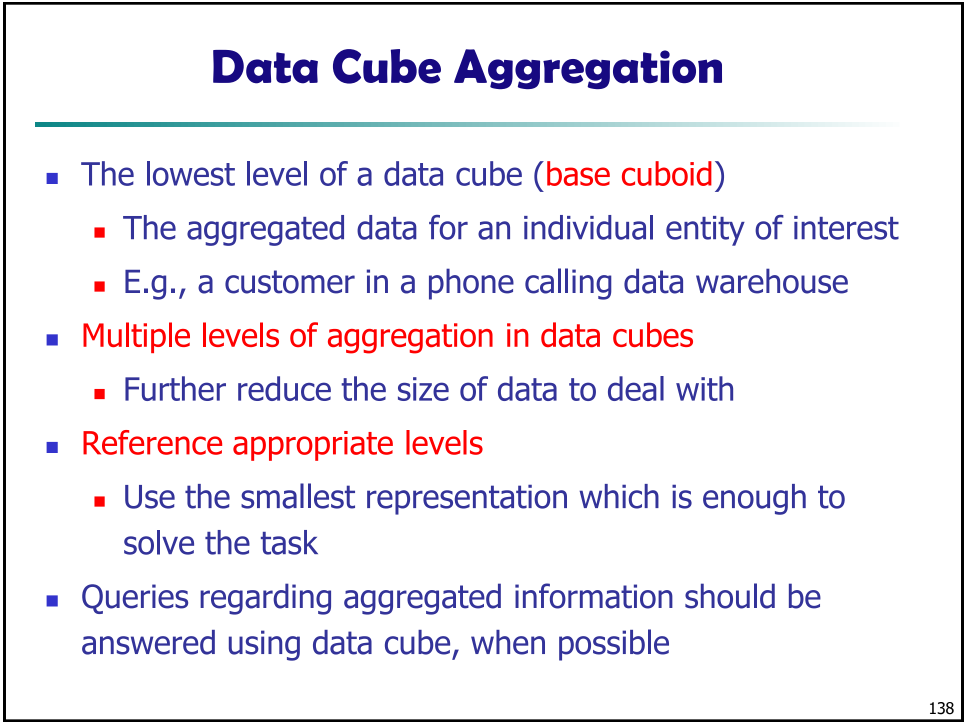
Sampling: Cluster or Stratified Sampling

Cluster sample
(*s* = 2)

Startified sample
(according to *age*)

T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

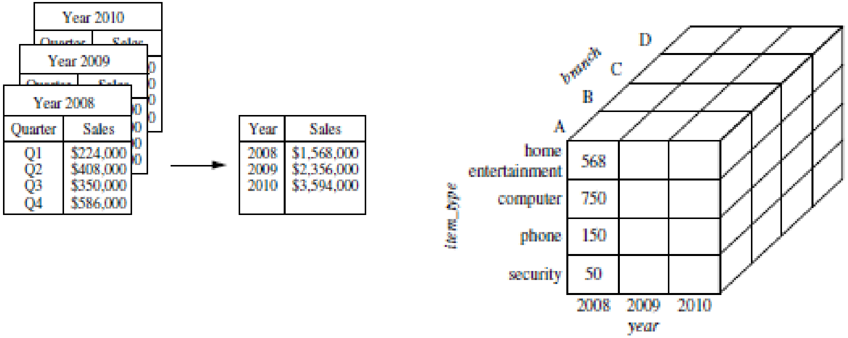
T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior



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 Cube Aggregation



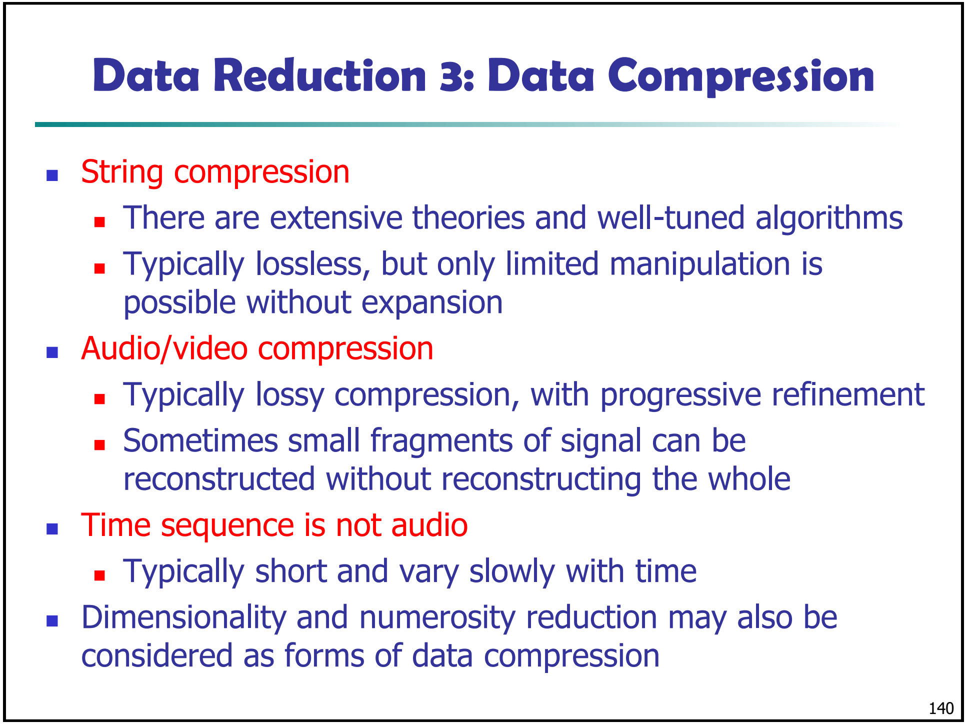
Year 2010

Year 2009

Year 2008

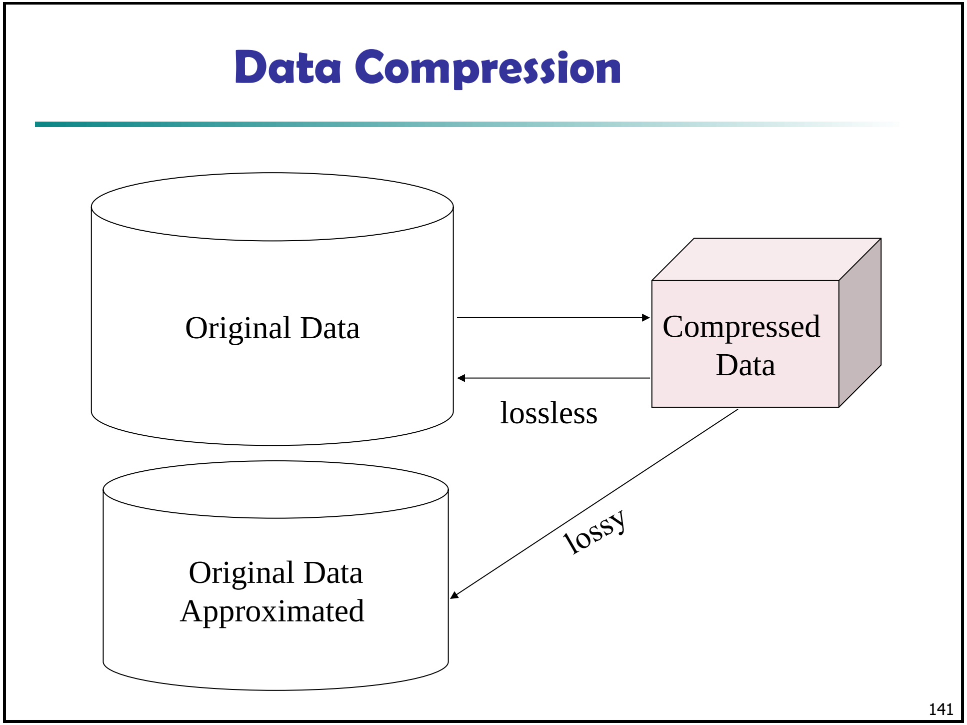
Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

