

Data Management

Design Data Architecture and manage the Data for analysis:

Data architecture is composed of

- ✓ models,
- ✓ policies,
- ✓ rules or standards

that govern which

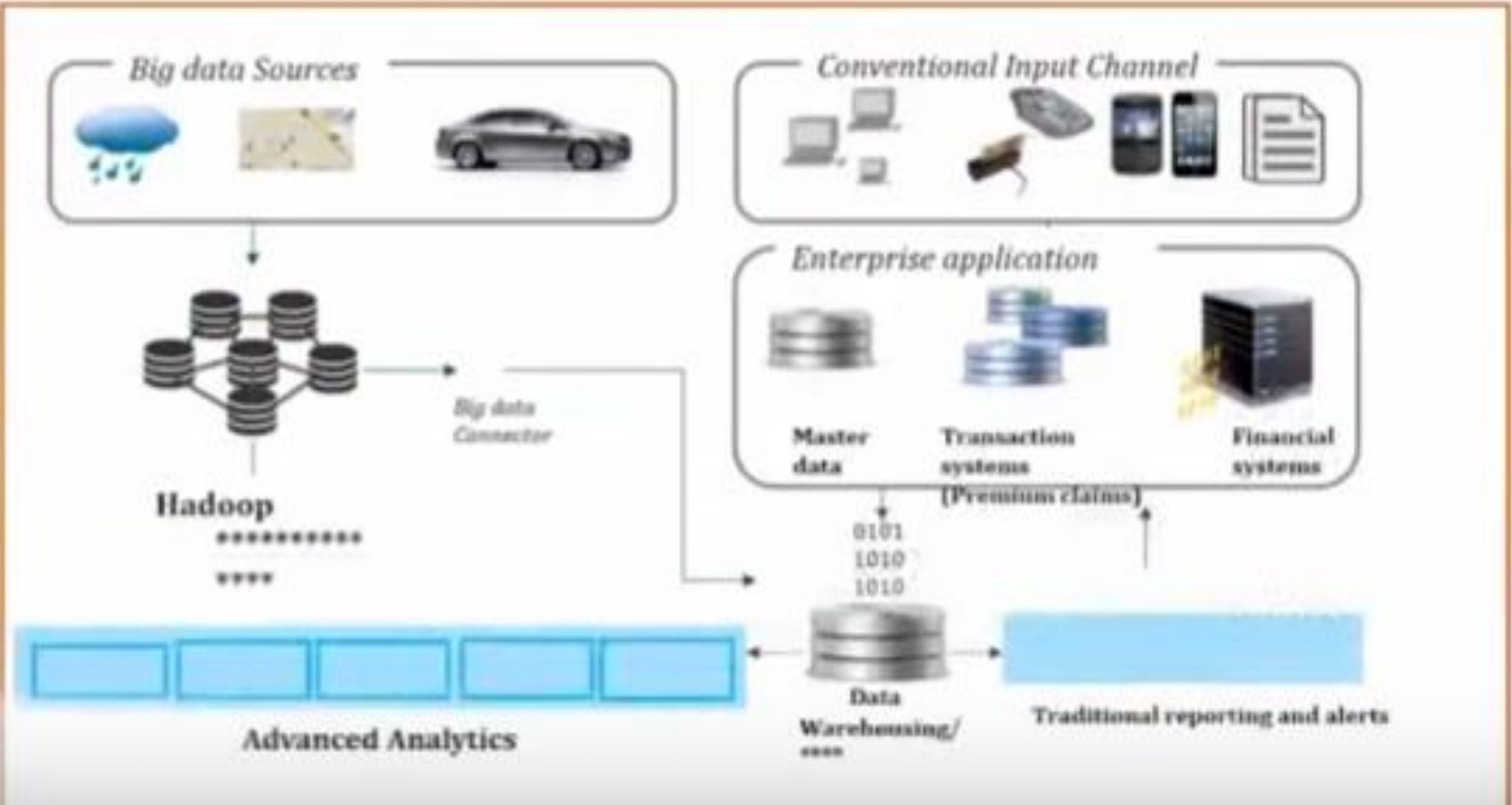
- data is collected,
- how it is stored,
- arranged,
- integrated,
- and put to use in data systems and in organizations.

Various constraints and influences will have an effect on data architecture design. These include:

- 1. Enterprise requirements,**
- 2. Technology drivers,**
- 3. Economics,**
- 4. Business policies and**
- 5. Data processing needs.**

Enterprise Data Architecture

An integrated view of architecture for data in an enterprise including structured and unstructured data.



Enterprise requirements :

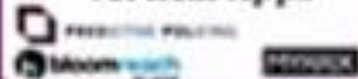
These will generally include such elements as **economical and effective system expansion, acceptable performance levels (especially system access speed), transaction reliability, and transparent data management.**

In addition, the conversion of raw data such as transaction records and image files into more useful information forms through such features as data warehouses is also a common organizational requirement, since this enables managerial decision making and other organizational processes.

One of the architecture techniques is the split between

1. Managing transaction data and (master) reference data.
2. Another one is splitting data capture systems from data retrieval systems (as done in a data warehouse).

Vertical Apps



Ad / Media Types



Business Intelligence



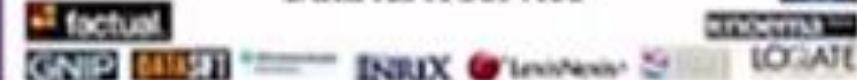
Analytics and Visualization



Log Data Apps



Data AS A Service



Analytics Infrastructure



Operational Infrastructure



Infrastructure AS A Service



Structured Databases



Technologies



Technology drivers :

These are usually suggested by the completed **data architecture** and **database architecture designs**.

In addition, some technology drivers will derive from existing

1. Organizational integration frameworks and standards,
2. Organizational economics, and existing site resources (e.g. previously purchased software licensing).

- **Economics**

These are also important factors that must be considered during the data architecture phase. It is possible that **some solutions, while optimal in principle, may not be potential candidates due to their cost.** External factors such as the **business cycle, interest rates, market conditions,** and legal considerations could all have an effect on decisions relevant to data architecture.

- **Business policies**

Business policies that also drive data architecture design include **internal organizational policies, rules of regulatory bodies, professional standards, and applicable governmental laws that can vary by applicable agency.** These policies and rules will help describe the manner in which enterprise wishes to process their data.

A data architecture aims to set data standards for all its data systems as a vision or a model of the eventual interactions between those data systems. [Data integration](#), for example, should be dependent upon data architecture standards since data integration requires data interactions between two or more data systems. A data architecture, in part, describes the [data structures](#) used by a business and its computer [applications software](#). Data architectures address data in storage, data in use and data in motion; descriptions of data stores, data groups and data items; and [mappings](#) of those data artifacts to data qualities, applications, locations etc.

Essential to realizing the target state, Data Architecture describes how data is processed, stored, and utilized in an [information system](#). It provides criteria for [data processing](#) operations so as to make it possible to design [data flows](#) and also control the flow of data in the system.

The [data architect](#) is typically responsible for defining the target state, aligning during development and then following up to ensure enhancements are done in the spirit of the original blueprint.

During the definition of the target state, the Data Architecture breaks a subject down to the atomic level and then builds it back up to the desired form. The data architect breaks the subject down by going through 3 traditional architectural processes:

- Conceptual - represents all [business entities](#).
- Logical - represents the logic of how entities are related.
- Physical - the realization of the data mechanisms for a specific type of functionality.

Data processing needs :

These include accurate and reproducible transactions performed in high volumes, data warehousing for the support of management information systems (and potential data mining), repetitive periodic reporting, ad hoc reporting, and support of various organizational initiatives as required (i.e. annual budgets, new product development).

The General Approach is based on designing the Architecture at three Levels of Specification :-

- ✓The logical layer
- ✓The physical layer
- ✓The implementation layer

Understand various sources of Data

Data can be generated from two types of sources
namely :

- ✓ **Primary**
- ✓ **Secondary Sources of Primary Data**

The sources of generating primary data are –

- ✓ Observation Method .
- ✓ Experimental Method
- ✓ Survey Method

There are number of experimental designs that are used in carrying out and experiment. However, Market researchers have used 4 experimental designs most frequently.

These are –

✓ **CRD** - Completely Randomized Design

✓ **RBD** - Randomized Block Design

✓ **LSD** - Latin Square Design

✓ **FD** - Factorial Designs

RBD - Randomized Block Design –

✓The Term Randomized Block Design has originated from agricultural research.

✓In this design several treatments of variables are applied to different blocks of land to ascertain their effect on the yield of the crop. Blocks are formed in such a manner that each block contains as many plots as a number of treatments so that one plot from each is selected at random for each treatment.

✓The production of each plot is measured after the treatment is given. These data are then interpreted and inferences are drawn by using the analysis of Variance Technique so as to know the effect of various treatments like different dozes of fertilizers, different types of irrigation etc.

LSD - Latin Square Design –

A Latin square is one of the experimental designs which has a **balanced two-way classification scheme** say for example - 4 X 4 arrangement. In this scheme each letter from A to D occurs only once in each row and also only once in each column. The balance arrangement, it may be noted that, will not get disturbed if any row gets changed with the other.

A B C D

B C D A

C D A B

D A B C

The balance arrangement achieved in a Latin Square is its main strength. In this design, the comparisons among treatments, will be free from both differences between rows and columns. Thus the magnitude of error will be smaller than any other design.

FD - Factorial Designs –

- ✓ This design allows the experimenter to **test two or more variables** simultaneously.
- ✓ It also measures interaction effects of the variables and analyzes the impacts of each of the variables.
- ✓ In a true experiment, randomization is essential so that the experimenter can infer cause and effect without any bias.

Sources of Secondary Data :

While primary data can be collected through questionnaires, depth interview, focus group interviews, case studies, experimentation and observation; The secondary data can be obtained through:

Internal Sources - These are within the organization

External Sources - These are outside the organization

The internal sources include:

Accounting resources- This gives so much information which can be used by the marketing researcher. They give information about internal factors.

Sales Force Report- It gives information about the sale of a product. The information provided is of outside the organization.

Internal Experts- These are people who are heading the various departments. They can give an idea of how a particular thing is working

Miscellaneous Reports- These are what information you are getting from operational reports.

If the data available within the organization are unsuitable or inadequate, the marketer should extend the search to external secondary data sources.

External Sources of Data:

External Sources are sources which are outside the company in a larger environment. Collection of external data is more difficult because the data have much greater variety and the sources are much more numerous.

- **Government Publications**
- **Central Statistical Organization**
- **Director General of Commercial Intelligence**
- **Ministry of Commerce and Industries**
- **Planning Commission**
- **Reserve Bank of India**

Understand various sources of Data like Sensors/signal/GPS etc.

Sensor data:

Sensor data is the output of a device that detects and responds to some type of **input from the physical environment**. The output may be used to provide information or input to another system or to guide a process.

Here are a few examples of sensors, just to give an idea of the number and diversity of their applications:

- **A photosensor** detects the presence of visible light, infrared transmission (IR) and/or ultraviolet (UV) energy.
- **Lidar, a laser-based method of detection**, range finding and mapping, typically uses a low-power, eye-safe pulsing laser working in conjunction with a camera.

- **A charge-coupled device (CCD)** stores and displays the data for an image in such a way that each pixel is converted into an electrical charge, the intensity of which is related to a color in the color spectrum.
- **Smart grid sensors** can provide real-time data about grid conditions, detecting outages, faults and load and triggering alarms.
- **Wireless sensor networks** combine specialized transducers with a communications infrastructure for monitoring and recording conditions at diverse locations. Commonly monitored parameters include temperature, humidity, pressure, wind direction and speed, illumination intensity, vibration intensity, sound intensity, power-line voltage, chemical concentrations, pollutant levels and vital body functions

What is signal?

- The simplest form of **signal** is a direct current (DC) that is switched on and off; this is the principle by which the early telegraph worked. More complex **signals** consist of an alternating-current (AC) or electromagnetic carrier that contains one or more **data** streams.

