

Applied Data Science Data Preprocessing

Vaibhav P. Vasani

Assistant Professor

Department of Computer Engineering

K. J. Somaiya College of Engineering

Somaiya Vidyavihar University

Data is: Big!

Analysis => Lots of Jobs


Lots of Data => Lots of

- 2.5 quintillion (10¹⁸) bytes of data are generated every day
- More data is publicly available
- Database technology is advancing
- Cloud based & mobile applications are widespread

how do we manage all this data?



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
- **Data reduction**
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression
- **Data transformation and data discretization**
 - Normalization
 - Concept hierarchy generation

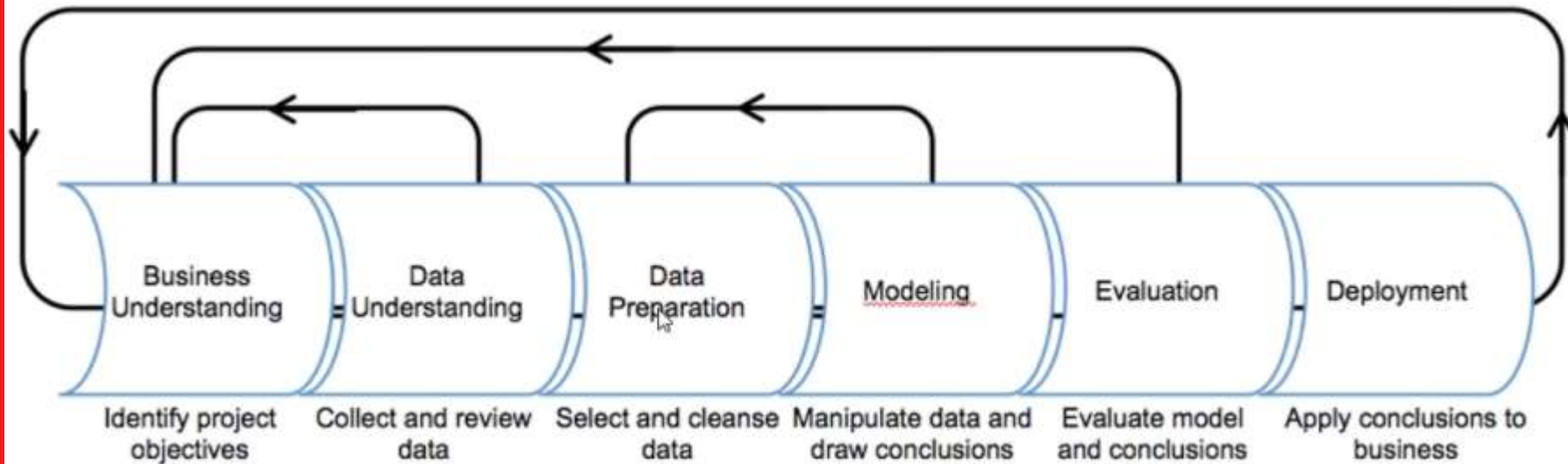
Chapter 3: Data Preprocessing

- 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 and Transformation process

- Iterative process



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

Missing Values

Col1	Col2	Col3	Col4	Col5
12456	0.99	Male	43	Small
98567	1.23		55	Medium
34567	9999	Female	NA	Large
67231	0.72	Male	35	?

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



Treating Missing Values

- Remove rows
- Substitute a specific values
- Interpolate values
- Forward fill
- Backward fill
- Impute