Year	Sales
2008	$1,568,000
2009	$2,356,000
2010	$3,594,000



Data Reduction 3: Data Compression

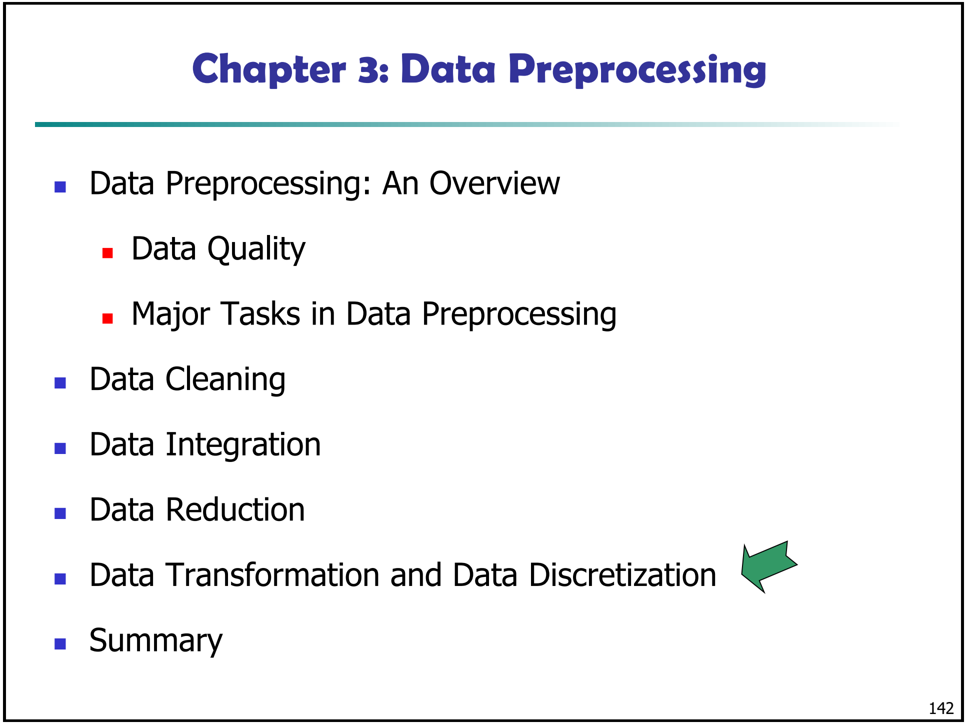
- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless, but only limited manipulation is possible without expansion
- Audio/video compression
  - Typically lossy compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed 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

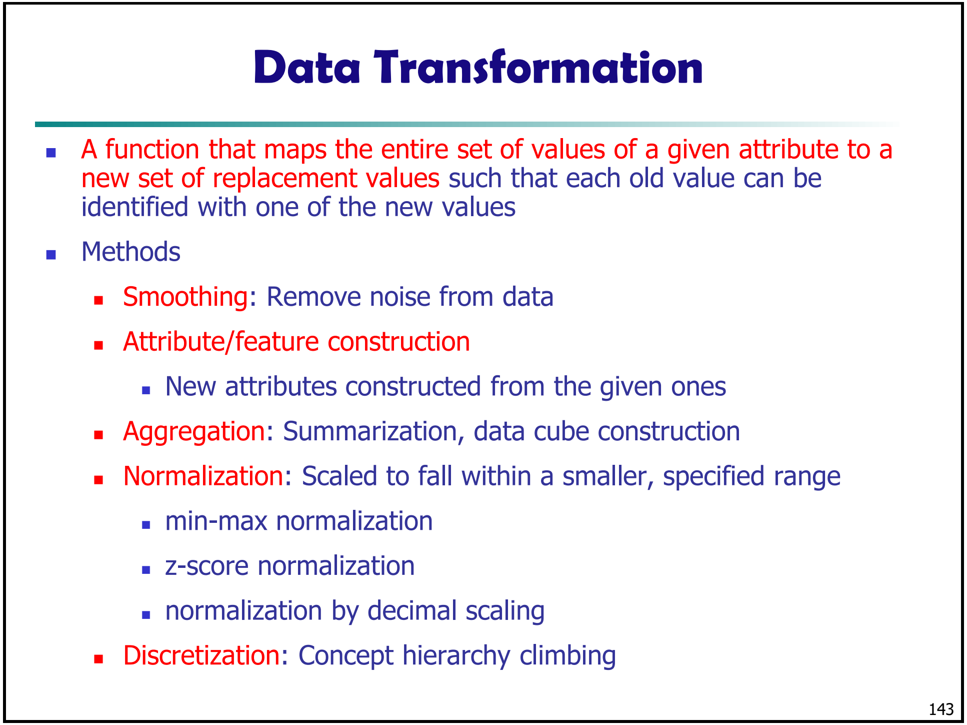
141

141



Chapter 3: 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 Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values such that 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$_A$]

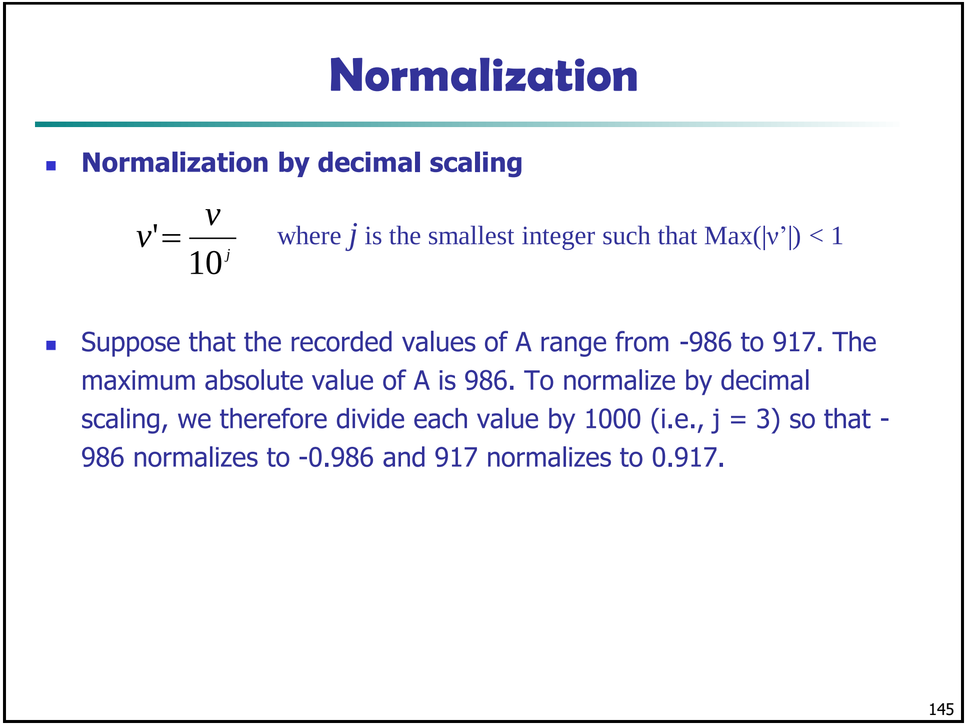
$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,600 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 μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