Data and Signals

- Data must be transformed into electromagnetic signals prior to transmission across a network.
- Data and signals can be either analog or digital.
- * A signal is periodic if it consists of a continuously repeating pattern.

Data like GPS

The **Global Positioning System (GPS)** is a space-based navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

The system provides critical capabilities to military, civil, and commercial users around the world.

The United States government created the system, maintains it, and makes it freely accessible to anyone with a GPS receiver.

Sources of Data

- Data collected and stored in a variety of sources, such as:
 - Electronic medical records.
 - Electronic health records.
 - Claims databases.
 - Health surveys.
 - Patient registries.
 - Health-related apps and mobile devices.
 - Social media.
 - E-commerce(online shopping)

Excuse me – But what is Data ?

Data

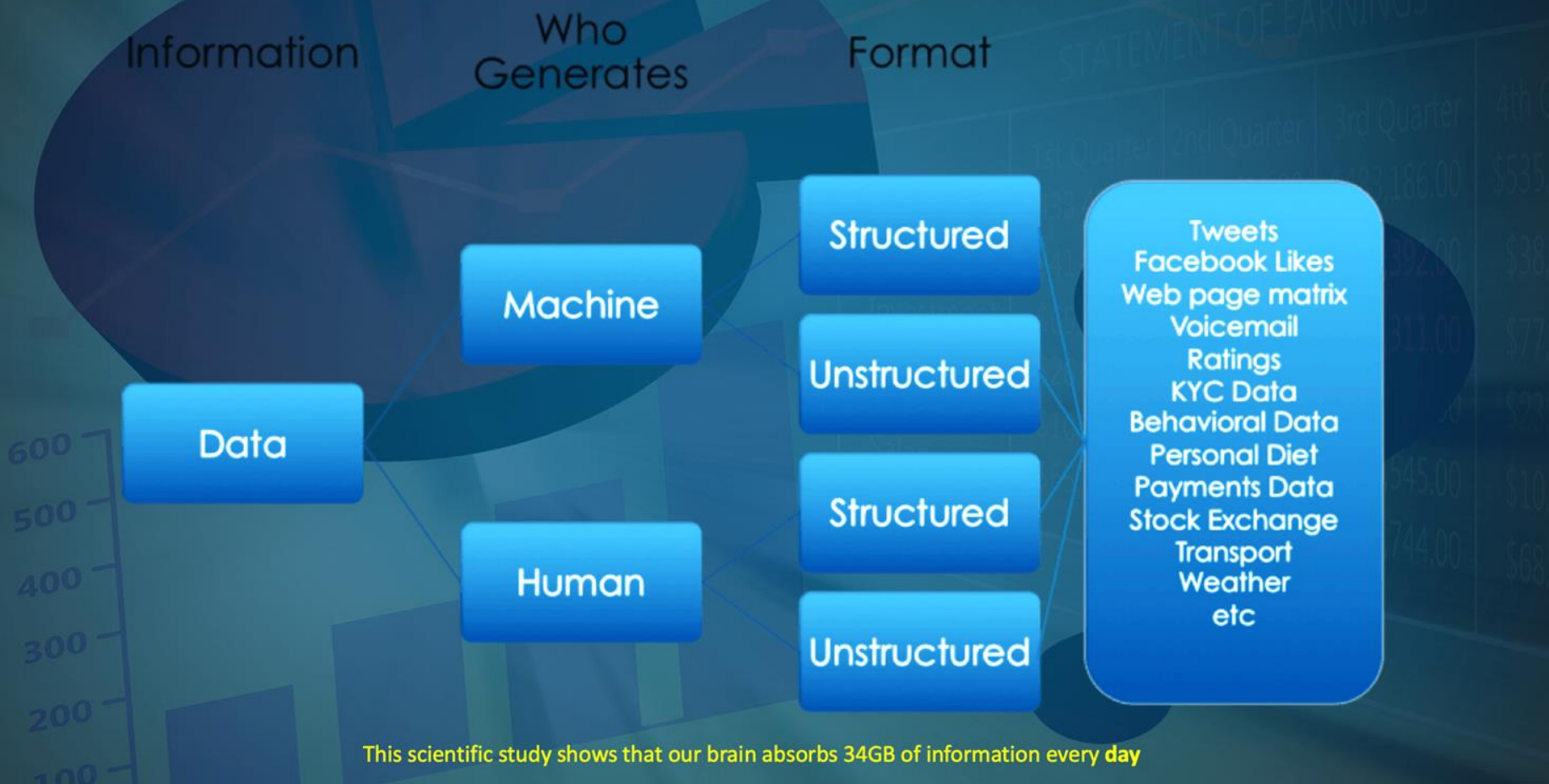
- From computer science perspective it is a combination of various chunks of information which has been formatted to a specified information dimension.
- Distinct facts, statistics around an event which have been put together to use for processing in producing signals, conclusions, derivatives and desired consequence.
 - **Structured Data**
 - Is a data organized to a specified data format based on data formatting methodologies into a formatted repository such as a database, as a way of making the data easy to analyze, as well as process.
 - Types of structured data are records, vectors and arrays.
 - **Unstructured Data**
 - Is information which is not in a format of any known data model, unstructured in a simple way.
 - Types of unstructured data are unstructured text, presentations, spreadsheets.

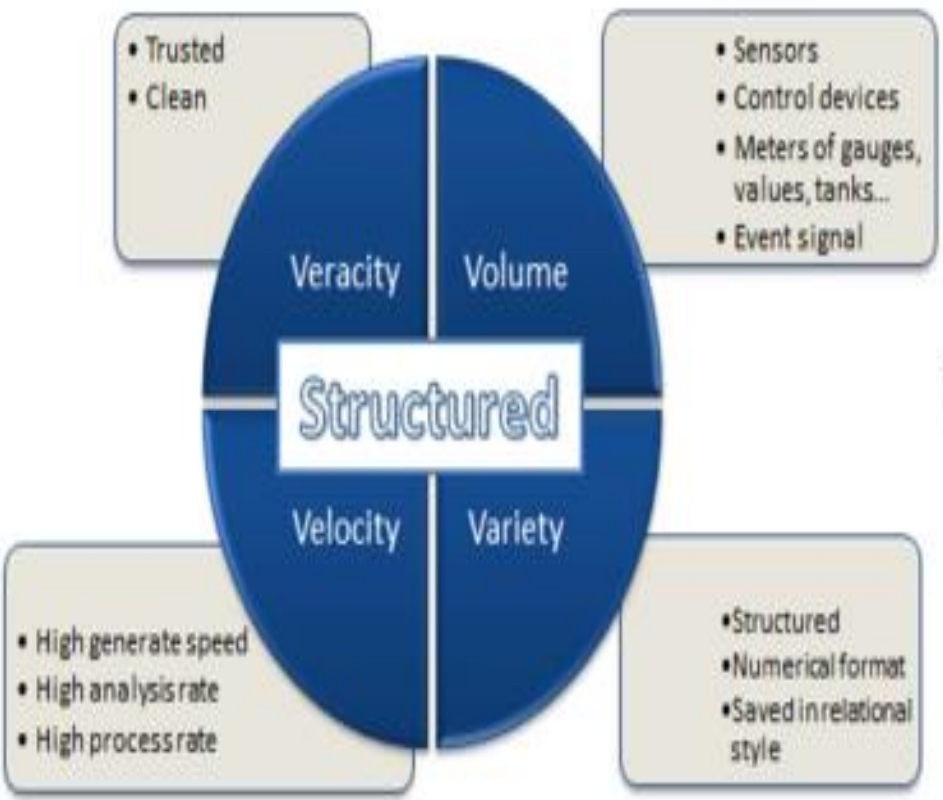
Difference Between Structured and Unstructured data

- **Structured Data** - Is in a format that easy for computers and computational models to understand.
- **Unstructured Data** - Is in a human understandable format, which is too rich, but very difficult for machines to understand.

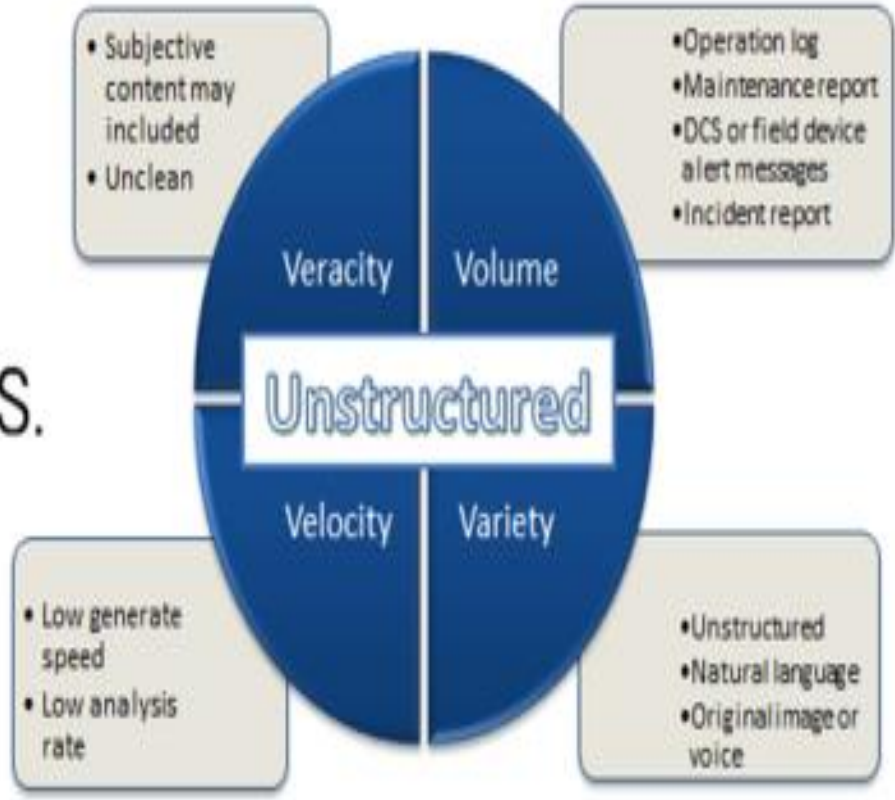
Structured data is meant for computers whilst unstructured data is human like format

From Where does the Data comes from?





V.S.



Data Mining:

Concepts and Techniques


(3rd ed.)

— Chapter 3 —

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

©2011 Han, Kamber & Pei. All rights reserved.

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
- **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

- 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?