Note

- Many ML algorithm don't deal with missing values

Note

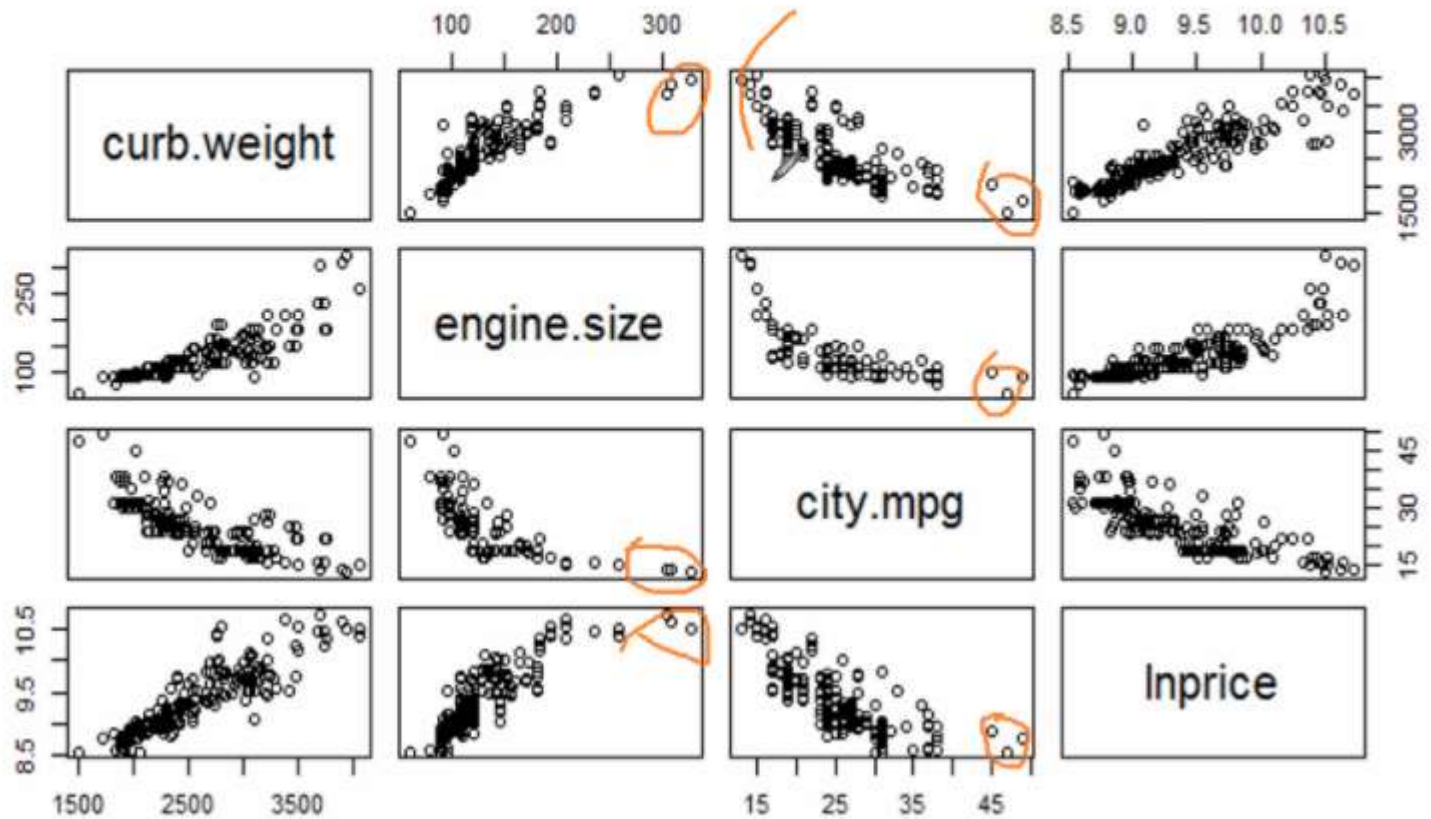
- Repeated Values bias results



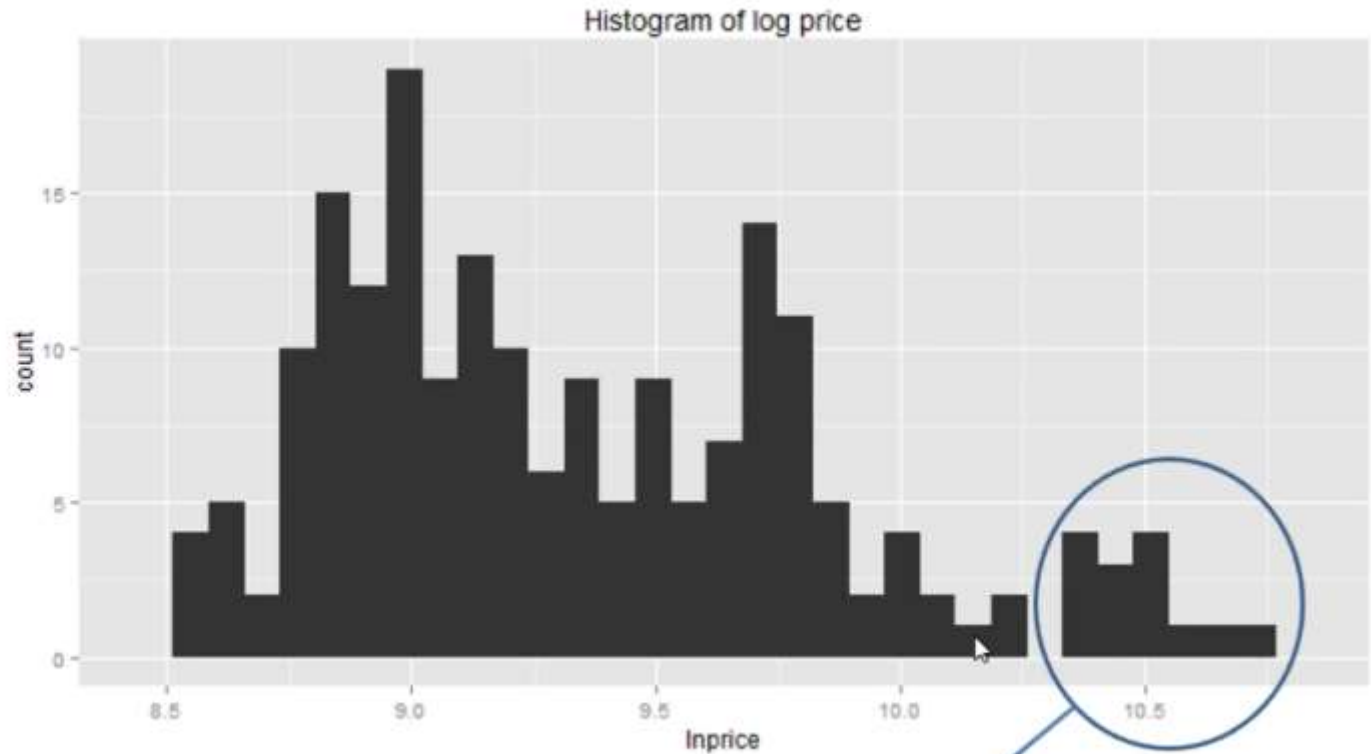
Treating Outlier and Error

- Error and Outliers can bias model training
- Many Possible Sources of errors
 - Erroneous measurement
 - Entry errors
 - Transposed values in table