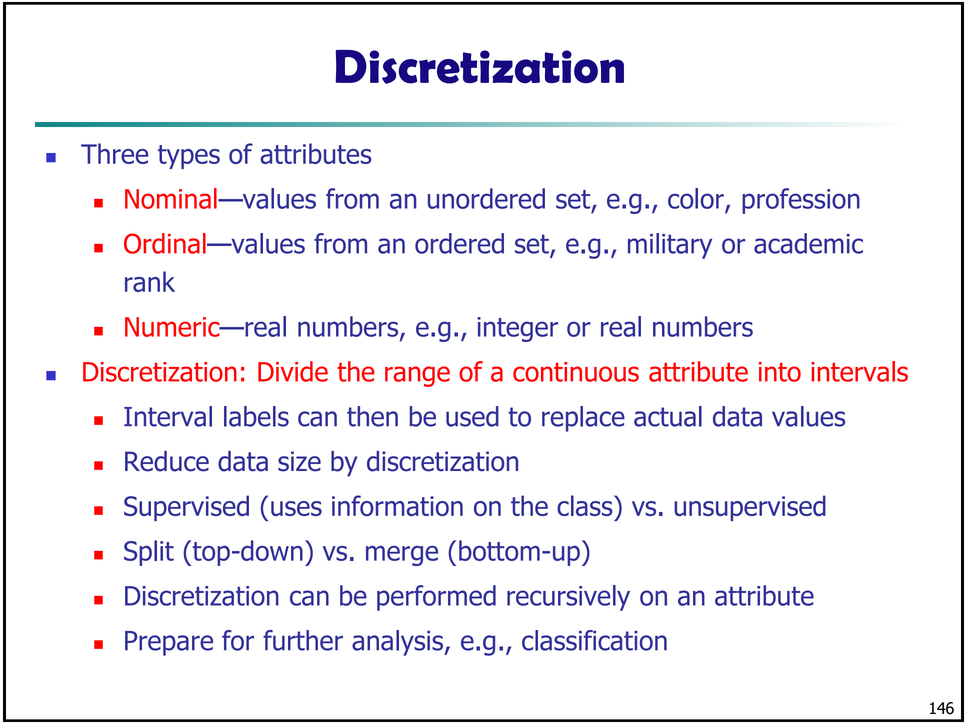


Normalization

- **Normalization by decimal scaling**

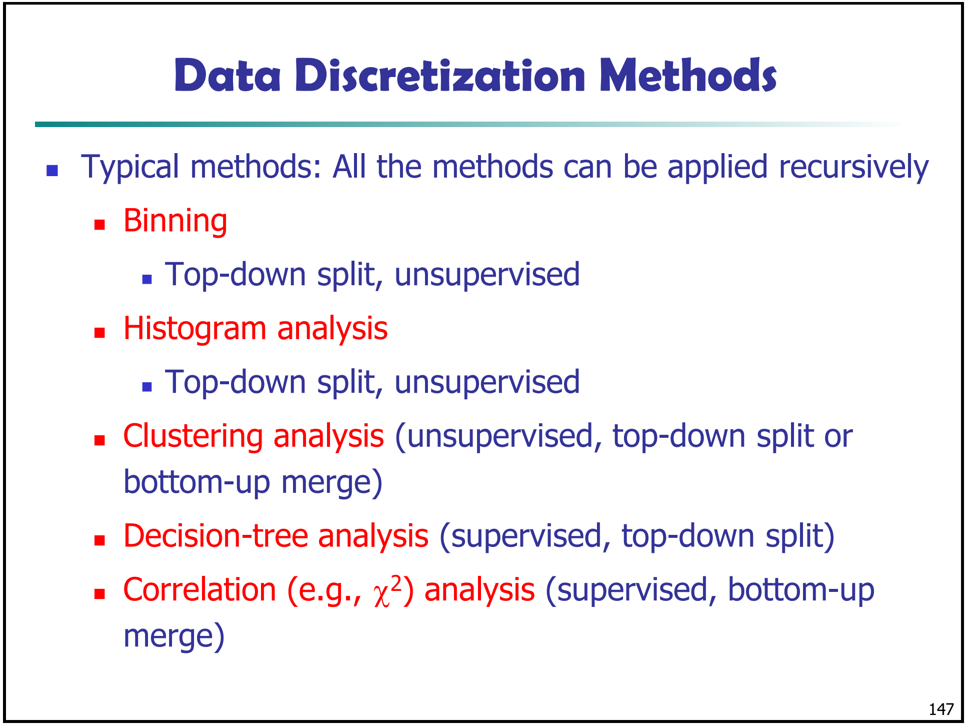
$$v' = \frac{v}{10^j}$$ where j is the smallest integer such that Max(|v’|) < 1

- Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling, we therefore divide each value by 1000 (i.e., j = 3) so that -986 normalizes to -0.986 and 917 normalizes to 0.917.



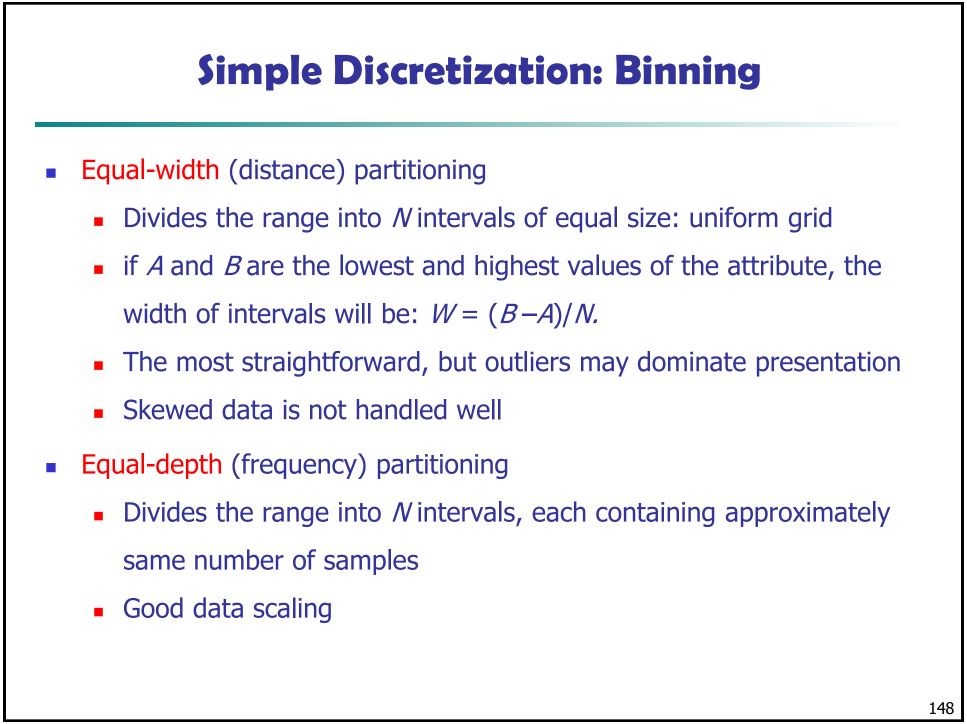
Discretization

- 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 (uses information on the class) vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification



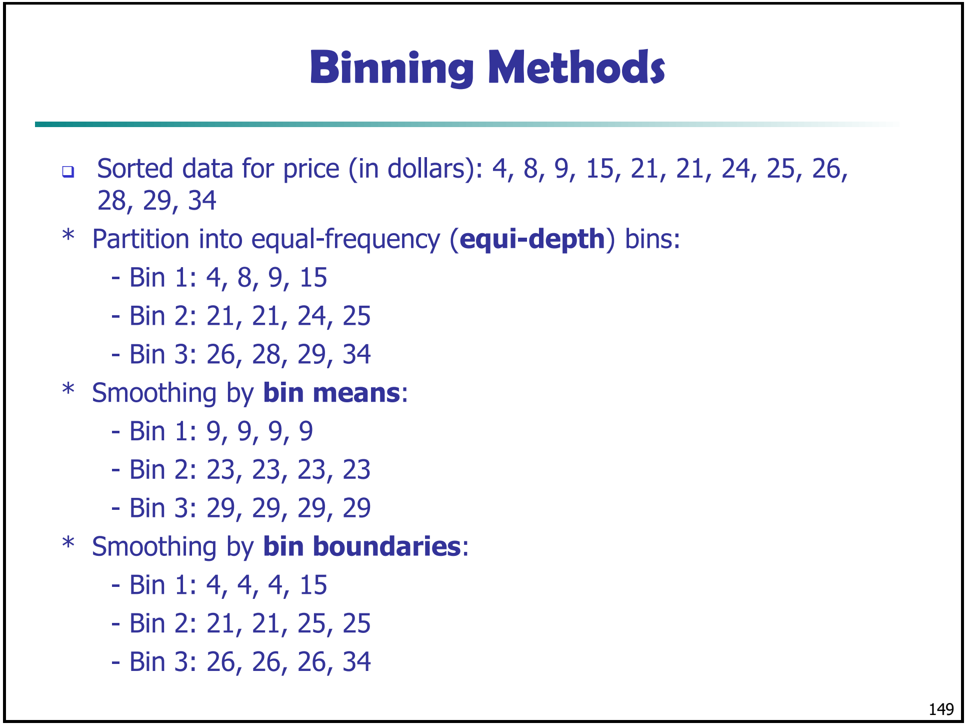
Data Discretization Methods

- 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., χ^2) analysis (supervised, 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