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

Data Cleaning as a Process

- Data discrepancy detection
 - Use metadata (e.g., domain, range, dependency, distribution)
 - Check field overloading
 - 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)

- **X² (chi-square) test**

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

- The larger the X² value, the more likely the variables are related
- The cells that contribute the most to the X² 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)

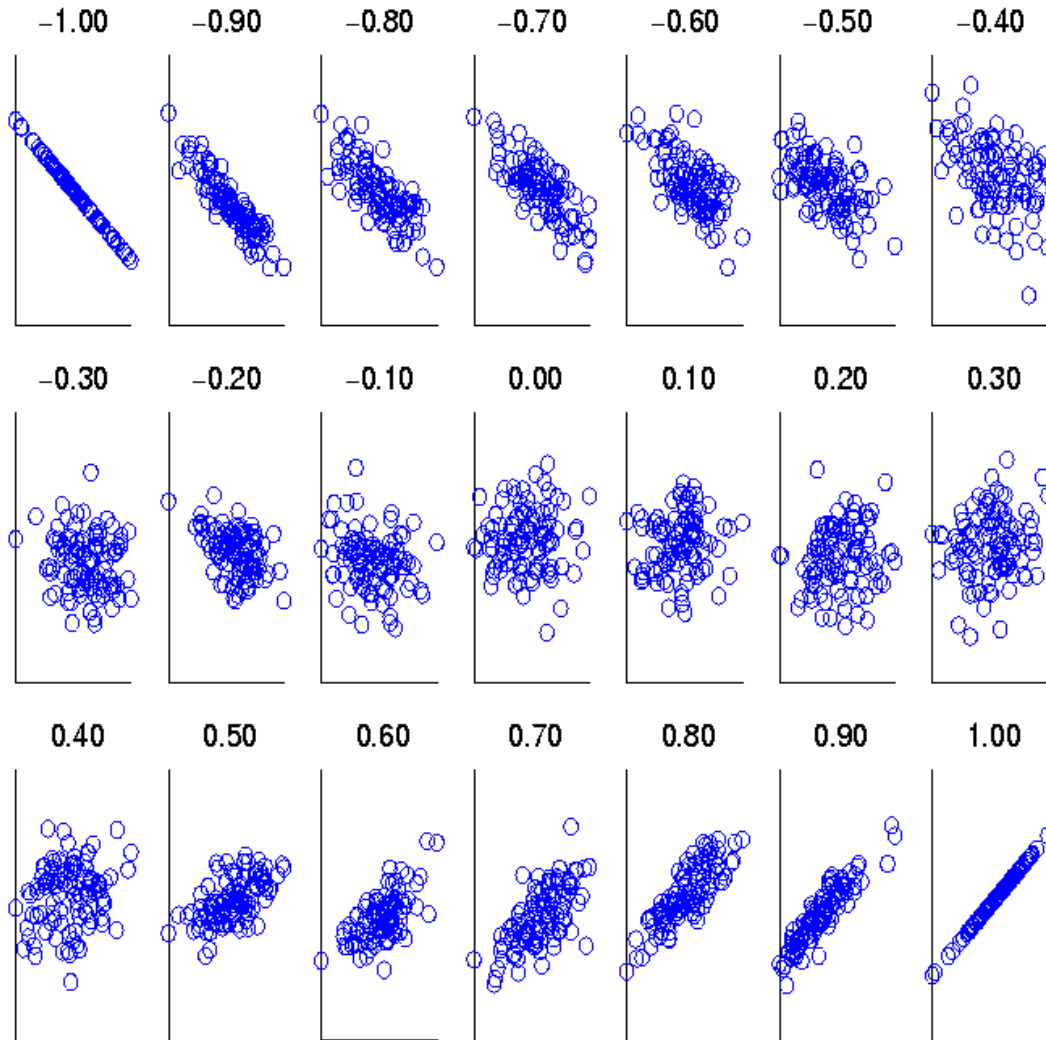
- Correlation coefficient (also called **Pearson's product moment coefficient**)

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

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

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

Visually Evaluating Correlation



Scatter plots showing the similarity from -1 to 1.

Correlation (viewed as linear relationship)

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

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

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

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

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 .

- **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 independence₂₀

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

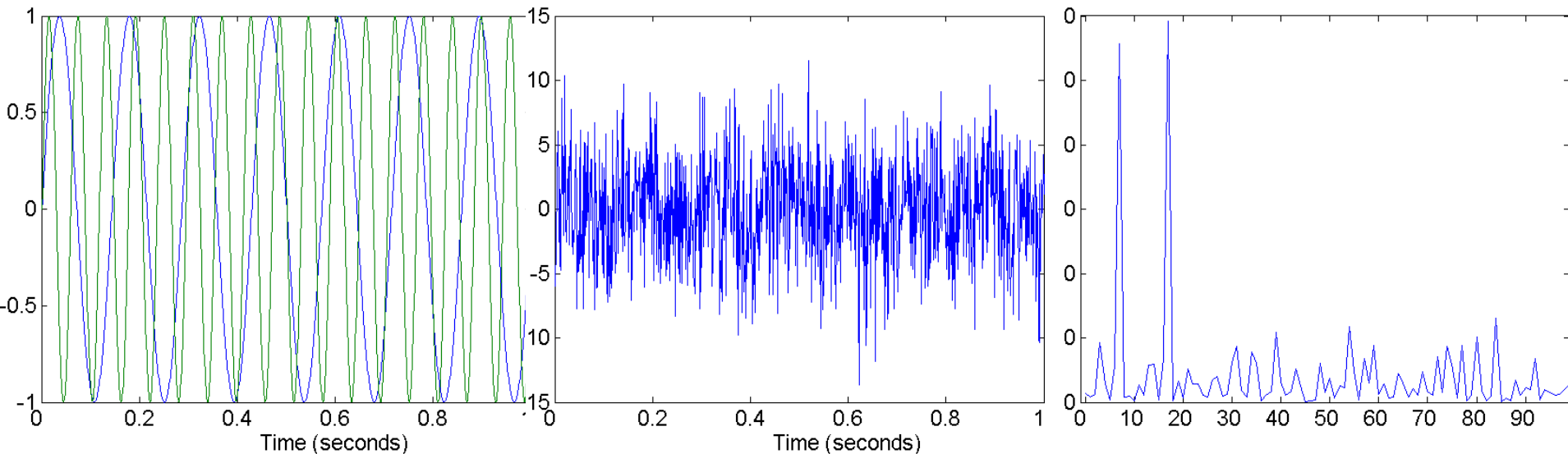
- **Data reduction:** Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run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

Mapping Data to a New Space

- **Fourier transform**
- **Wavelet transform**



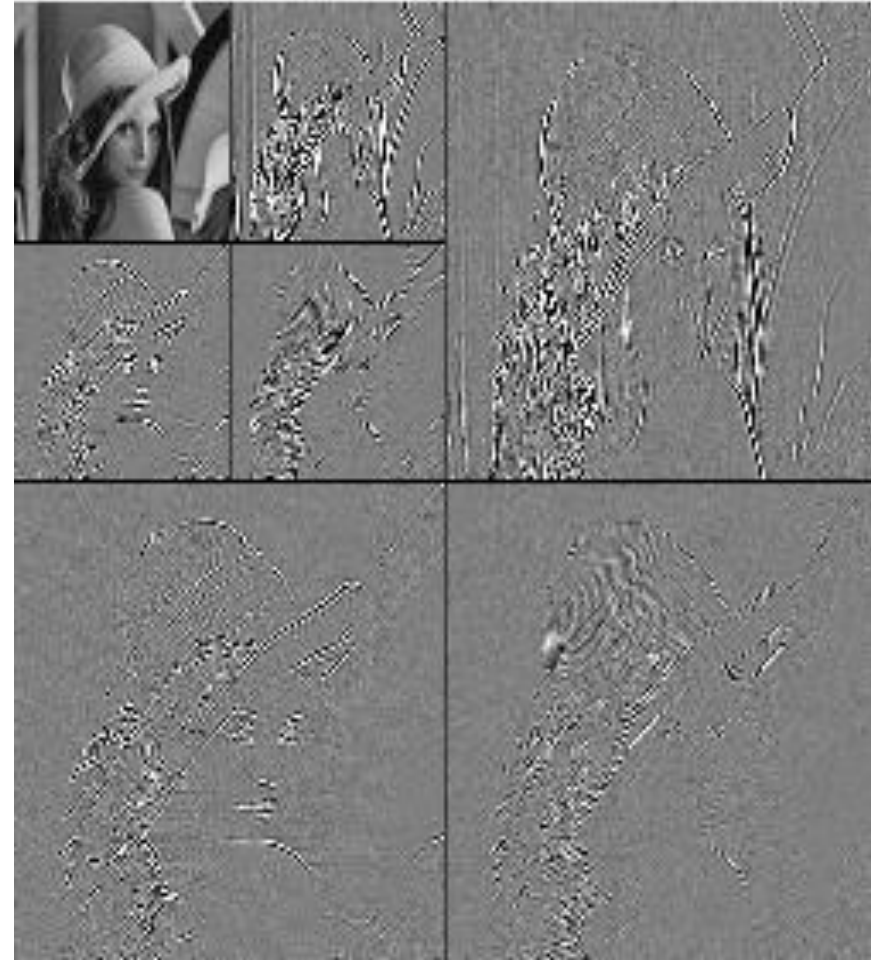
Two Sine Waves

Two Sine Waves + Noise

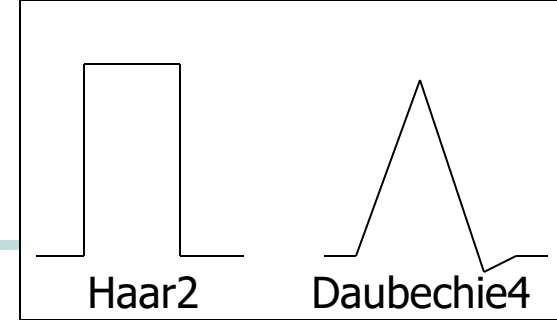
Frequency

What Is Wavelet Transform?

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



Wavelet Transformation



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

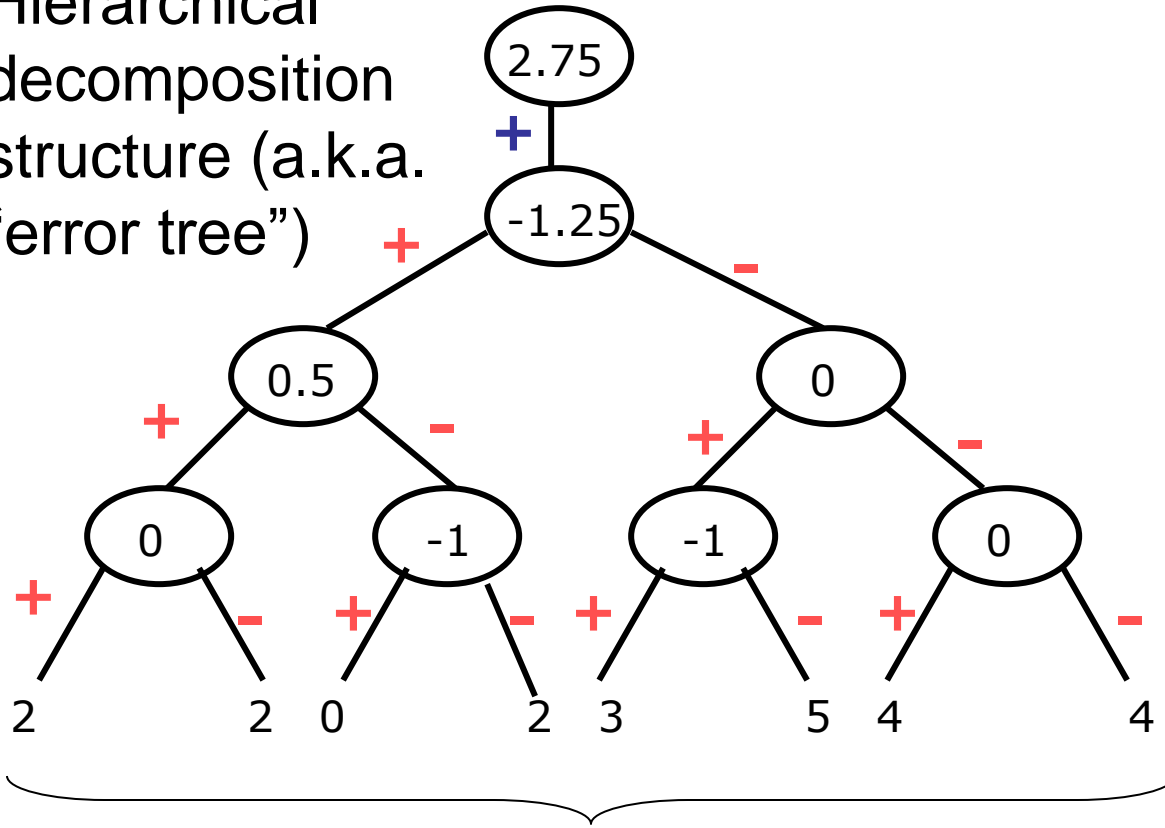
Wavelet Decomposition

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

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

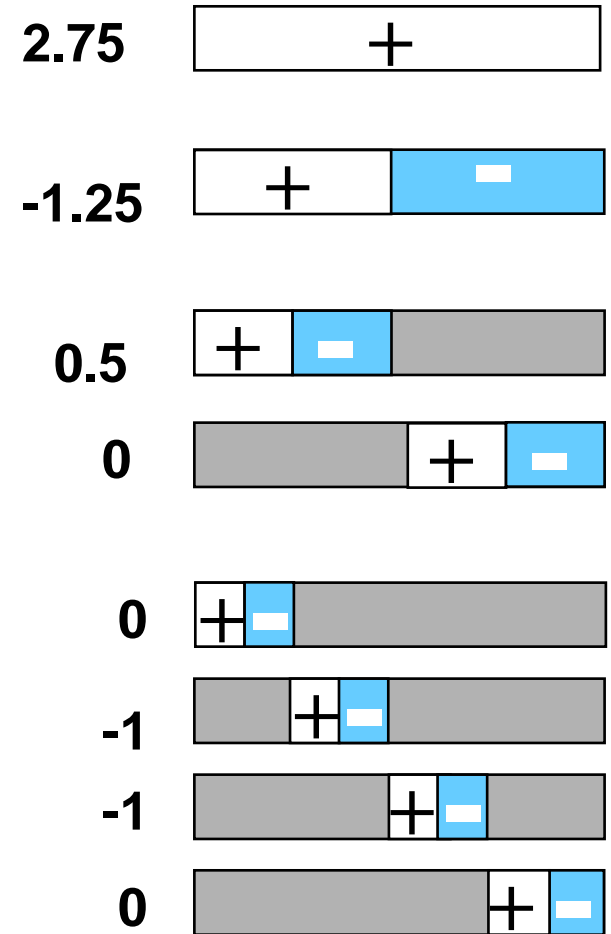
Haar Wavelet Coefficients

Hierarchical decomposition structure (a.k.a. "error tree")