Visualizing Outliers

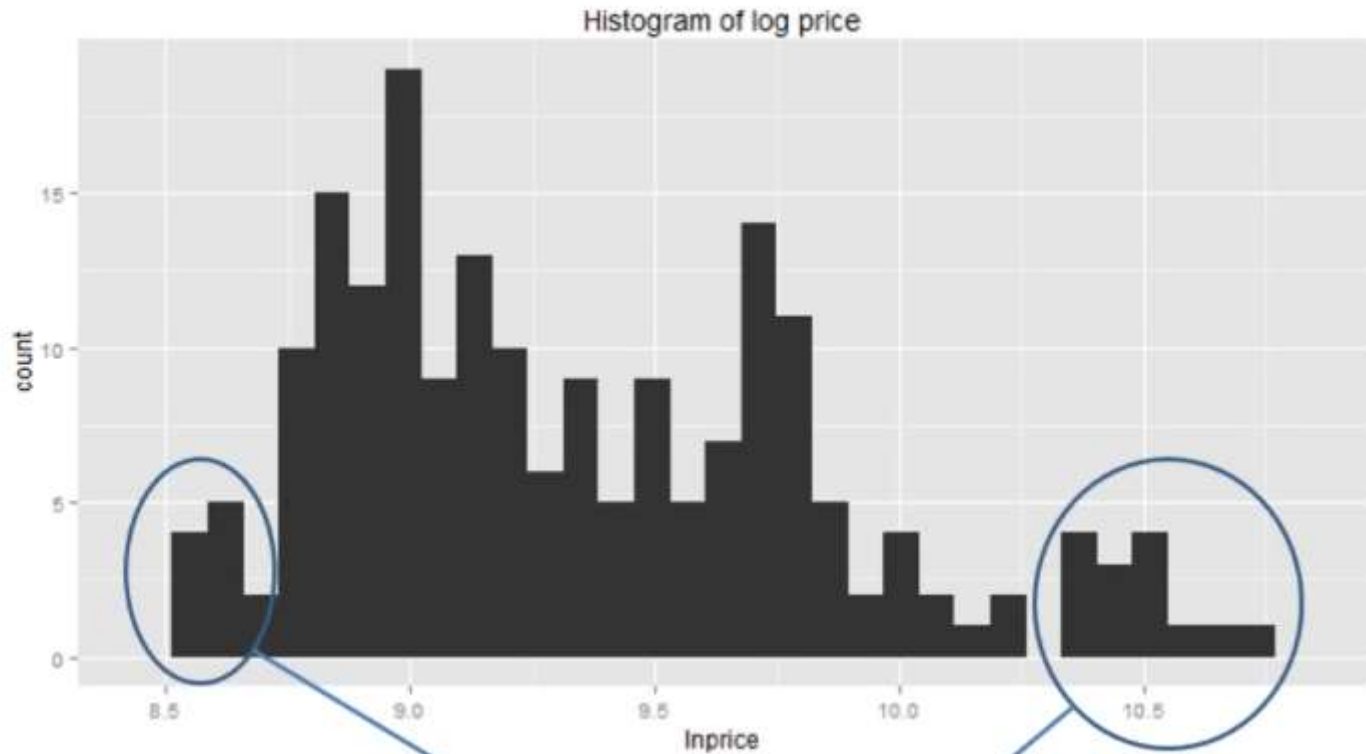


Identify Outliers and Errors



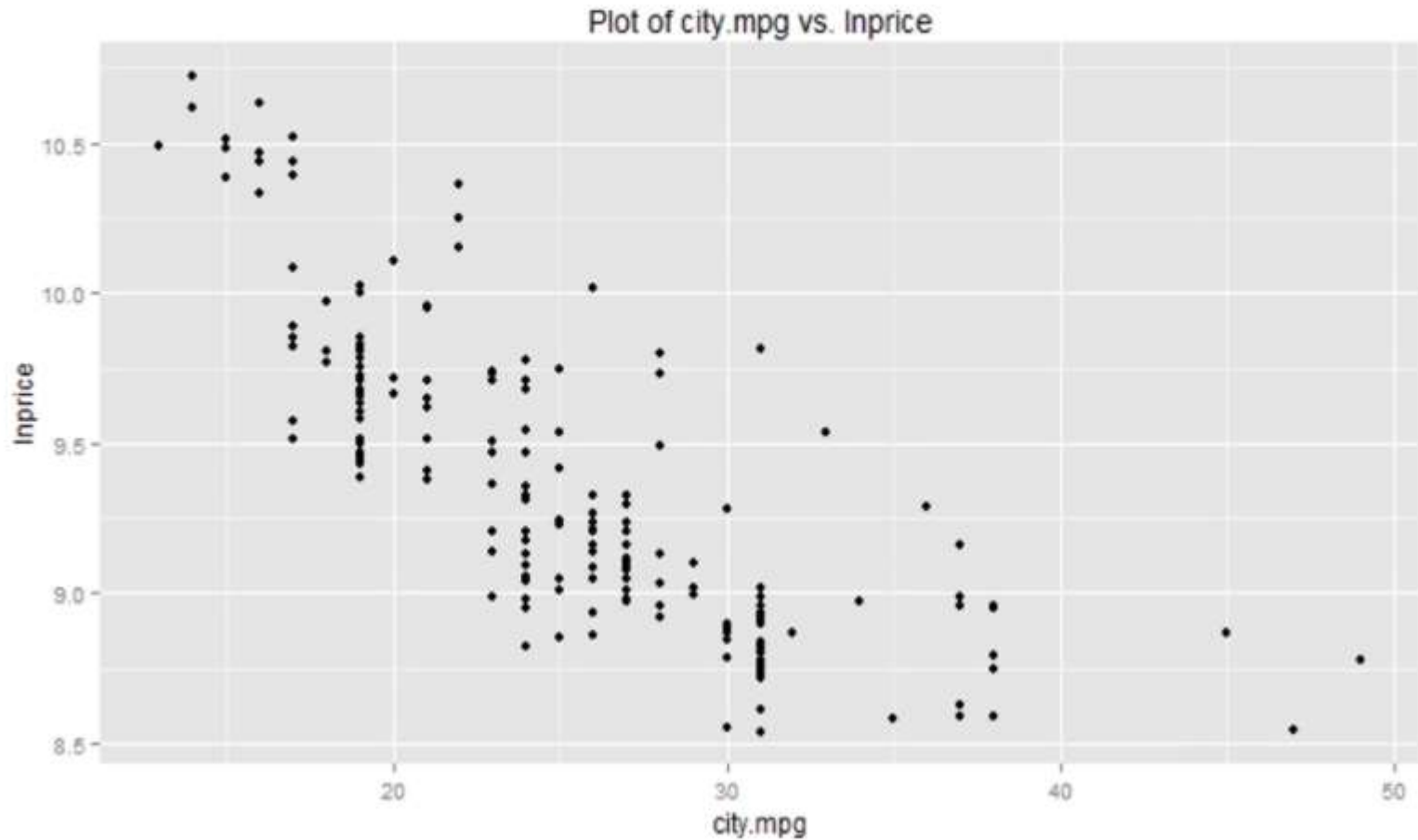
Outliers?

Identify Outliers and Errors

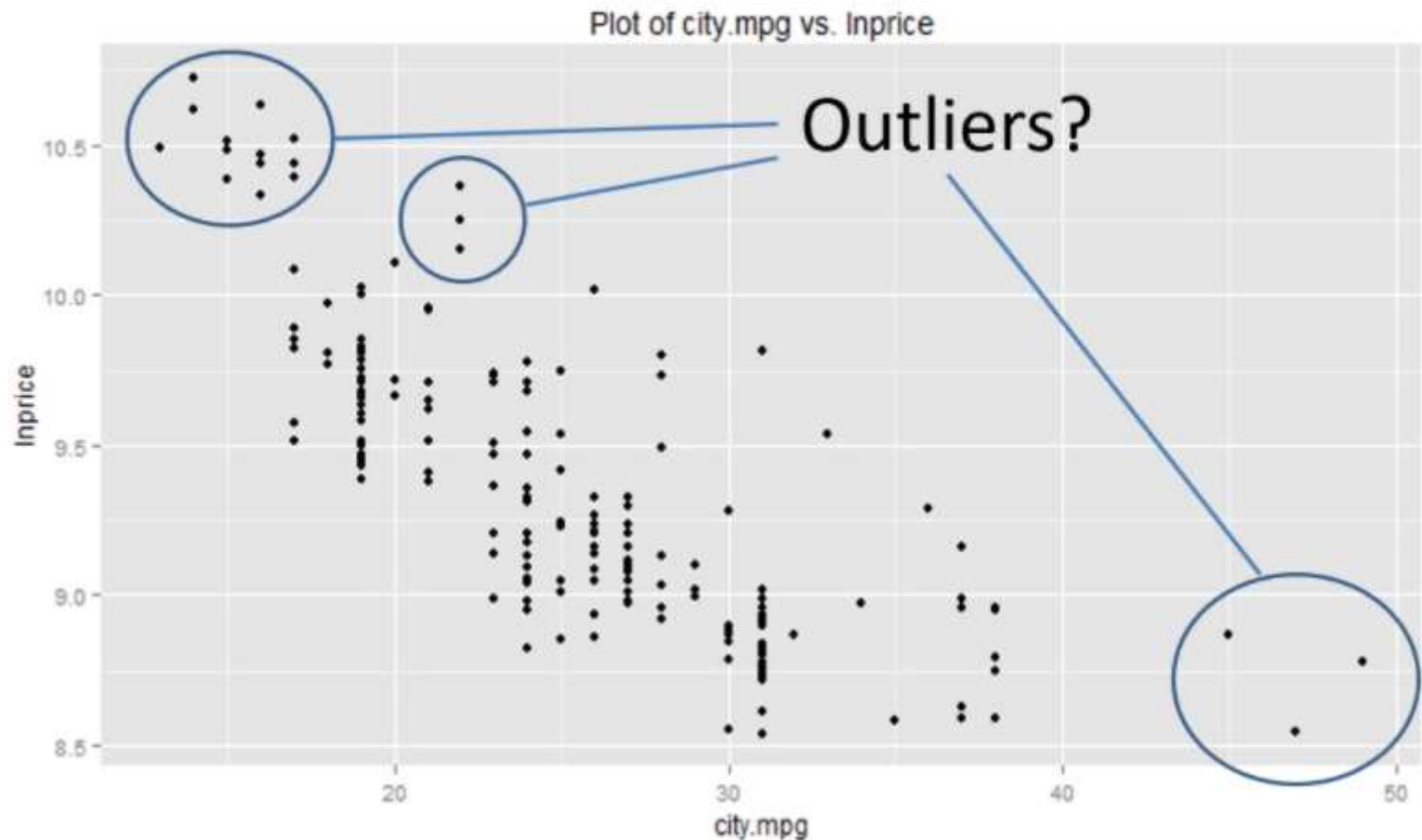


Outliers?

Identify Outliers and Errors



Identify Outliers and Errors



Clean outliers and errors

- Error treatments
 - Censor
 - Trim
 - Interpolate
 - Substitute
- Clip values module

Sample code snippet

Removing Outliers

R: `data.frame = data.frame[filter.expression,]`

```
library(dplyr)
```

```
frame1 <- frame1 %>% filter(Col1 > 40) %>%  
                    filter(Col2 < 30) %>%  
                    filter(Col3 < 3)
```

Python: `DataFrame = DataFrame[filter_expression]`

```
frame1 = frame1[(frame1["Col1"] > 40.0) &  
                (frame1["Col2"] < 30.0) &  
                (frame1["Col3"] < 3.0)]
```

Noisy Data

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

How to Handle Noisy Data?

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



Smoothing (noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

- equal-frequency partitioning

Smoothing (noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

- equal-frequency partitioning

bin 1	5,10,11,13
bin 2	15,35,50,55
bin 3	72,92,204,215

Smoothing (noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

- equal-frequency partitioning

bin 1	5,10,11,13
bin 2	15,35,50,55
bin 3	72,92,204,215

What is Smoothing by bin mean/median/boundary?

Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

- equal-frequency partitioning

bin 1	5,10,11,13
bin 2	15,35,50,55
bin 3	72,92,204,215

What is Smoothing by bin mean/median/boundary?

Replace each bin value is replaced by mean/median/nearest boundary

Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

- equal-width partitioning



Smoothing(noisy data)

Suppose a group of 12 *sales price* records has been sorted as follows:

5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215

Partition them into three bins by each of the following methods.

- equal-width partitioning

The width of each interval is $(215 - 5)/3 = 70$.

bin 1	5,10,11,13,15,35,50,55,72
bin 2	92
bin 3	204,215


Perform Smoothing by bin mean/median/boundary.