Binning Methods

- 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

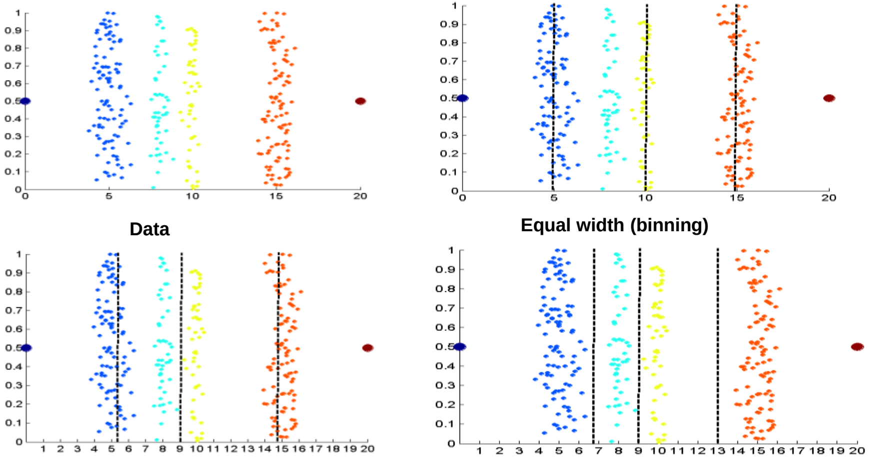
* Smoothing by **bin means**:

- 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)

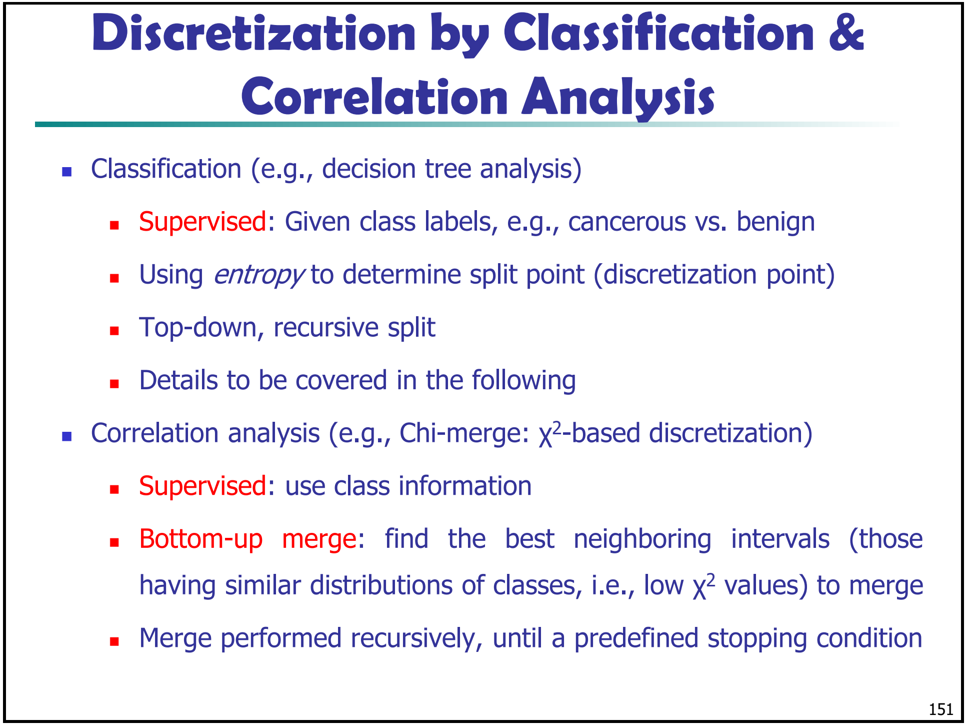


Data

Equal width (binning)

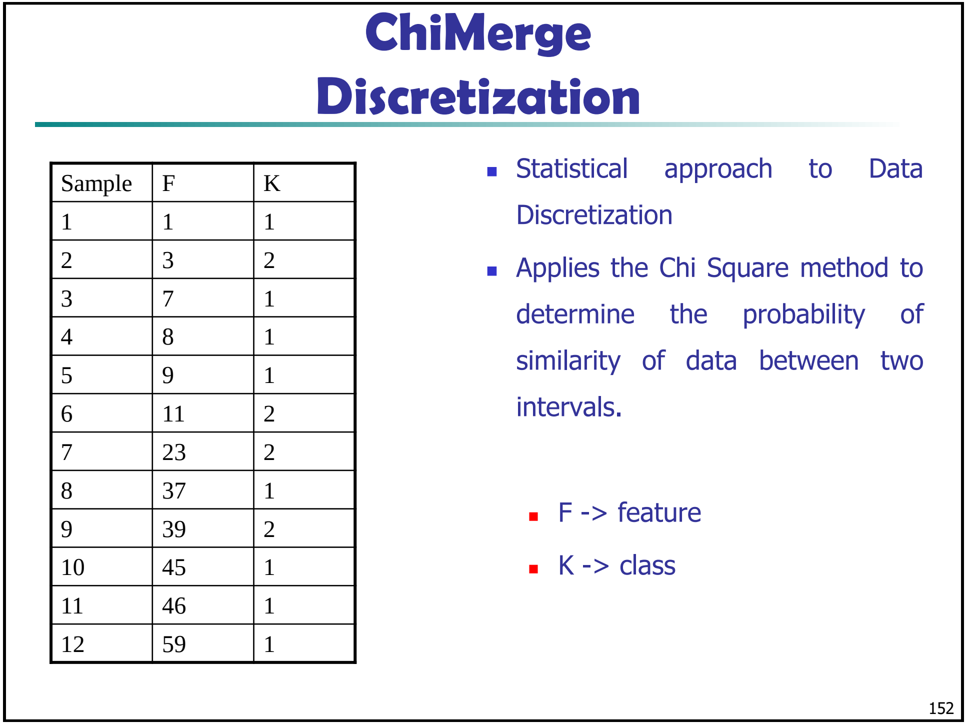
Equal frequency (binning)