Original frequency distribution

Coefficient "Supports"

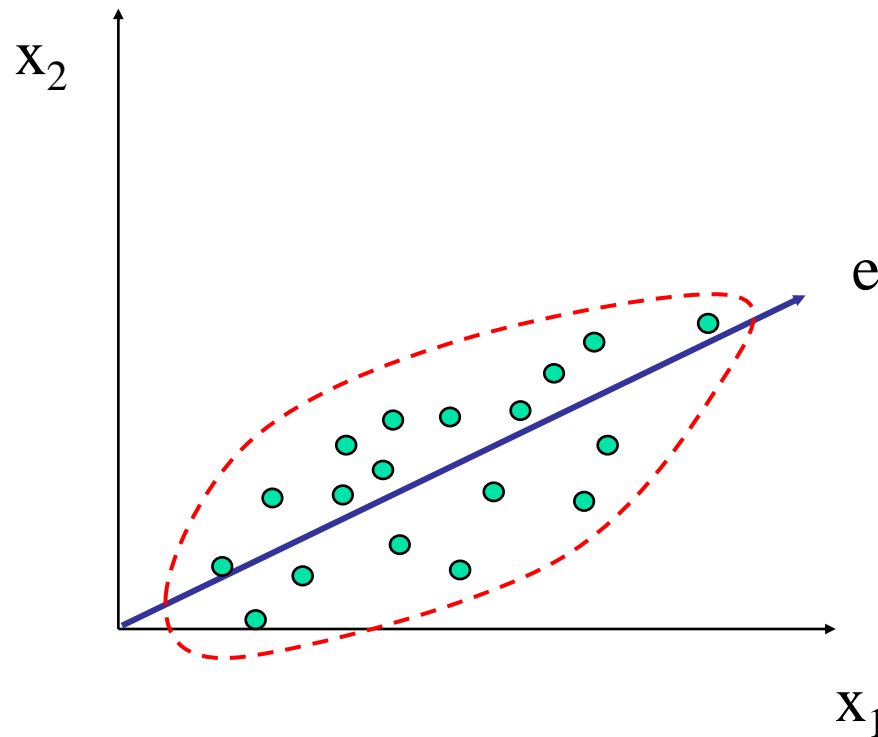


Why Wavelet Transform?

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

Principal Component Analysis (PCA)

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



Principal Component Analysis (Steps)

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

Attribute Subset Selection

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

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

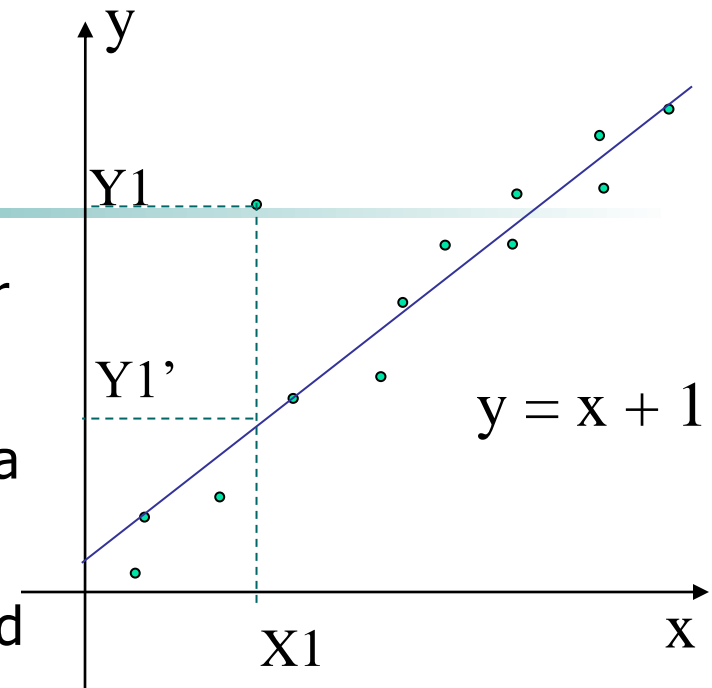
- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** (e.g., regression)
 - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - Ex.: Log-linear models—obtain value 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



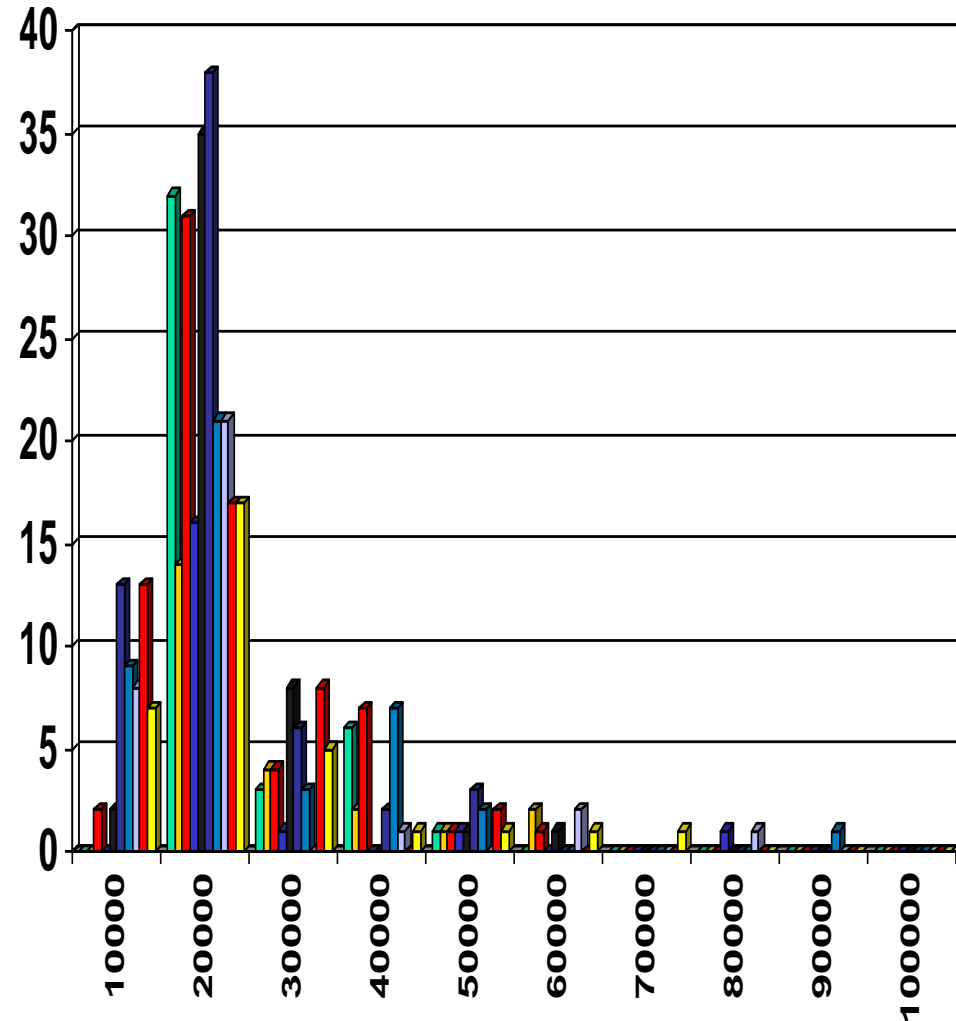
- Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships

Regress Analysis and Log-Linear Models

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

Histogram Analysis

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



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

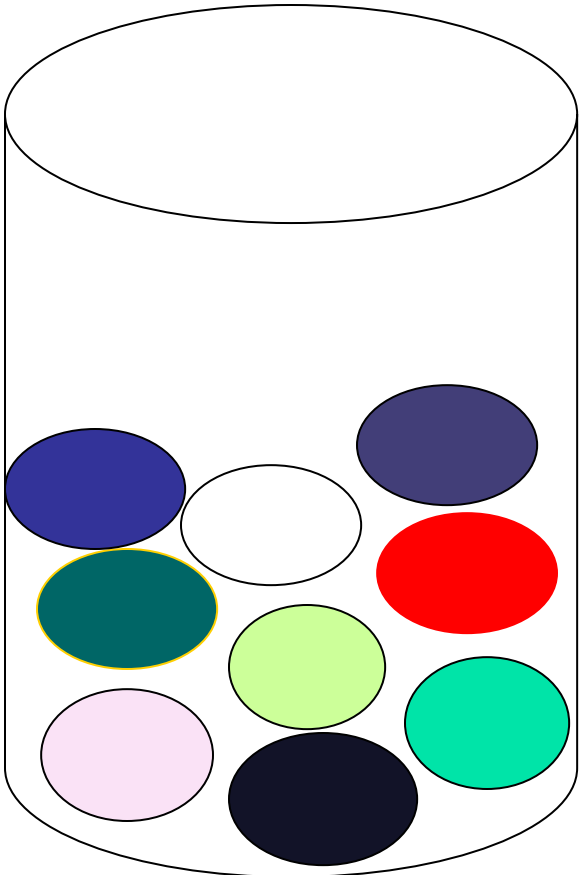
Sampling

- 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

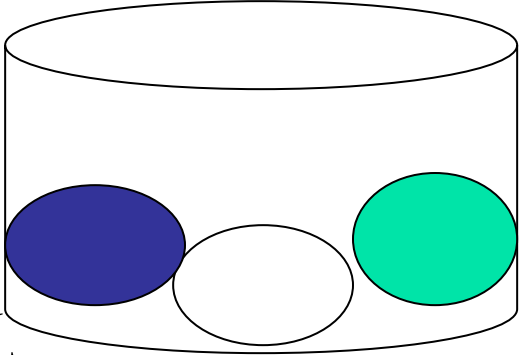
- **Simple random sampling**
 - There is an equal probability of selecting any particular item
- **Sampling without replacement**
 - Once an object is selected, it is removed from the population
- **Sampling with replacement**
 - A selected object is not removed from the population
- **Stratified sampling:**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

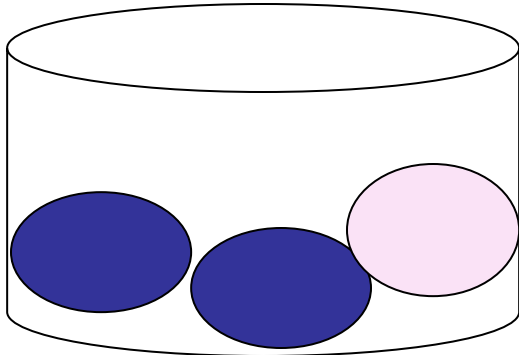


Raw Data

SRSWOR
(simple random
sample without
replacement)