- 4,8, 15, 21, 21, 24, 25, 28, 34
- #no of bins=3
- Equal frequency bins
- Bin1:4,8,15
- Bin2:21,21, 24
- Bin3:25,28,34
- Smooting by bin bin boundaries
- Bin1: 4,4,15
- Bin 2:21,21,24
- **Bin 3: 25,25,34**

Data Cleaning as a Process

- Data discrepancy detection
 - Use metadata (e.g., domain, range, dependency, distribution)
 - Check field overloading
 - Check uniqueness rule, consecutive rule and null rule
 - Use commercial tools
 - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
 - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)
- Data migration and integration
 - Data migration tools: allow transformations to be specified
 - ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface
- Integration of the two processes
 - Iterative and interactive (e.g., Potter's Wheels)

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., $A.cust-id \equiv B.cust-\#$
 - 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

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)

- **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

Chi-Square Calculation: An Example

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

- 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 Analysis (Numeric Data)

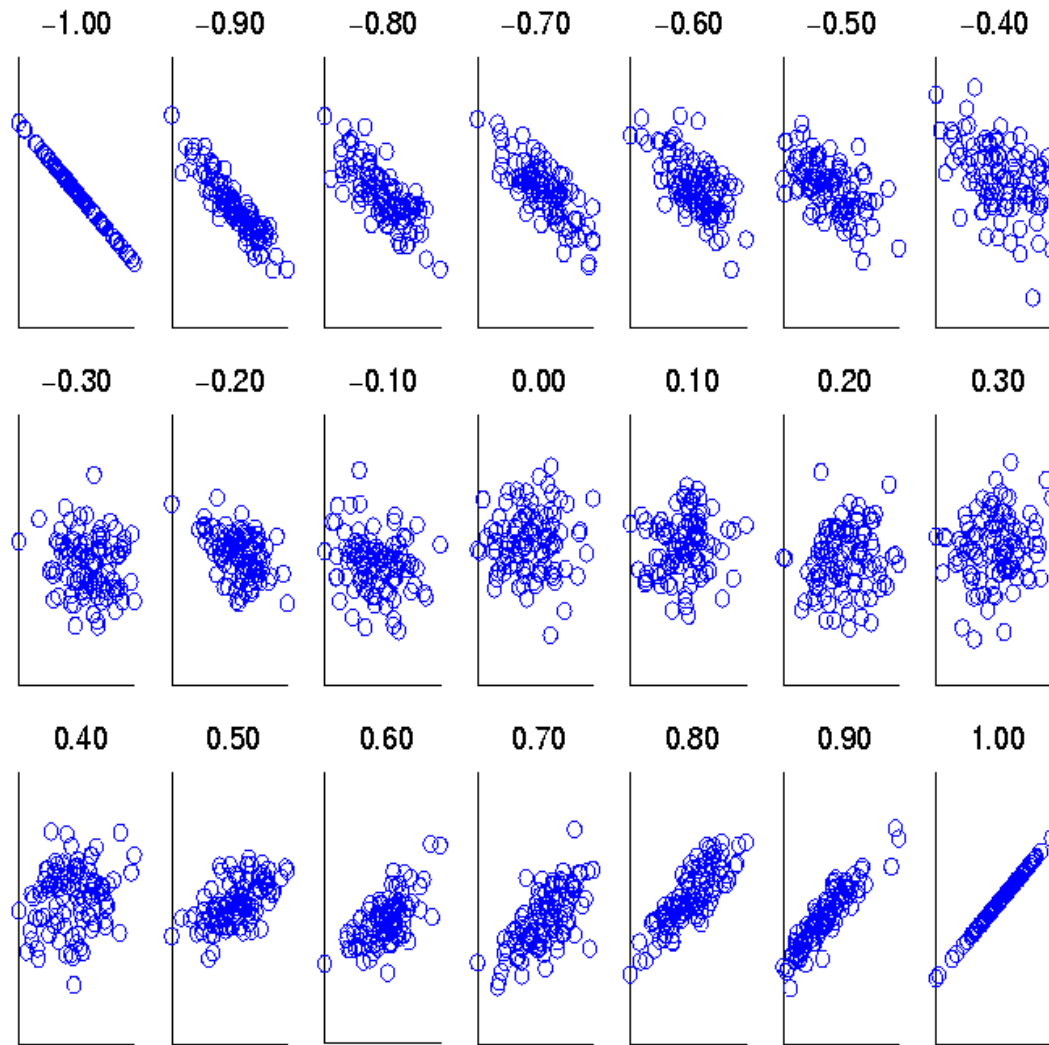
- Correlation coefficient (also called **Pearson's product moment coefficient**)

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

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 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.

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 .

- Positive covariance:** If $Cov_{A,B} > 0$, then A and B both tend to be larger than the 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?
 - $E(A) = (2 + 3 + 5 + 4 + 6) / 5 = 20/5 = 4$
 - $E(B) = (5 + 8 + 10 + 11 + 14) / 5 = 48/5 = 9.6$
 - $Cov(A, B) = (2 \times 5 + 3 \times 8 + 5 \times 10 + 4 \times 11 + 6 \times 14) / 5 - 4 \times 9.6 = 4$

- Thus, A and B rise together since $Cov(A, B) > 0$.

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

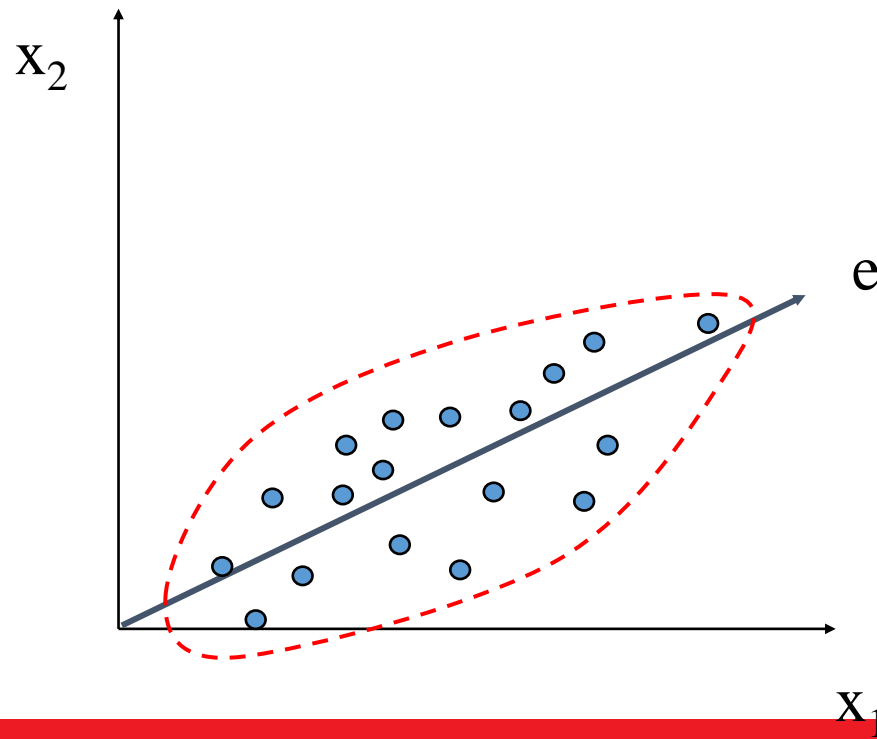


Data Reduction 1: Dimensionality Reduction

- **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



Principle Component Analysis (Steps)

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

Attribute Subset Selection

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

Heuristic Search in Attribute Selection

- There are 2^d possible attribute combinations of d attributes
- 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

Attribute Creation (Feature Generation)

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

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.: 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, ...