K-means clustering leads to better results



Discretization by Classification & Correlation Analysis

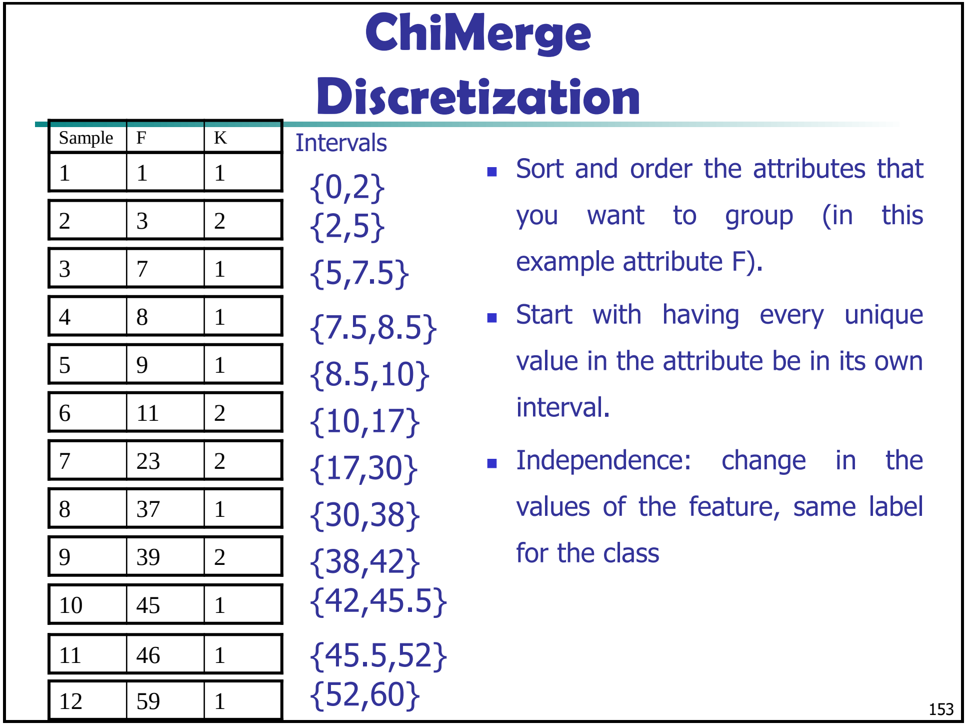
- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using *entropy* to determine split point (discretization point)
  - Top-down, recursive split
  - Details to be covered in the following
- Correlation analysis (e.g., Chi-merge: χ^2-based 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



ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

- Statistical approach to Data Discretization
- Applies the Chi Square method to determine the probability of similarity of data between two intervals.
  - F -> feature
  - K -> class



ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Intervals

{0,2}
{2,5}
{5,7.5}
{7.5,8.5}
{8.5,10}
{10,17}
{17,30}
{30,38}
{38,42}
{42,45.5}
{45.5,52}
{52,60}

- Sort and order the attributes that you want to group (in this example attribute F).
- Start with having every unique value in the attribute be in its own interval.
- Independence: change in the values of the feature, same label for the class

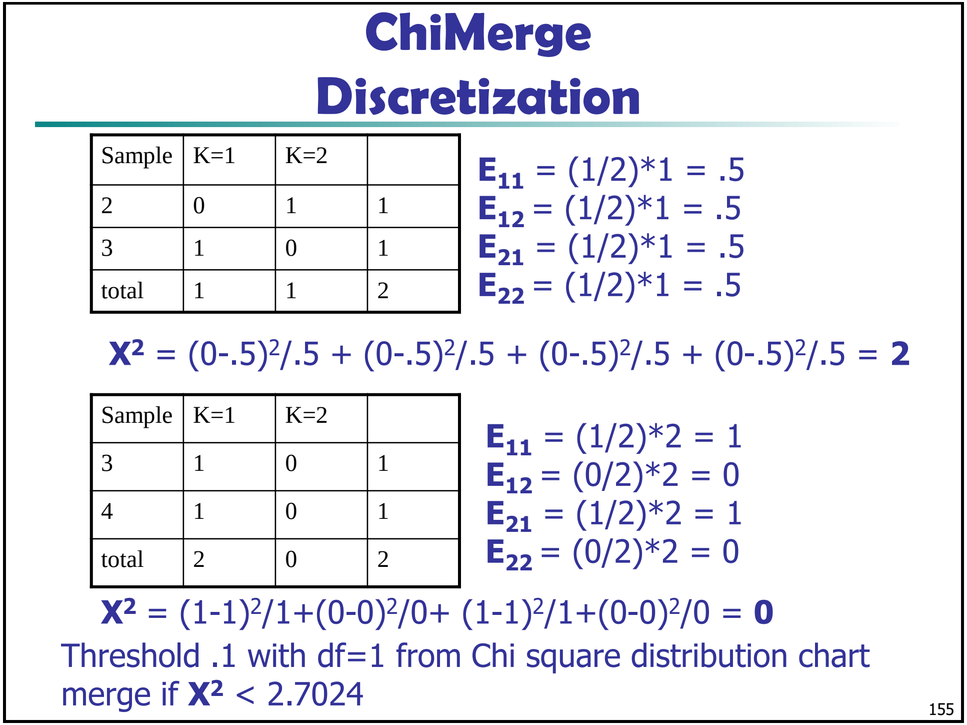
ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

- Begin calculating the Chi Square test on every interval

Sample	K=1	K=2	
2	0	1	1
3	1	0	1
total	1	1	2

Sample	K=1	K=2	
3	1	0	1
4	1	0	1
total	2	0	2



ChiMerge Discretization

Sample	K=1	K=2	
2	0	1	1
3	1	0	1
total	1	1	2

$\mathbf{E_{11}} = (1/2)*1 = .5$
$\mathbf{E_{12}} = (1/2)*1 = .5$
$\mathbf{E_{21}} = (1/2)*1 = .5$
$\mathbf{E_{22}} = (1/2)*1 = .5$

$$\mathbf{X^2} = (0-.5)^2/.5 + (0-.5)^2/.5 + (0-.5)^2/.5 + (0-.5)^2/.5 = \mathbf{2}$$

Sample	K=1	K=2	
3	1	0	1
4	1	0	1
total	2	0	2

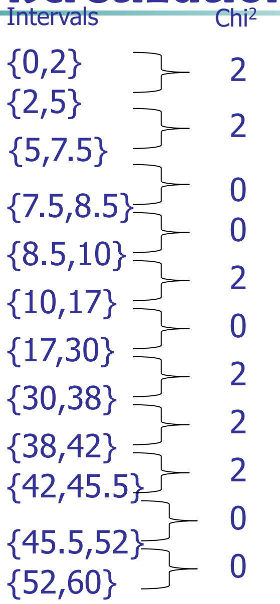
$\mathbf{E_{11}} = (1/2)*2 = 1$
$\mathbf{E_{12}} = (0/2)*2 = 0$
$\mathbf{E_{21}} = (1/2)*2 = 1$
$\mathbf{E_{22}} = (0/2)*2 = 0$

$$\mathbf{X^2} = (1-1)^2/1+(0-0)^2/0+ (1-1)^2/1+(0-0)^2/0 = \mathbf{0}$$

Threshold .1 with df=1 from Chi square distribution chart
merge if $\mathbf{X^2} < 2.7024$

ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1



Intervals

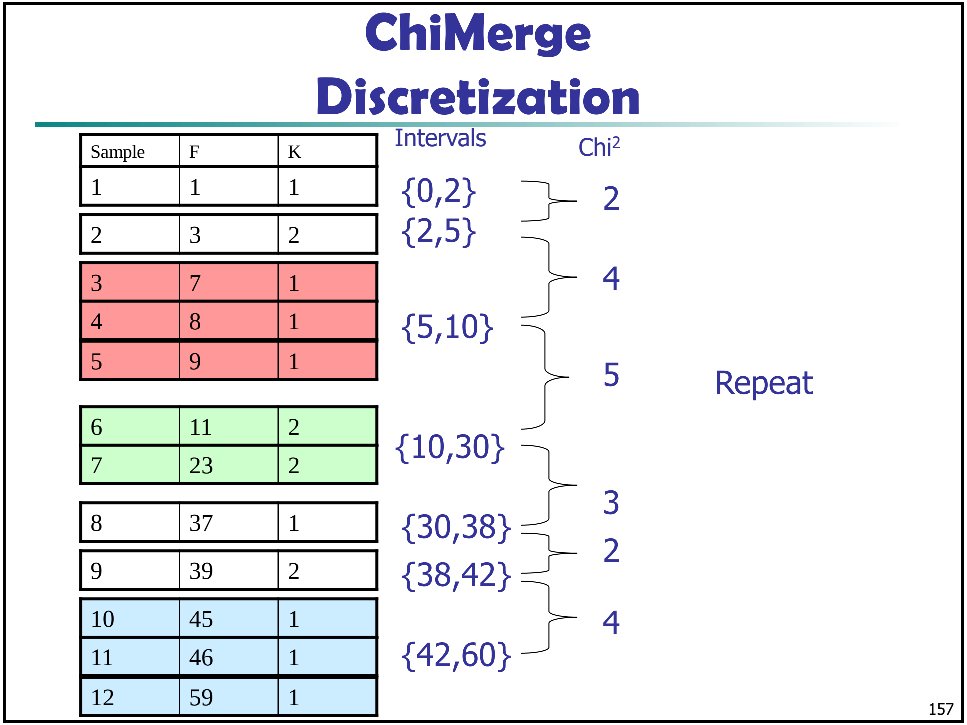
{0,2}
{2,5}
{5,7.5}
{7.5,8.5}
{8.5,10}
{10,17}
{17,30}
{30,38}
{38,42}
{42,45.5}
{45.5,52}
{52,60}