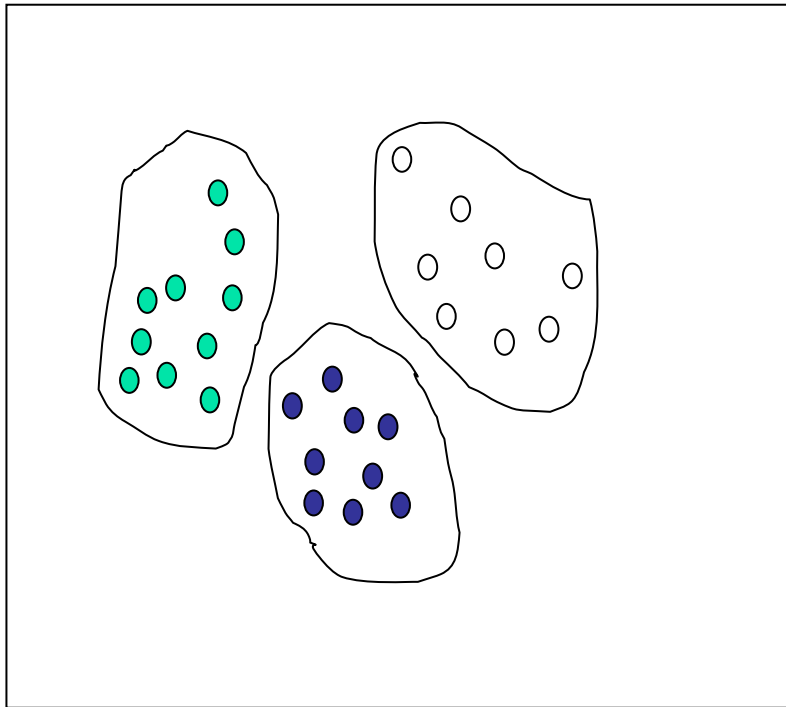


SRSWR

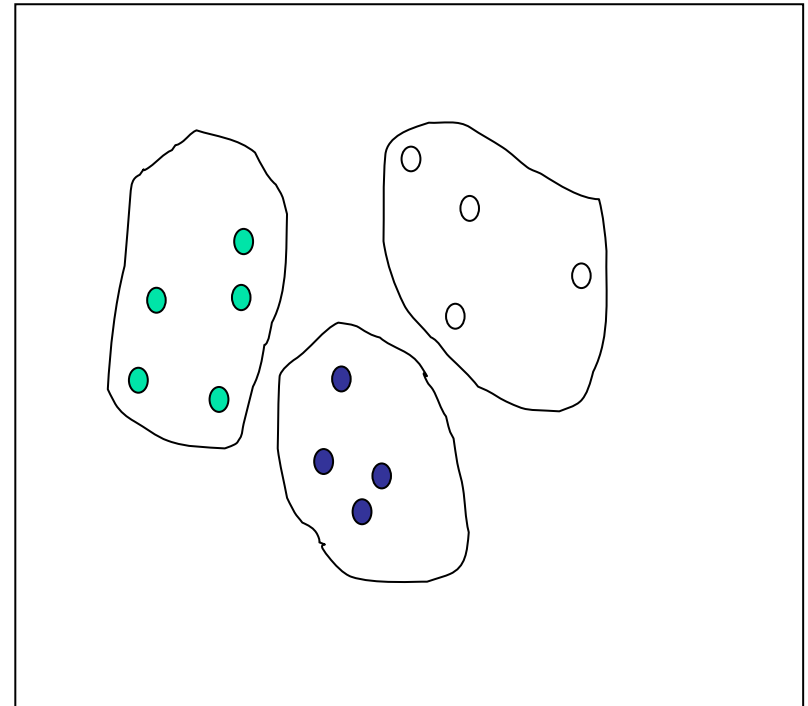


Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample



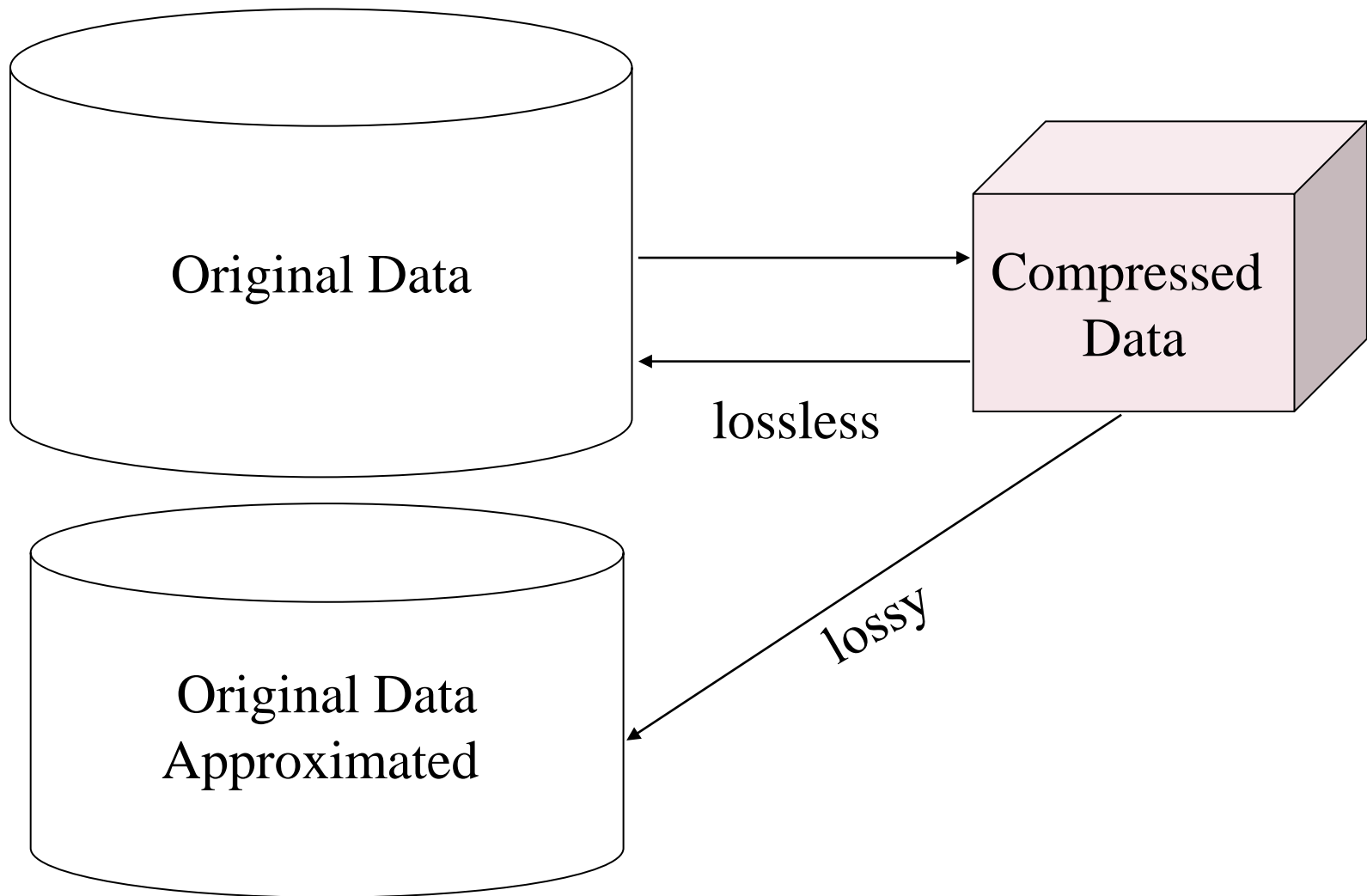
Data Cube Aggregation

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

Data Reduction 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

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**

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

Discretization

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

Data Discretization Methods

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

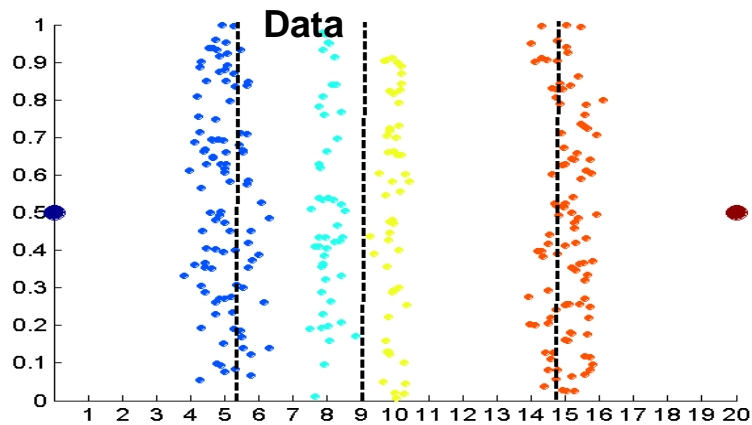
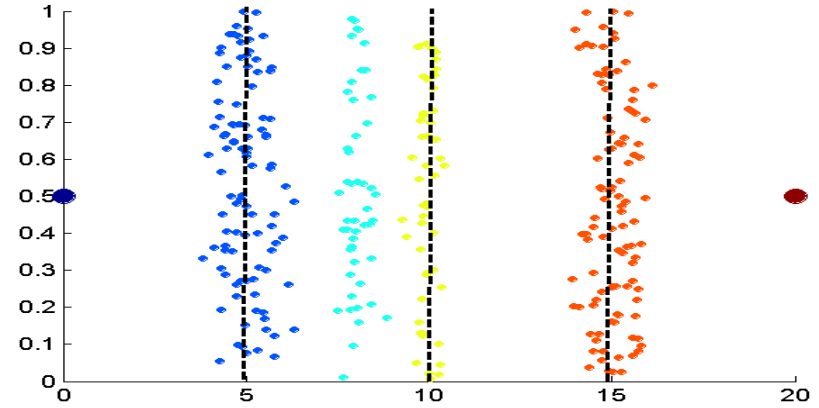
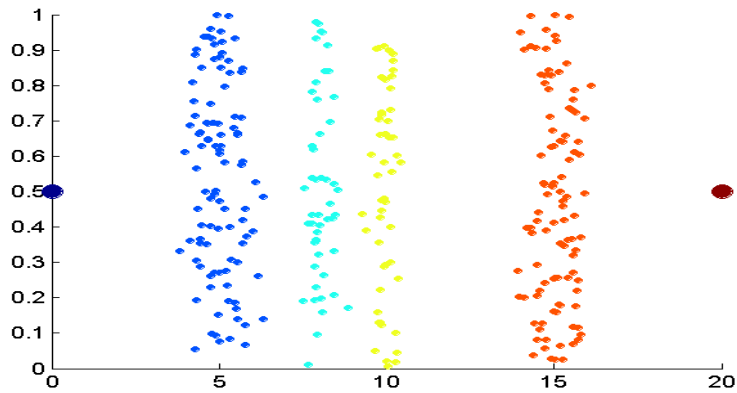
Simple Discretization: Binning