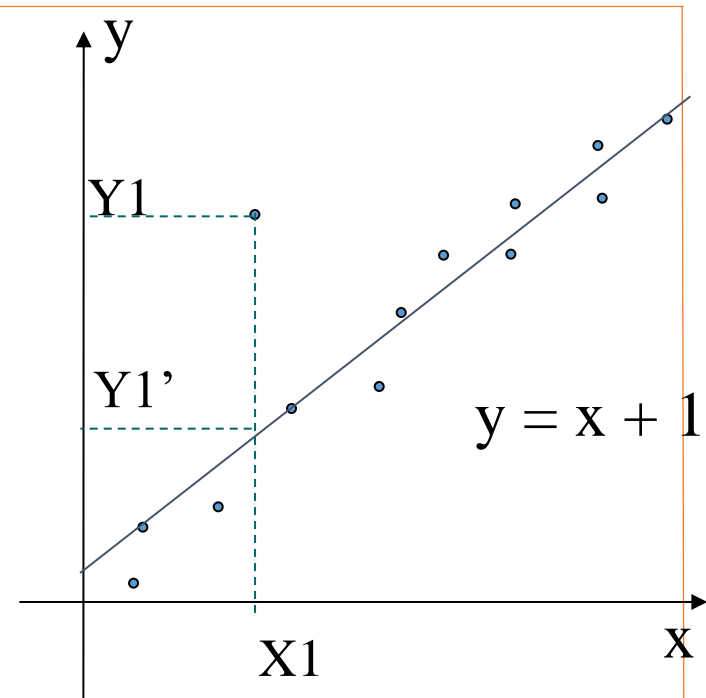
Parametric Data Reduction: Regression and Log-Linear Models

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



Regression Analysis

- 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)

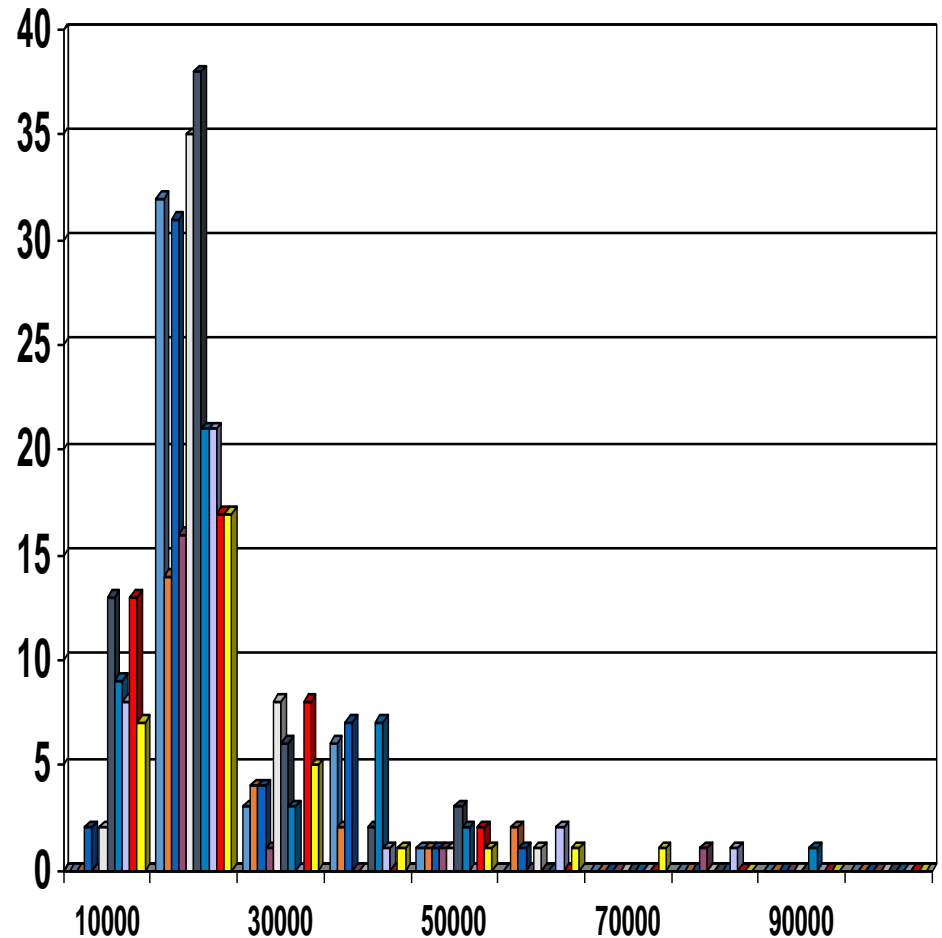
Regress Analysis and Log-Linear Models

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



Clustering

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



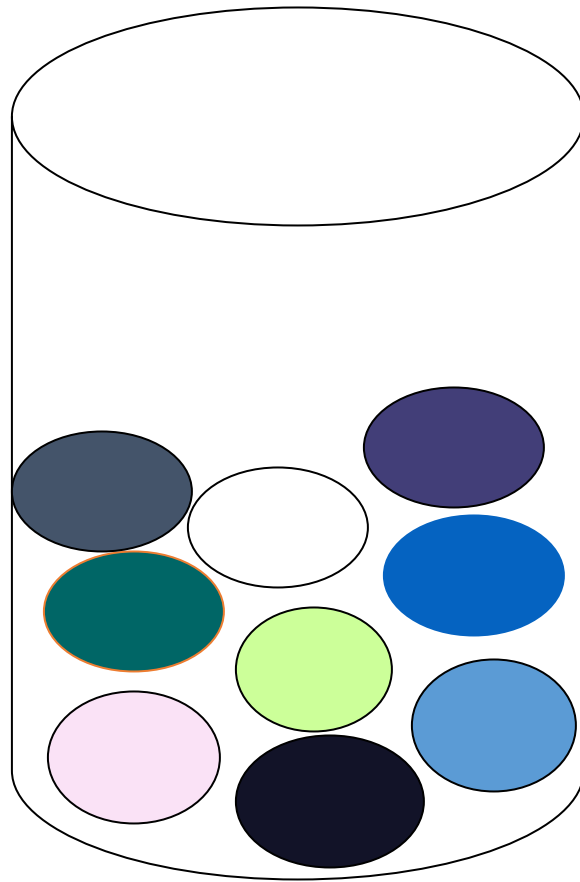
Sampling

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

Types of Sampling

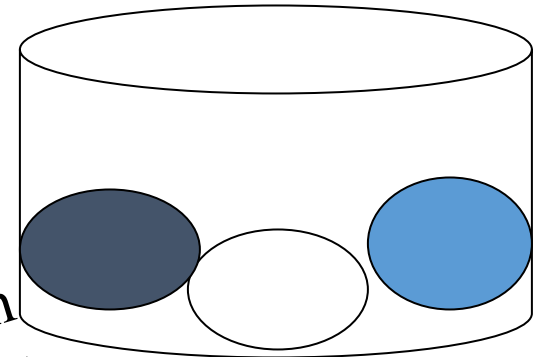
- **Simple random sampling**
 - There is an equal probability of selecting any particular item
- **Sampling without replacement**
 - Once an object is selected, it is removed from the population
- **Sampling with replacement**
 - A selected object is not removed from the population
- **Stratified sampling:**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

Sampling: With or without Replacement

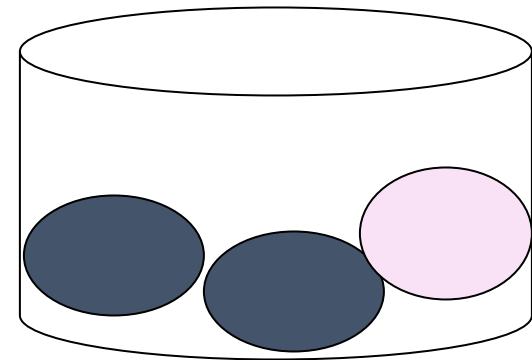


Raw Data

SRSWOR
(simple random
sample without
replacement)