Chi^2

2
2
0
0
2
0
2
2
2
0
0

Calculate all the Chi Square value for all intervals

Merge the intervals with the smallest Chi values



ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Intervals

{0,2}

{2,5}

{5,10}

{10,30}

{30,38}

{38,42}

{42,60}

Chi^2

2

4

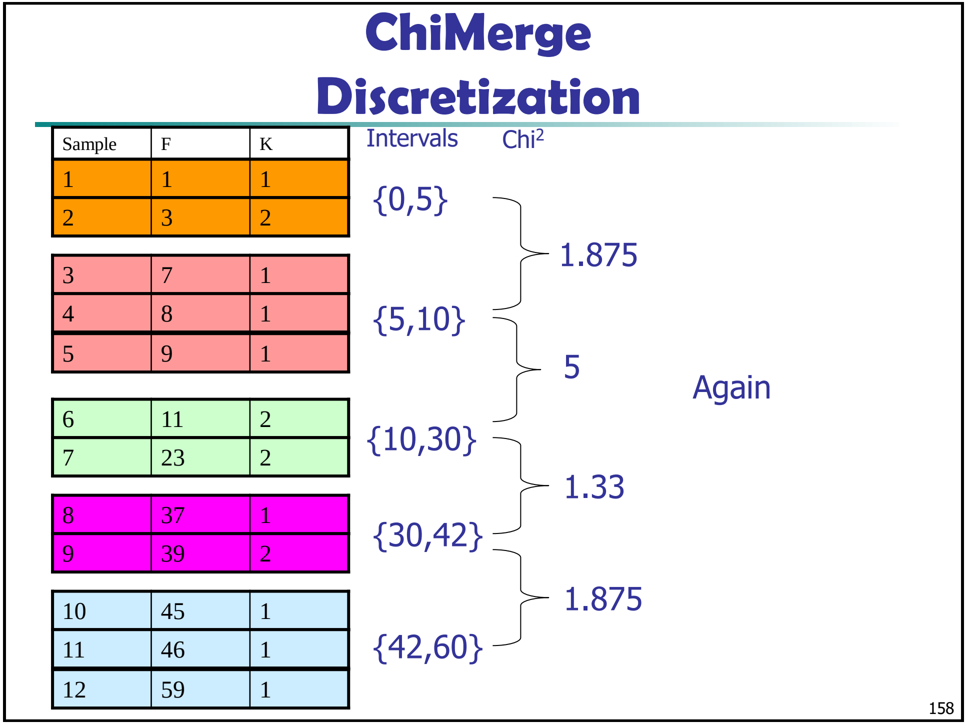
5

3

2

4

Repeat



ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Intervals Chi^2

{0,5}

1.875

{5,10}

5

Again

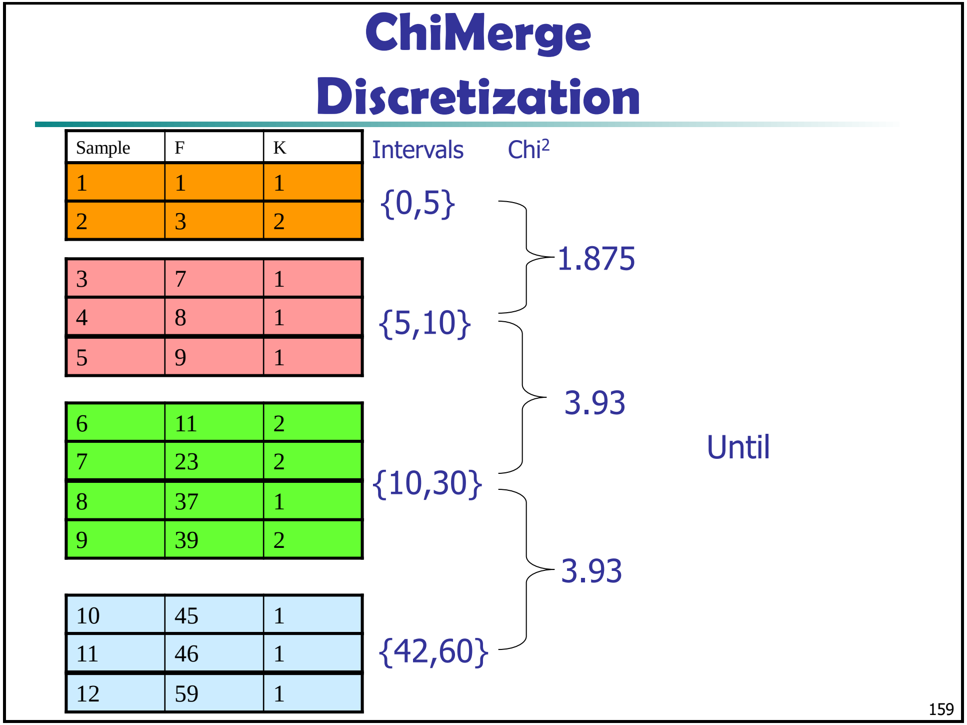
{10,30}

1.33