- **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

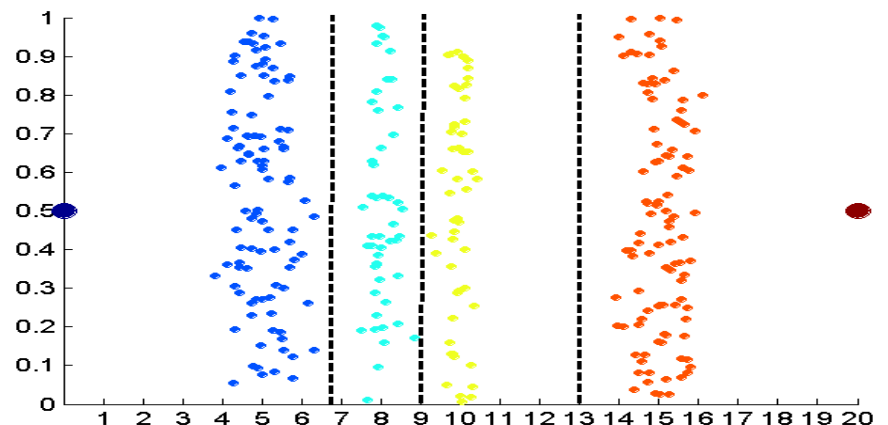
Binning Methods for Data Smoothing

- 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

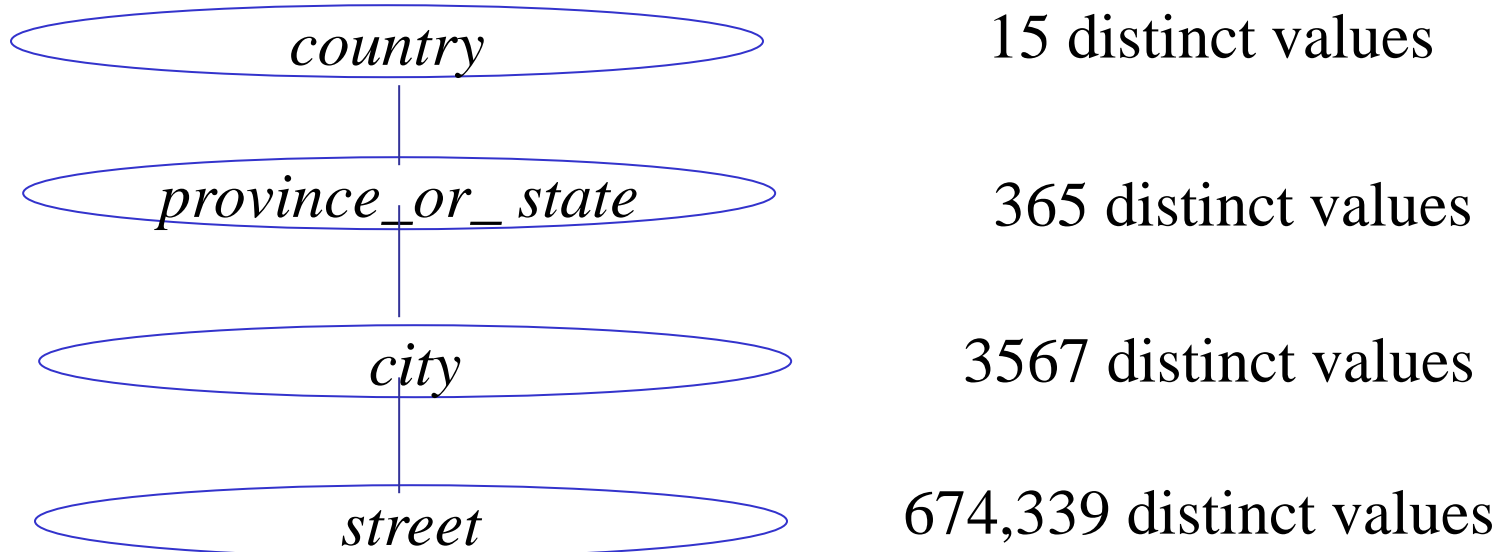
- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively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

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



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

Unit2: Data Quality and

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?

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

Missing Values,

- In statistics, **missing** data, or **missing values**, occur when no data **value** is stored for the variable in an observation. **Missing** data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.

Duplicate Data

- In computing, **data** deduplication is a specialized **data** compression technique for eliminating **duplicate** copies of repeating **data**. Related and somewhat synonymous terms are intelligent (**data**) compression and single-instance (**data**) storage.

Handling missing values in the dataset:

Few ways to handle missing values in the dataset:

- 1. Ignore the tuple:** This is usually done when the class label is missing (assuming the ~~mining task involves classification or description~~).
 - This method is not very effective, unless the tuple contains several attributes with missing values.
 - It is especially poor when the percentage of missing values per attribute varies considerably.
 - (A tuple is **one record (one row)**.)
- 2. Fill in the missing value manually:** In general, this approach is time-consuming and may not be feasible given a large data set with many missing values.
- 3. Use a global constant to fill in the missing value:** Replace all missing attribute values by the same constant, such as a label like \Unknown", or 1.

If missing values are replaced by, say, \Unknown", then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common | that of \Unknown".

Hence, although this method is simple, it is not recommended.

- 4. Use the attribute mean to fill in the missing value:** For example, suppose that the average income of AI Electronics customers is \$28,000. Use this value to replace the missing value for income.

- 5. Use the attribute mean for all samples belonging to the same class as the given tuple**

MISSING VALUES

Column 0 age years_seniority income parking_space attending_party entree pets emergency_contact

Column 0	age	years_seniority	income	parking_space	attending_party	entree	pets	emergency_contact
Tony	48	27	<input type="text"/>	1	5	shrimp	<input type="text"/>	Pepper
Donald	67	25	86	10	2	beef	<input type="text"/>	Jane
Henry	69	21	95	6	1	chicken	62	Janet
Janet	62	21	110	3	1	beef	<input type="text"/>	Henry
Nick	<input type="text"/>	17	<input type="text"/>	4	<input type="text"/>	<input type="text"/>		NA
Bruce	37	14	63	<input type="text"/>	1	veggie	<input type="text"/>	n/a
Steve	83	<input type="text"/>	77	7	1	chicken		None
Clint	27	9	118	9	<input type="text"/>	shrimp	3	empty
Wanda	19	7	52	2	2	shrimp	<input type="text"/>	_
Natasha	26	4	162	5	3	<input type="text"/>	<input type="text"/>	""
Carol	<input type="text"/>	3	127	11	1	veggie	1	null
Mandy	44	2	68	8	1	chicken	<input type="text"/>	

MISSING VALUES

Missing values

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q

How To Handle Duplicate Data

Multiple **records** for the same person or account signal that you have inaccurate or stale **data**, which leads to **bad** reporting, skewed metrics, and poor sender reputation. It can even result in different sales representatives calling on the same account.

Identifying and **removing** or merging these **duplicate records** from your database is a key part of forming an effective Single Customer View (SCV).

Results in complete version of the truth of your customer base allowing you to base strategic decisions on accurate **data**

Duplicate Data

	id	first_name	last_name	email
▶	1	Carine	Schmitt	carine.schmitt@verizon.net
	4	Janine	Labrune	janine.labrune@aol.com
	6	Janine	Labrune	janine.labrune@aol.com
	2	Jean	King	jean.king@me.com
	12	Jean	King	jean.king@me.com
	5	Jonas	Bergulfsen	jonas.bergulfsen@mac.com
	10	Julie	Murphy	julie.murphy@yahoo.com
	11	Kwai	Lee	kwai.lee@google.com
	3	Peter	Ferguson	peter.ferguson@google.com
	9	Roland	Keitel	roland.keitel@yahoo.com
	14	Roland	Keitel	roland.keitel@yahoo.com
	7	Susan	Nelson	susan.nelson@comcast.net
	13	Susan	Nelson	susan.nelson@comcast.net
	8	Zbyszek	Piestrzeniewicz	zbyszek.piestrzeniewicz@att.net

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)

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.

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

Figure 3.2 Binning methods for data smoothing.

How to Handle Noisy Data?

- Clustering

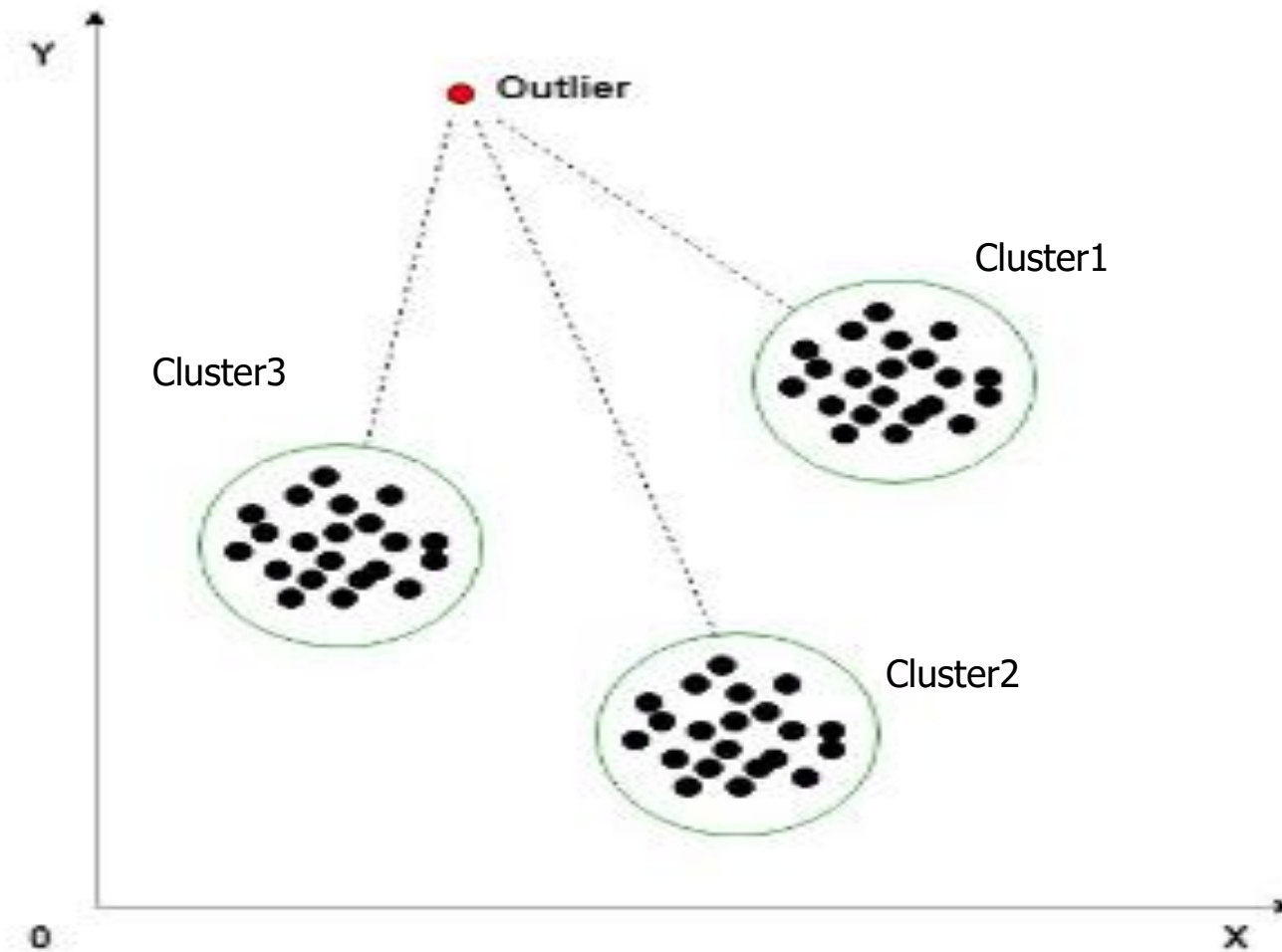
- detect and remove outliers

- An **outlier** is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

- **Why outlier analysis?**

Most data mining methods discard outliers noise or exceptions, however, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring one and hence, the outlier analysis becomes important in such case

Outlier Data



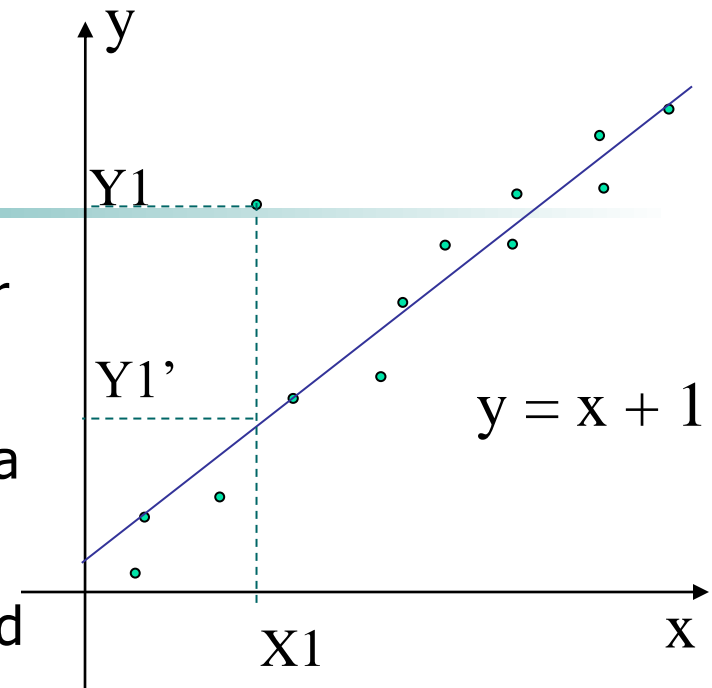
How To Handle and Detect outliers

Detecting Outlier:

- *Clustering based outlier detection using distance to the closest cluster:*
In the K-Means clustering technique, each cluster has a mean value.
- Objects belong to the cluster whose mean value is closest to it.
- In order to identify the Outlier, firstly we need to initialize the threshold value such that any distance of any data point greater than it from its nearest cluster identifies it as an outlier for our purpose.
- Then we need to find the distance of the test data to each cluster mean.
- Now, if the distance between the test data and the closest cluster to it is greater than the threshold value then we will classify the test data as an outlier.