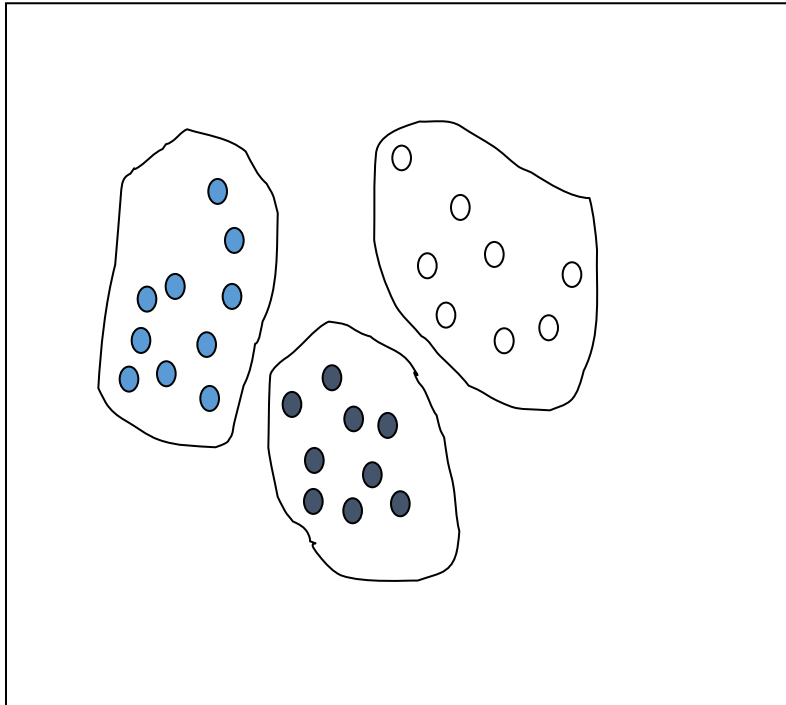


SRSWR

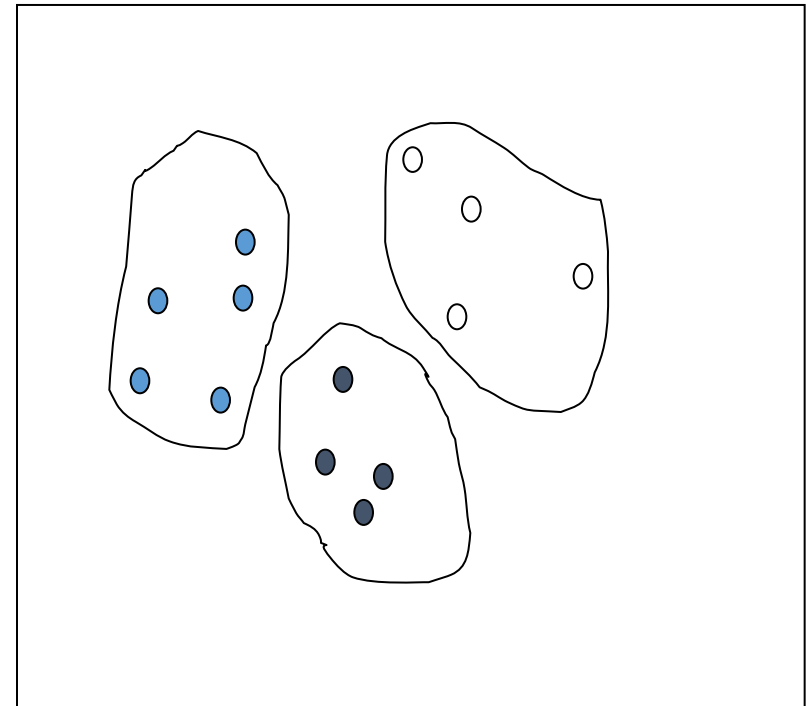


Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample

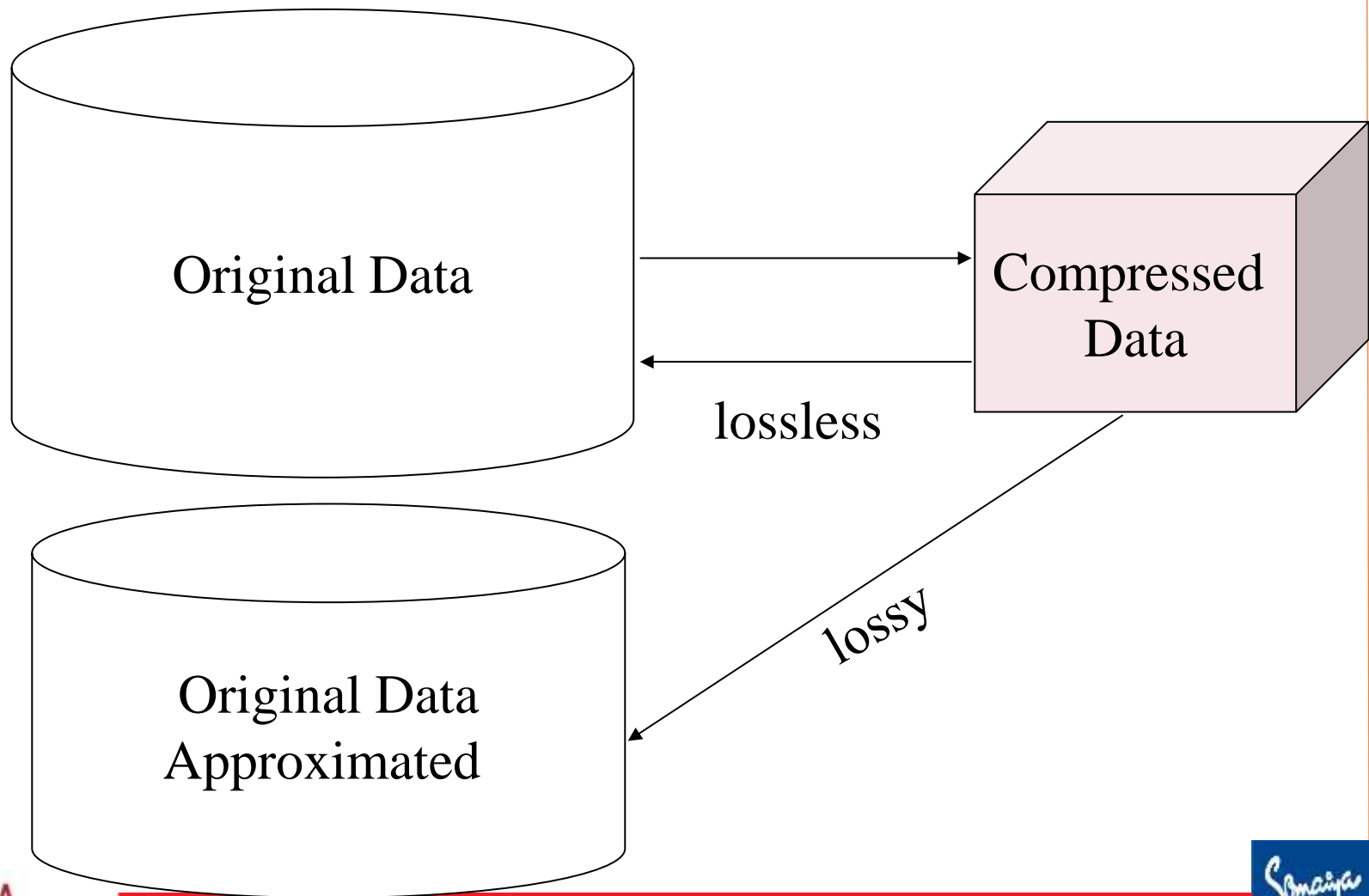


Data Reduction 3: Data Compression

- 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



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 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

Data Normalization in Data Mining

- **Normalization** is used to scale the data of an attribute so that it falls in a smaller range, such as -1.0 to 1.0 or 0.0 to 1.0.
- It is generally useful for classification algorithms.



Need Normalization

- Normalization is generally required when we are dealing with attributes on a different scale, otherwise, it may lead to a dilution in effectiveness of an important equally important attribute (on lower scale) because of other attribute having values on larger scale.
- In simple words, when multiple attributes are there but attributes have values on different scales, this may lead to poor data models while performing data mining operations. So they are normalized to bring all the attributes on the same scale.



person_name	Salary	Year_of_experience	Expected Position Level
Aman	100000	10	2
Abhinav	78000	7	4
Ashutosh	32000	5	8
Dishi	55000	6	7
Abhishek	92000	8	3
Avantika	120000	15	1
Ayushi	65750	7	5

The attributes salary and year_of_experience are on different scale and hence attribute salary can take high priority over attribute year_of_experience in the model.