{30,42}

1.875

{42,60}



ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Intervals Chi^2

{0,5}

1.875

{5,10}

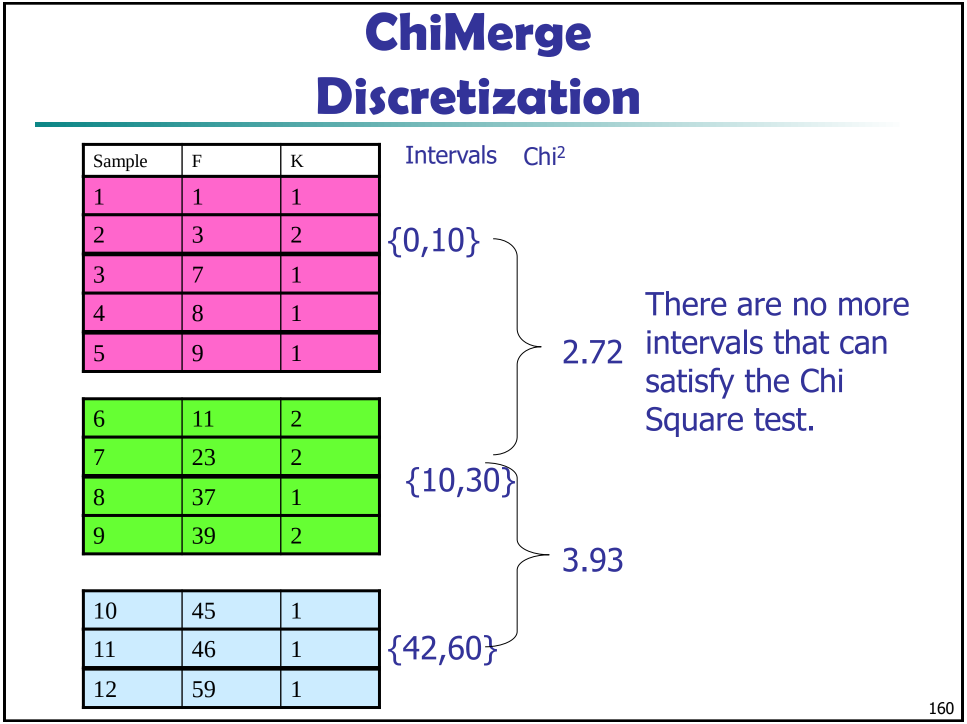
3.93

{10,30}

3.93

{42,60}

Until

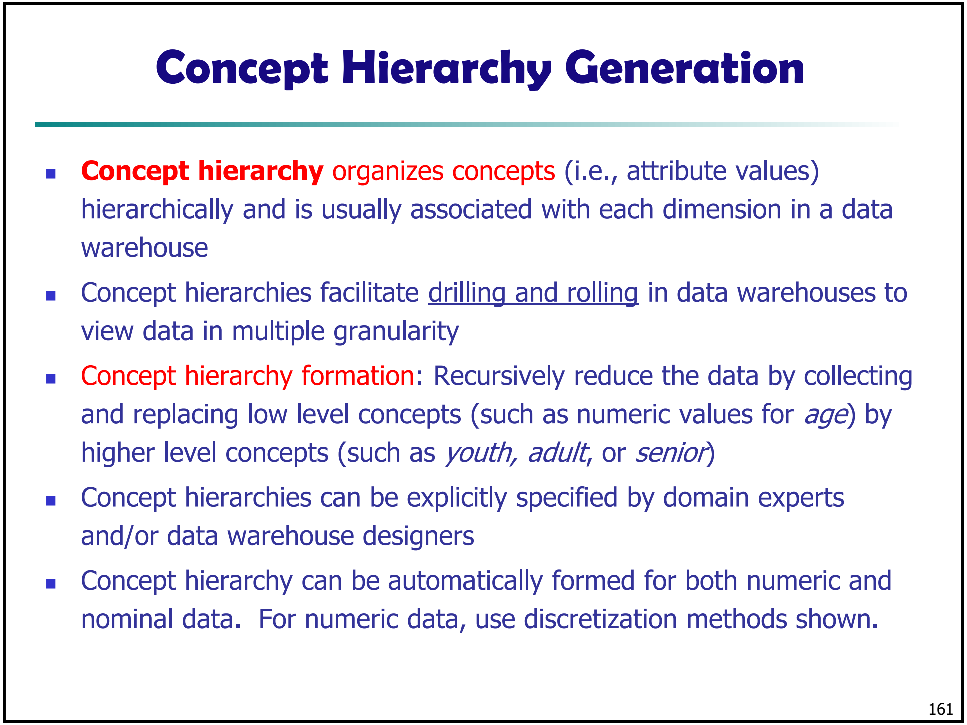


ChiMerge Discretization

Sample	F	K
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

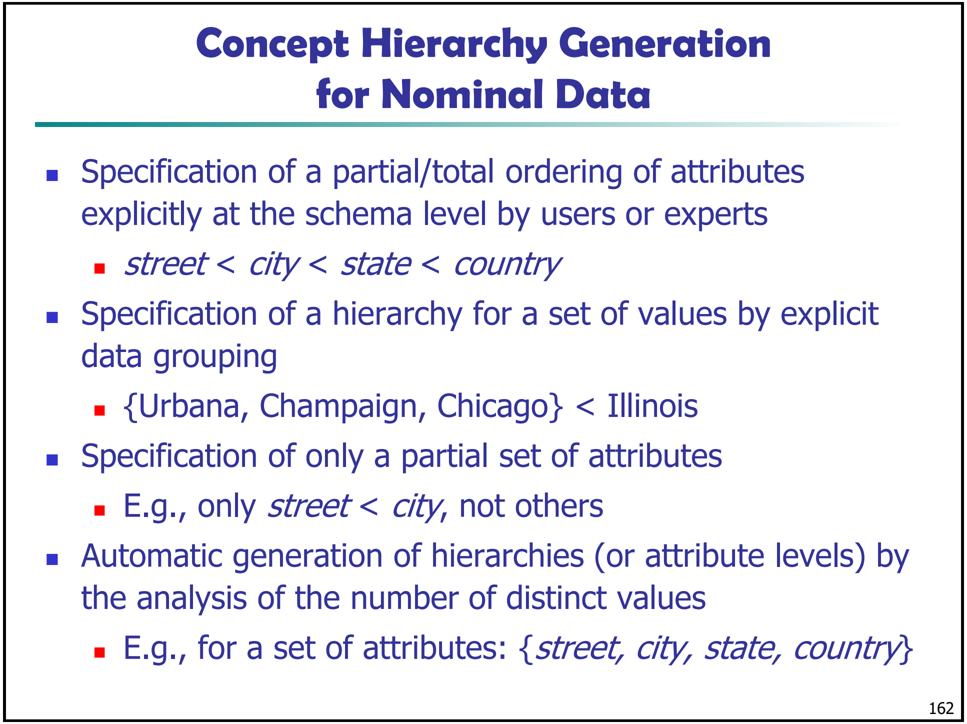
Intervals	Chi^2
{0,10}	
	2.72
{10,30}	
	3.93
{42,60}	

There are no more intervals that can satisfy the Chi Square test.