Normalization

- **Min-max normalization:** to $[new_min_A, new_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

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

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

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

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

- **Normalization by decimal scaling**

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

- **Min-Max Normalization:** This transforms the original data linearly.
- Suppose that: \min_P is the minima and \max_P is the maxima of an attribute, P
- We Have the Formula:

$$v' = \frac{v - \min_P}{\max_P - \min_P} (\text{new_max}_P - \text{new_min}_P) + \text{new_min}_P$$

- Where v is the value you want to plot in the new range.
- v' is the new value you get after normalizing the old value.
- Solved **example**:

Suppose the minimum and maximum value for an attribute profit(P) are Rs. 10, 000 and Rs. 100, 000. We want to plot the profit in the range $[0, 1]$. Using min-max normalization the value of Rs. 20, 000 for attribute profit can be plotted to:

$$\frac{20000 - 10000}{100000 - 10000} (1 - 0) + 0 = 0.11$$

And hence, we get the value of v' as 0.11

- **Z-Score Normalization:** In z-score normalization (or zero-mean normalization) the values of an attribute (A), are normalized based on the mean of A and its standard deviation
- A value, v, of attribute A is normalized to v' by computing

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

- For **example**:

Let mean of an attribute $P = 60,000$, Standard Deviation = $10,000$, for the attribute P . Using z-score normalization, a value of 85000 for P can be transformed to:

$$\frac{85000 - 60000}{10000} = 2.50$$

And hence we get the value of v' to be 2.5

- **Decimal Scaling:**
- It normalizes the values of an attribute by changing the position of their decimal points
- The number of points by which the decimal point is moved can be determined by the absolute maximum value of attribute A.
- A value, v , of attribute A is normalized to v' by computing

$$v' = \frac{v}{10^j}$$

where j is the smallest integer such that $\text{Max}(|v'|) < 1$.

- For **example**:
- Suppose: Values of an attribute P varies from -99 to 99.
- The maximum absolute value of P is 99.
- For normalizing the values we divide the numbers by 100 (i.e., $j = 2$) or (number of integers in the largest number) so that values come out to be as 0.98, 0.97 and so on.





SOMAIYA
VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering



Data Discretization

- Data discretization refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data become easy.
- Data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss.
- There are two forms
 - Supervised discretization, and
 - Unsupervised discretization.
- Supervised discretization refers to a method in which the class data is used.
- Unsupervised discretization refers to a method depending upon the way which operation proceeds.
- It means it works on the top-down splitting strategy and bottom-up merging strategy.



- Suppose we have an attribute of Age with the given values

Age	1,5,9,4,7,11,14,17,13,18, 19,31,33,36,42,44,46,70,74,78,77
-----	--

Attribute	Age	Age	Age	Age
	1,5,4,9,7	11,14,17,13,18,19	31,33,36,42,44,46	70,74,77,78
After Discretization	Child	Young	Mature	Old

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 can be done by :
 - Interval labels can then be used to replace actual data values
 - Divide the range of a continuous attribute into intervals
 - Reduce data size by discretization(hierarchy)
 - Supervised vs. unsupervised(
 - Split (top-down) vs. merge (bottom-up)(specialization and generalization)
 - Discretization can be performed recursively on an attribute(profit per month from each day)
 - 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 (unsupervised, bottom-up merge)

Simple Discretization: Binning

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

Binning Methods for Data Smoothing

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

* Partition into equal-frequency (**equi-depth**) bins:

- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

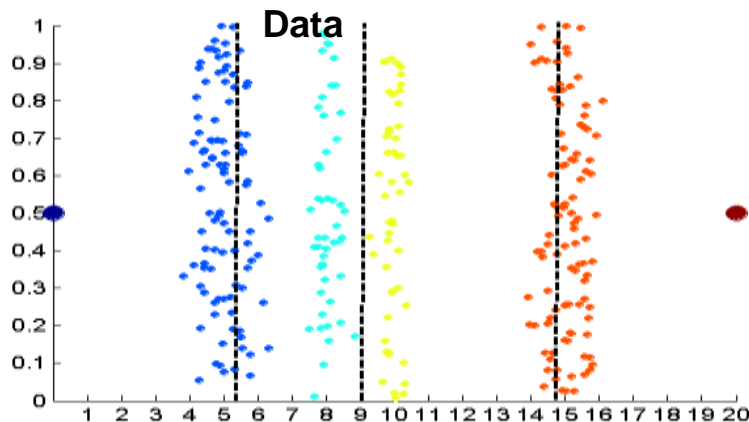
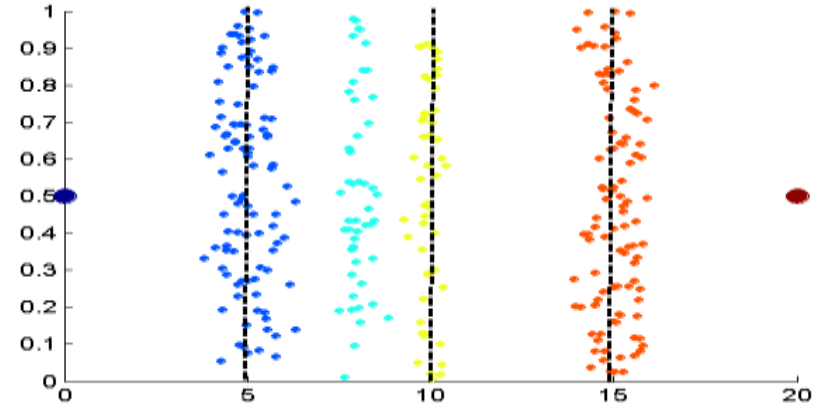
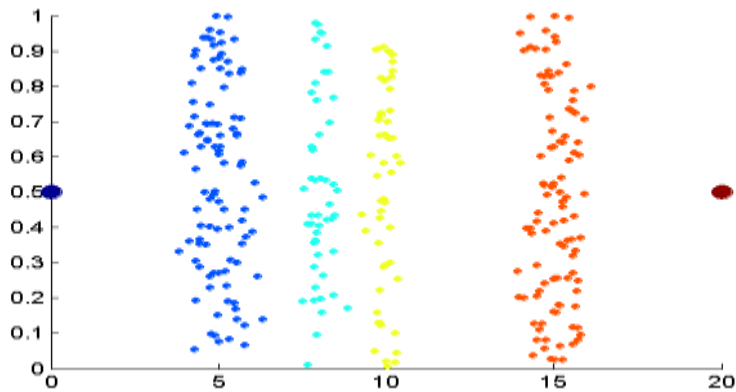
* Smoothing by **bin means**:

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

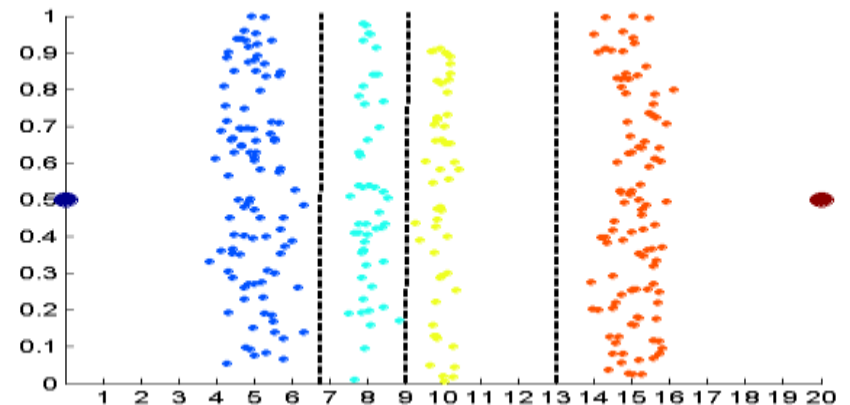
* Smoothing by **bin boundaries**:

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

Discretization Without Using Class Labels (Binning vs. Clustering)



Equal frequency (binning)



K-means clustering leads to better results

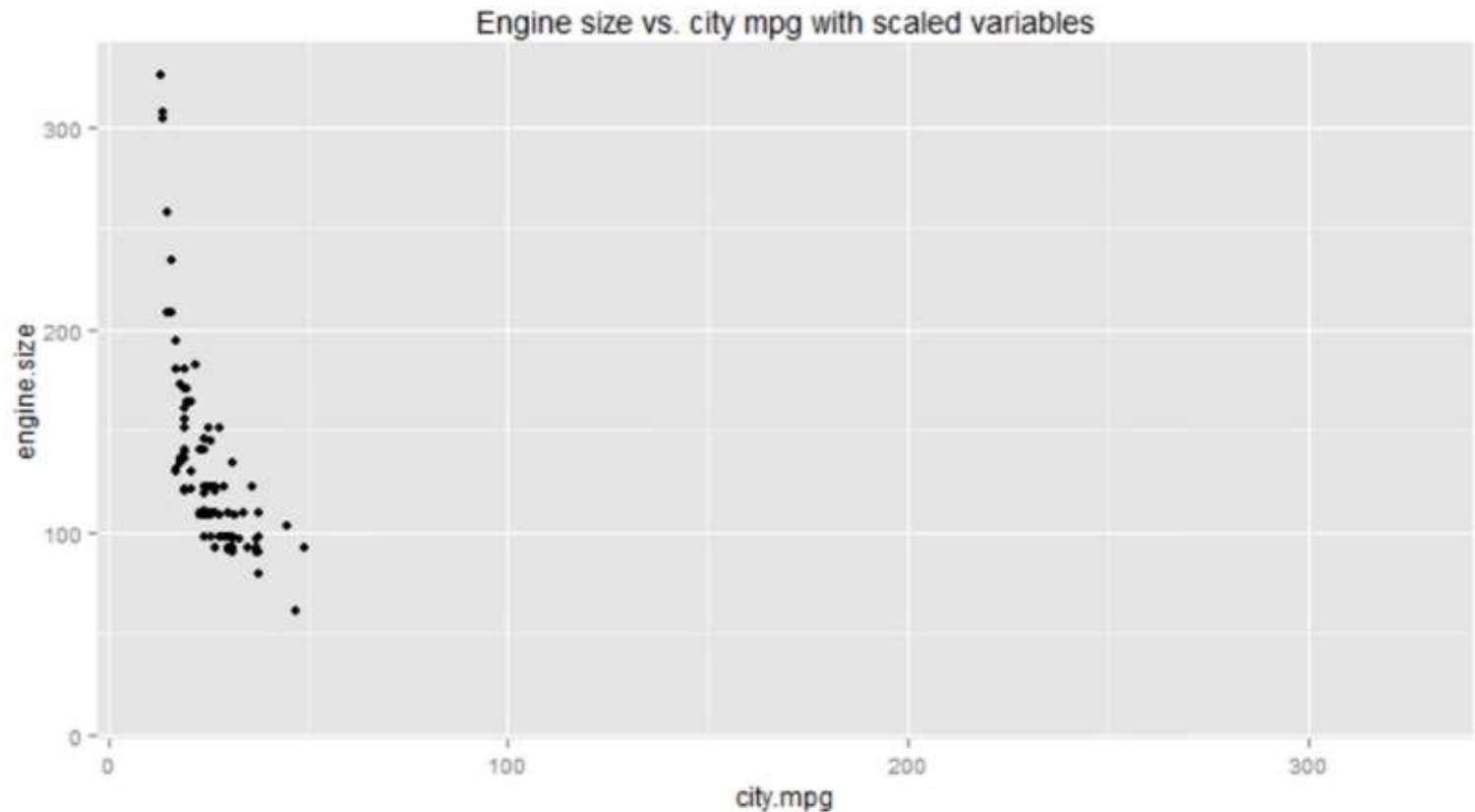
Discretization by Classification & Correlation Analysis

- 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 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

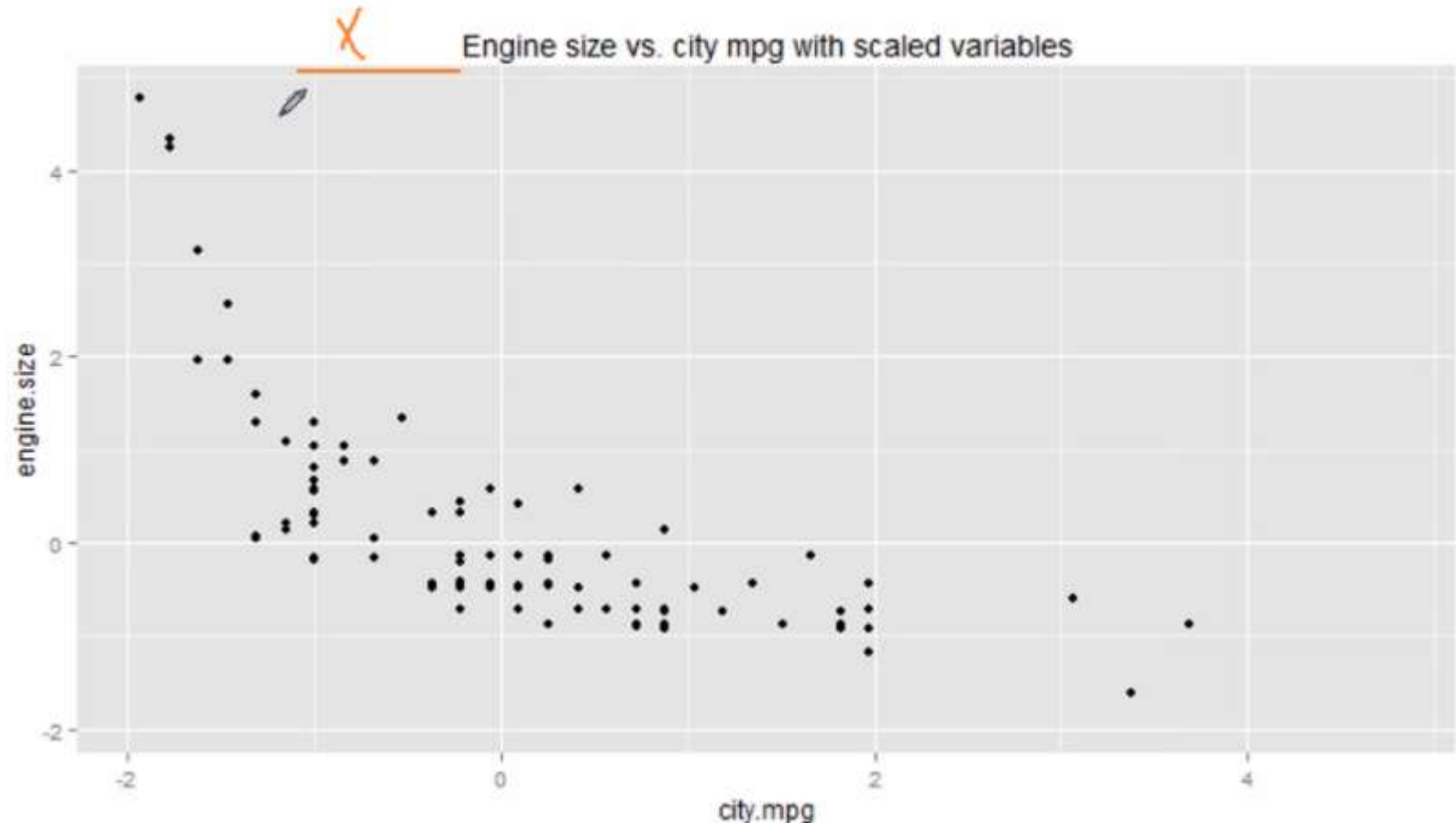
Concept Hierarchy Generation

- **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.

Unscaled data biases model construction



Scaled data biases model construction



Data Modeling
Algorithms
Python
Programming Skills
Update yourself
R Statistics
Read a lot Probability
Programming Fundamentals



SOMAIYA
VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering



Question ?



SOMAIYA
VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering