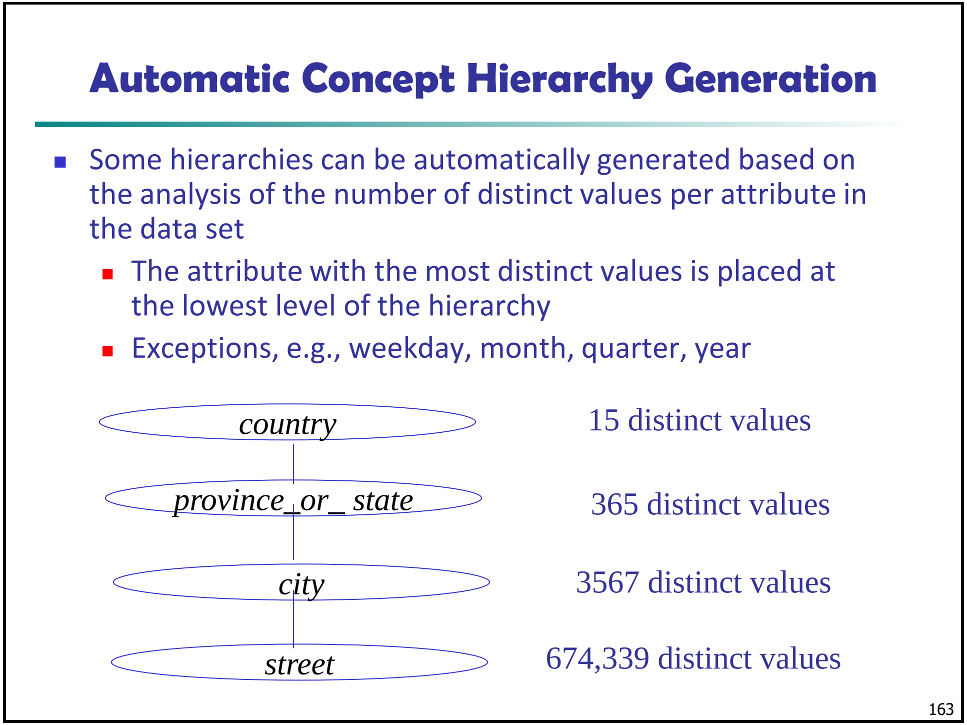
Concept Hierarchy Generation

- **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

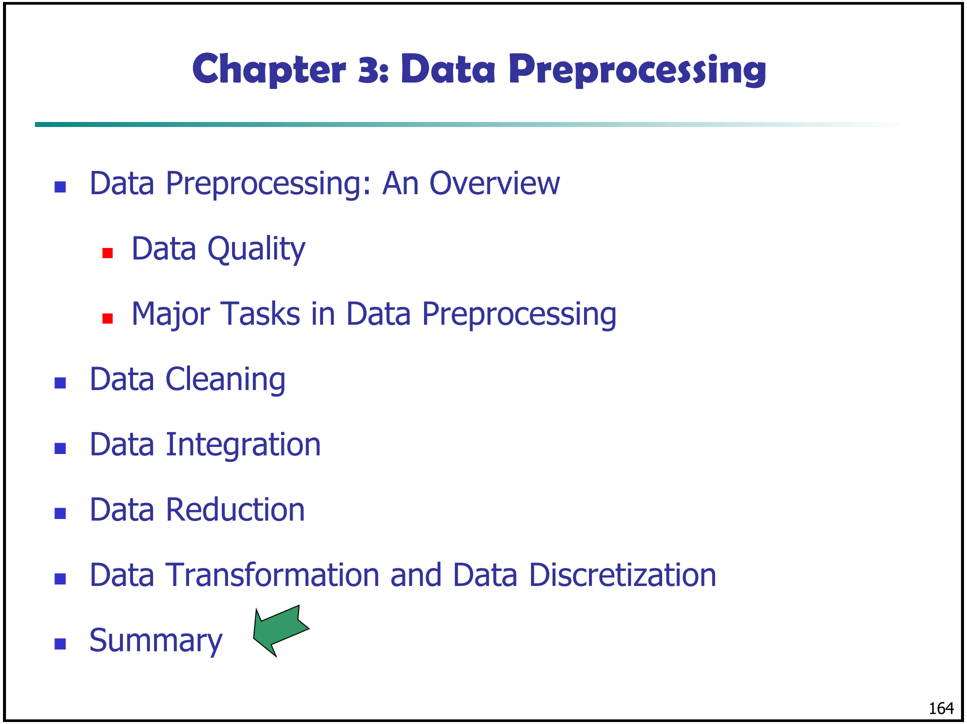
- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Exceptions, e.g., weekday, month, quarter, year

country 15 distinct values

province_or_ state 365 distinct values

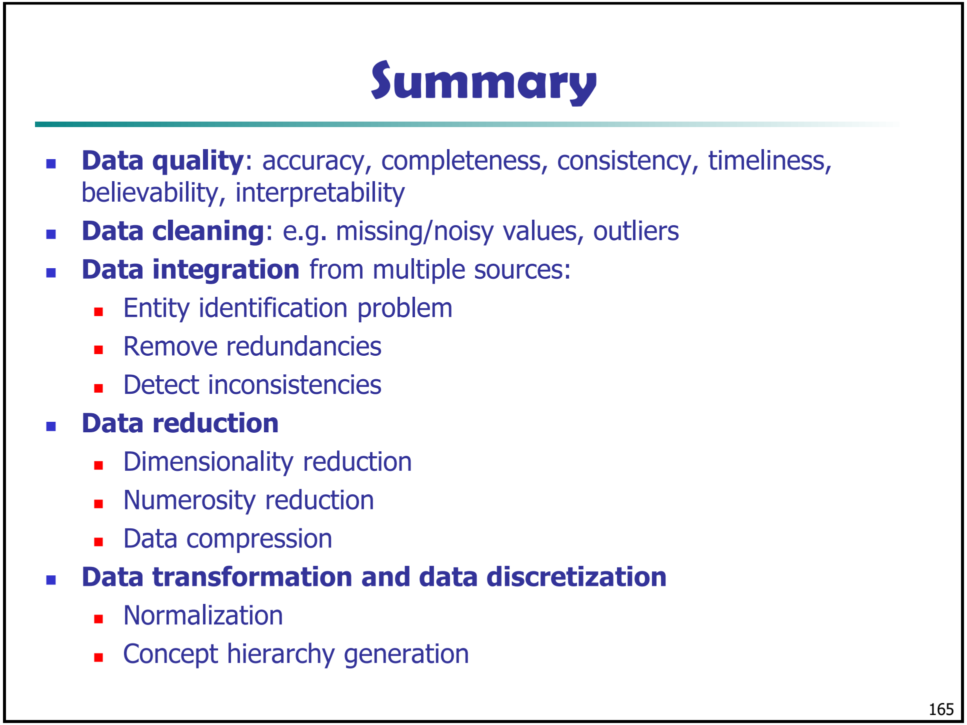
city 3567 distinct values

street 674,339 distinct values



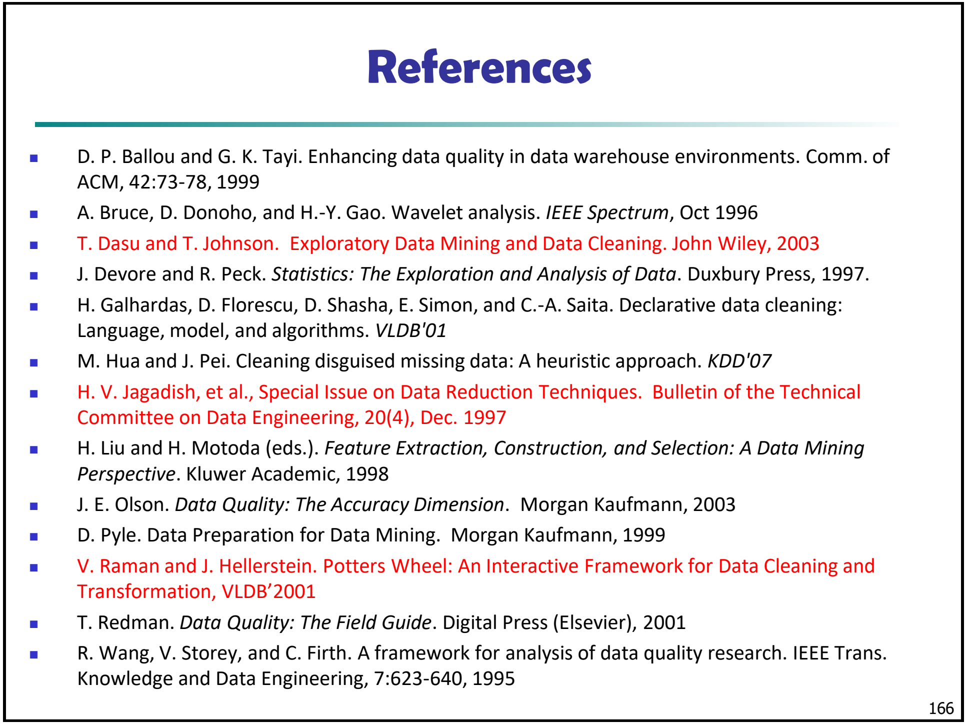
Chapter 3: Data Preprocessing

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



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



References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Comm. of ACM, 42:73-78, 1999
- A. Bruce, D. Donoho, and H.-Y. Gao. Wavelet analysis. *IEEE Spectrum*, Oct 1996
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- J. Devore and R. Peck. *Statistics: The Exploration and Analysis of Data*. Duxbury Press, 1997.
- H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. *VLDB'01*
- M. Hua and J. Pei. Cleaning disguised missing data: A heuristic approach. *KDD'07*
- H. V. Jagadish, et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), Dec. 1997
- H. Liu and H. Motoda (eds.). *Feature Extraction, Construction, and Selection: A Data Mining Perspective*. Kluwer Academic, 1998
- J. E. Olson. *Data Quality: The Accuracy Dimension*. Morgan Kaufmann, 2003
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB’2001
- T. Redman. *Data Quality: The Field Guide*. Digital Press (Elsevier), 2001
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995