How to Handle Noisy Data? 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

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)

DATA ANALYTICS



- Data Analytics Definition.
 - Data Analytics Types.
 - Data Analytics Life Cycle.
 - Analysis vs Reporting.
 - Tools & Environment.
 - Unstructured Data.
- 

Introduction to Data Analytics

- Data Analytics refers to the techniques used to analyze data to enhance productivity and business gain.
- Data is extracted from various sources and is cleaned and categorized to analyze various behavioral patterns.
- The techniques and the tools used vary according to the organization or individual.

What Is Data Analytics?

Data analytics is the science of analyzing raw data in order to make conclusions about that information. Many of the techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.

Data analytics techniques can reveal trends and metrics that would otherwise be lost in the mass of information. This information can then be used to optimize processes to increase the overall efficiency of a business or system.

Q.1 What is data analytics ? Why we need analytics ?

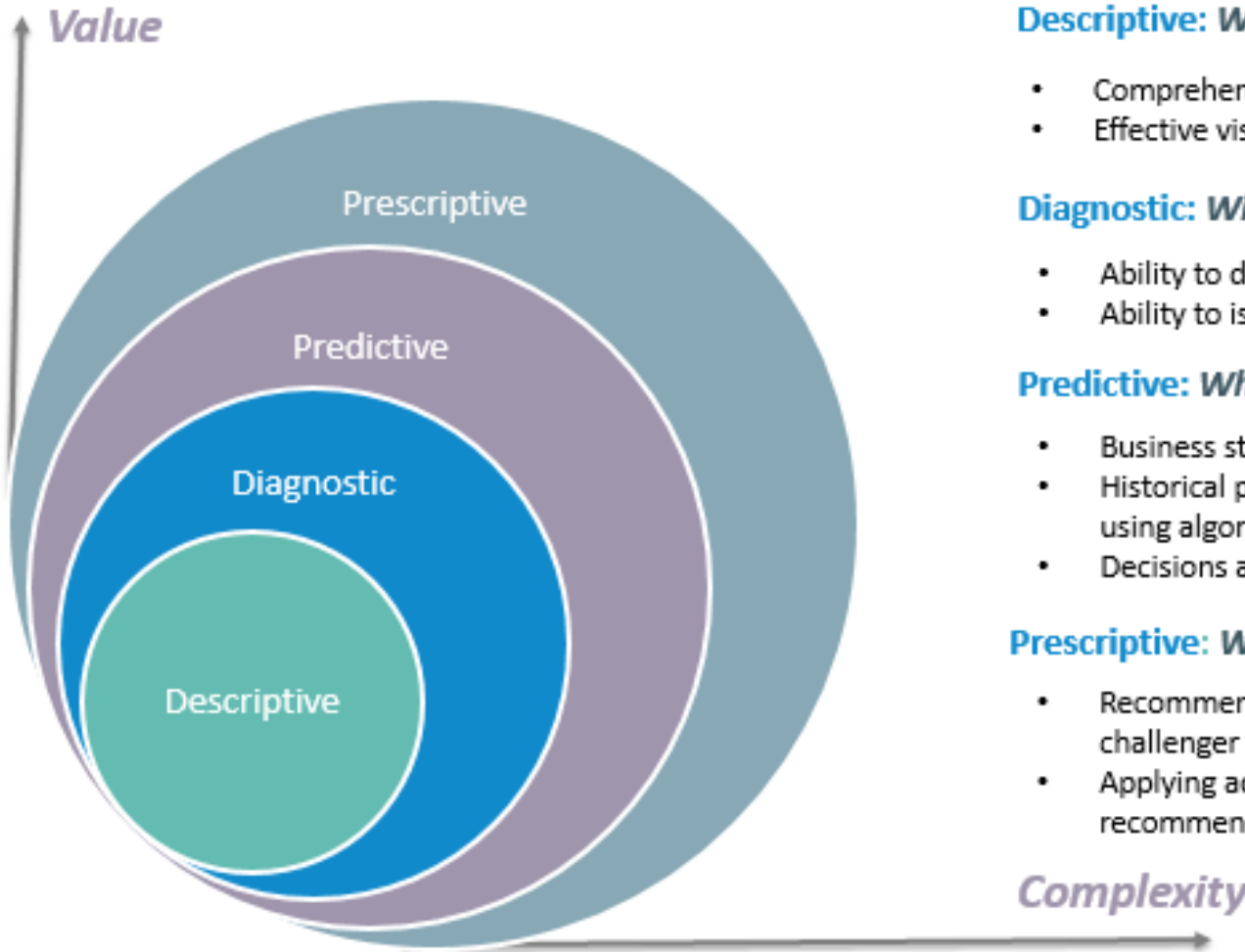
Ans. : • Data analytics is a process of inspecting, transforming and modeling a data.

- The overall purpose of big data analytics is to analyze large masses of data that will aid an organization in their decision making.
- Big data analytics is used to uncover unknown patterns, market trends, preferences of customers
- Big data analytics is the actual process of understanding and using data to benefit an organization.

- Need of data analytics :

1. Compete : Secure most powerful and unique competitive stronghold.
2. Growing sales revenues, retaining customers and discovering new customer bases.
3. Enforcing security policies by better managing, detecting and preventing fraud.
4. Improving operations and advancing your core business capacity to become more competitive.
5. Satisfying ever-accelerating customer expectations.
6. Learning vital insights that can enhance business decision-making.
7. Acting on insights to achieve desired outcomes.

4 types of Data Analytics



What is the data telling you?

Descriptive: *What's happening in my business?*

- Comprehensive, accurate and live data
- Effective visualisation

Diagnostic: *Why is it happening?*

- Ability to drill down to the root-cause
- Ability to isolate all confounding information

Predictive: *What's likely to happen?*

- Business strategies have remained fairly consistent over time
- Historical patterns being used to predict specific outcomes using algorithms
- Decisions are automated using algorithms and technology

Prescriptive: *What do I need to do?*

- Recommended actions and strategies based on champion / challenger testing strategy outcomes
- Applying advanced analytical techniques to make specific recommendations

Types of Data Analytics

Data analytics is broken down into four basic types.

1. **Descriptive analytics** describes what has happened over a given period of time. Have the number of views gone up? Are sales stronger this month than last?
2. **Diagnostic analytics** focuses more on why something happened. This involves more diverse data inputs and a bit of hypothesizing. Did the weather affect beer sales? Did that latest marketing campaign impact sales?
3. **Predictive analytics** moves to what is likely going to happen in the near term. What happened to sales the last time we had a hot summer? How many weather models predict a hot summer this year?
4. **Prescriptive analytics** suggests a course of action. If the likelihood of a hot summer is measured as an average of these five weather models is above 58%, we should add an evening shift to the brewery and rent an additional tank to increase output.

What and Why analytics:

Why Data Analytics?

01

Gather Hidden Insights

Generate Reports

02

03

Perform Market Analysis

Improve Business Requirement

04



What is Data Analytics?

Data Analytics refers to the techniques to analyse data to enhanced productivity and business gain.

Business
Administration



Exploratory Data
Analysis



Growth in Business



Who is a Data Analyst?



Introduction to tools and Environment

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

[Redacted]

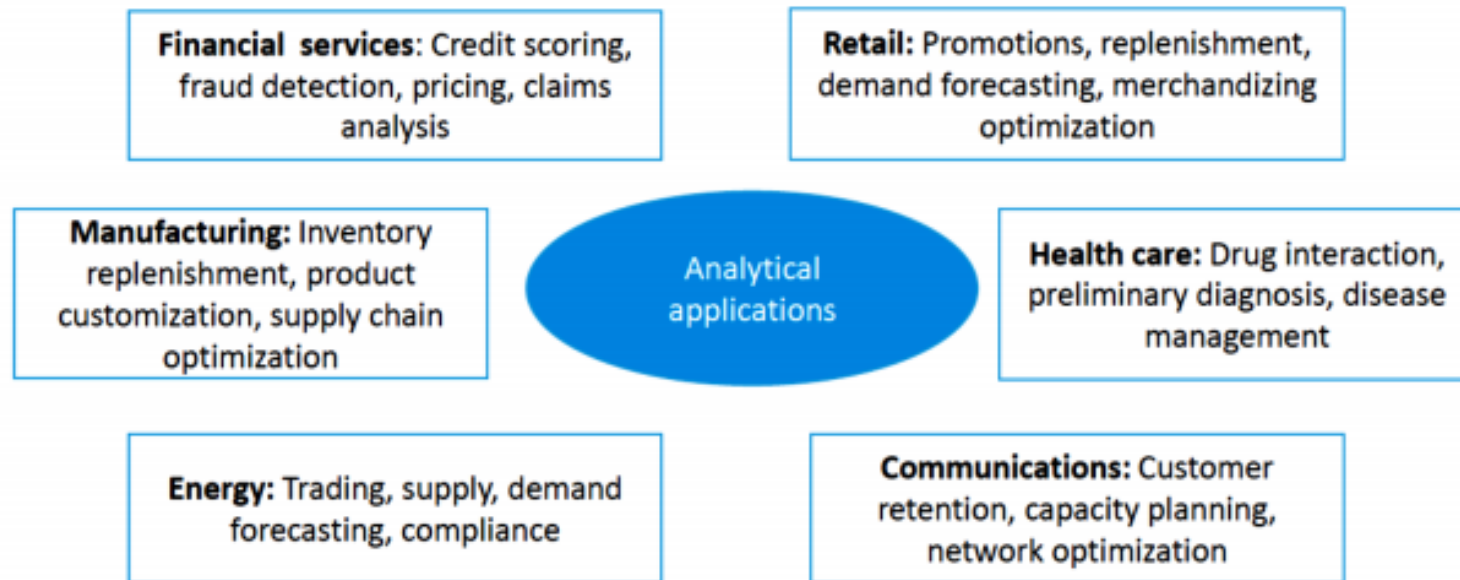
[Redacted]

[Redacted]



What and Why analytics:

Places where Analytics is used:



Q.2 Explain different types of data analytics.

Ans. : • Types of data analytics are descriptive, predictive and prescriptive.

1. Descriptive model

- It simple method and used in first phase of analytics, involves gathering, organizing,

tabulating and depicting data then the characteristics of what we are studying.

- The descriptive model shows relationships between the customer and product/service with the acquired data.
- This model can be used to organize a customer by their personal preferences for example.
- Descriptive statistics are useful to show things like, total stock in inventory, average dollars spent per customer and Year over year change in sales.
- Common examples of descriptive analytics are reports that provide historical insights regarding the company's production, financials, operations, sales, finance, inventory and customers

2. Predictive Analytics

- Predictive analytics helps your organization predict with confidence what will happen next so that you can make smarter decisions and improve business outcomes.
- The purpose of the predictive model is finding the likelihood different samples will perform in a specific way.
- The predictive model typically calculates live transactions multiple times to help evaluate the benefit of a customer transaction.
- Predictive models typically utilize a variety of variable data to make the prediction. The variability of the component data will have a relationship with what it is likely to predict
- Predictive analytics can be used throughout the organization, from forecasting customer behavior and purchasing patterns to identifying trends in sales activities.
- They also help forecast demand for inputs from the supply chain, operations and inventory.

3. Prescriptive

- This model suggests a course of action.
- The prescriptive model utilizes an understanding of what has happened, why it has happened and a variety of "what-might-happen" analysis to help the user determine the best course of action to take.
- A prescriptive analysis is typically not just with one individual response but is, in fact, a host of other actions.
- An example of this is a traffic application helping you choose the best route home and taking into account the distance of each route, the speed at which one can travel on each road and, crucially, the current traffic constraints.
- Another example might be producing an exam time-table such that no students have clashing schedules.
- Larger companies are successfully using prescriptive analytics to optimize production; scheduling and inventory in the supply chain to make sure that are delivering the right products at the right time and optimizing the customer experience.

Q.4 Difference between descriptive, predictive and prescriptive data analytics model.

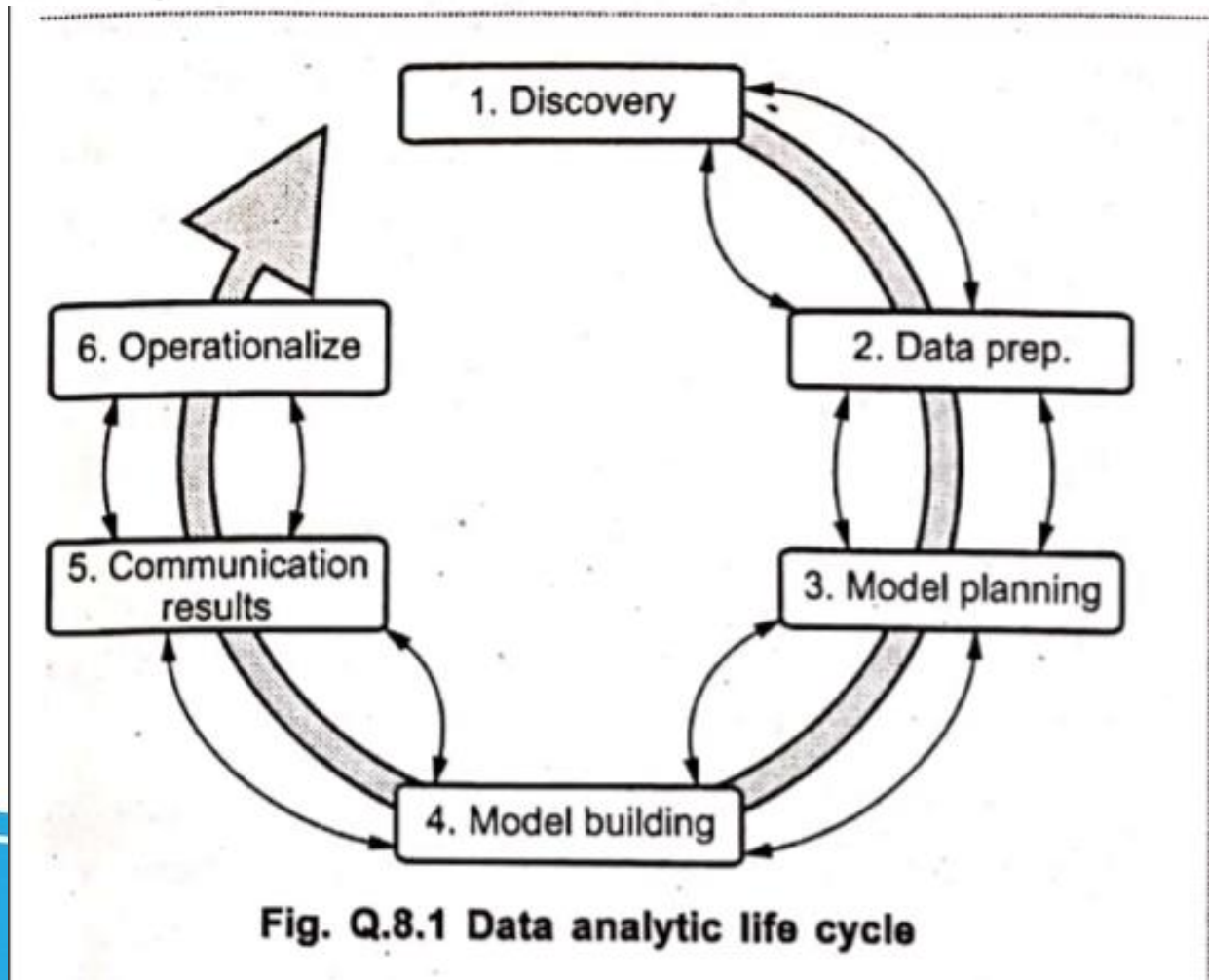
Ans. :

Descriptive Model	Predictive Model	Prescriptive Model
It use data aggregation and data mining to provide insight into the past and answer	Use statistical models and forecasts techniques to understand the future and answer	Use optimization and simulation algorithms to advice on possible outcomes and answer
What has happened ?	What could happened ?	What should we do ?

<p>Descriptive analytics is the analysis of past or historical data to understand trends and evaluate metrics over time.</p>	<p>Predictive analytics predicts future trends</p>	<p>Prescriptive analytics showcases viable solutions to a problem and the impact of considering a solution on future trend</p>
<p>Examples of tools used : Data aggregation and data mining.</p>	<p>Examples of tools used : Machine learning, statistical models, and simulation.</p>	<p>Examples of tools used : Optimization and heuristics</p>
<p>Used when user want to summarize results for all or part of your business.</p>	<p>Used when user want to make an educated guess at likely results.</p>	<p>Used when user have important, interdependent , complex or time-sensitive decisions to make.</p>
<p>Limitation : Snapshot of the past, often with limited ability to help guide decisions.</p>	<p>Limitation : Guess at the future, helps inform low complexity decisions.</p>	<p>Limitation : Most effective where user have some control over what is being modeled.</p>

Q.8 Describe data analytic life cycle model.

Ans. : • The data analytic lifecycle is designed for Big Data problems and data science projects. With six phases the project work can occur in several phases simultaneously. Fig. Q.8.1 shows data analytic life cycle model.



1. **Discovery** : In Phase 1, the team learns the business domain, including relevant history such as whether the organization or business unit has attempted similar projects in the past from which they can learn. The team assesses the resources available to support the project in terms of people, technology, time, and data.
2. **Data preparation** : Phase 2 requires the presence of an analytic sandbox, in which the team can work with data and perform analytics for the duration of the project. The team needs to execute extract, load, and transform (ELT) or extract, transform and load (ETL) to get data into the sandbox.
3. **Model planning** : The team determines the methods, techniques, and workflow it intends to follow for the subsequent model building phase.

4. **Model building** : In Phase 4, the team develops datasets for testing, training, and production purposes. In addition, in this phase the team builds and executes models based on the work done in the model planning phase.
5. **Communicate results** : In Phase 5, the team, in collaboration with major stakeholders, determines if the results of the project are a success or a failure based on the criteria developed in Phase 1.
6. **Operationalize** : In Phase 6, the team delivers final reports, briefings, code, and technical documents. In addition, the team may run a pilot project to implement the models in a production environment.

Q.9 What is difference between reporting and analysis ?

Ans. :

Sr. No.	Reporting	Analysis
1.	The process of organizing data into informational summaries in order to monitor how different areas of a business are performing.	The process of exploring data and reports in order to extract meaningful insights, which can be used to better understand and improve business performance.
2.	Reporting translates raw data into information.	Analysis transforms data and information into insights.


3.	Reporting helps companies to monitor their online business and be alerted to when data falls outside of expected ranges.	The goal of analysis is to answer questions by interpreting the data at a deeper level and providing actionable recommendations.
4.	Good reporting should raise questions about the business from its end users.	Through the process of performing analysis you may raise additional questions, but the goal is to identify answers, or at least potential answers that can be tested.
5.	Reporting shows you what is happening.	Analysis focuses on explaining why it is happening and what you can do about it.

2.3 : Application of Modelling in Business

Q.15 Write short note on : Application of modelling in business.

Ans. : • A business model is a framework for how a company creates value.


- A business model captures the fundamental assumptions and any key learnings about a new venture. For example, it might enumerate the company's core value proposition, targeting customers, key resources, and assumed revenue streams.
- Statistical model represents a set of assumptions concerning the generation of the observed data, and similar data from a larger population.
- In data generating process, model is considered as ideal form.
- Signal processing is an enabling technology that encompasses the fundamental theory, applications, algorithms.

- It also implements the processing or transferring information in many different physical, symbolic, or abstract formats broadly designated as signals.
 - It uses mathematical, statistical and computational representations.
 - In manufacturing statistical models are used to define Warranty policies, solving various conveyor related issues, Statistical Process Control etc.
- 

Q.16 What are the components of a business model ?

Ans. : Detailed list of components that are found in most business models :

1. Problem : The target customers' pain points
2. Solution : How the company intends to meet the customers' needs (aka the product)
3. Key Resources : Physical, intellectual, human, and financial assets at the company
4. Customer segments : Who are the target customers
5. Unique value proposition : Why the customer is willing to buy the solution
6. Competitive landscape : What alternatives can customers use
7. Competitive advantage : Characteristics not easily copied or bought elsewhere

8. Sales channels : How the company will reach customers
 9. Revenue streams : How the company generates income
 10. Revenue model : Framework for how the company will be profitable
 11. Key partners : Partners and suppliers essential to the business
 12. Cost structure : What are the company's costs and how should that affect pricing
 13. Key metrics : How the company measures success
- 

Q.20 What is unstructured data ?

Ans. : • Unstructured data has internal structure but is not structured via pre-defined data models or schema.

- It may be textual or non-textual, and human-or machine-generated. It may also be stored within a non-relational database like NoSQL.
- Unstructured data continues to grow in influence in the enterprise as organizations try to leverage new and emerging data sources.
- These new data sources are made up largely of streaming data coming from social media platforms, mobile applications, location services, and Internet of Things technologies.
- Since the diversity among unstructured data sources is so prevalent, businesses have much more trouble managing it than they do with old-school structured data.
- As a result, companies are being challenged in a way they weren't before, and are having to get creative in order to pull relevant data for analytics.

• Typical human-generated unstructured data includes :


1. **Text files** : Word processing, spreadsheets, presentations, email, logs.
2. **Email** : Email has some internal structure thanks to its metadata, and we sometimes refer to it as semi-structured. However, its message field is unstructured and traditional analytics tools cannot parse it.
3. **Social Media** : Data from Facebook, Twitter, LinkedIn.
4. **Website** : YouTube, Instagram, photo sharing sites.
5. **Mobile data** : Text messages, locations.
6. **Communications** : Chat, IM, phone recordings, collaboration software.
7. **Media** : MP3, digital photos, audio and video files.
8. **Business applications** : MS Office documents, productivity applications.

Typical machine-generated unstructured data includes :

2 -

1. **Satellite imagery** : Weather data, land forms, military movements.
2. **Scientific data** : Oil and gas exploration, space exploration, seismic imagery, atmospheric data.
3. **Digital surveillance** : Surveillance photos and video.
4. **Sensor data** : Traffic, weather, oceanographic sensors

R Studio

- R Studio is an IDE for R with advanced and more user-friendly GUI.
 - R is the substrate on which we can mount various features using PACKAGES like RCMDR- R Commander or R-Studio.
 - R was started by Bell Laboratories as “S” for Fortran Library This tool is the leading analytics tool used for statistics and data modeling.
 - R compiles and runs on various platforms such as UNIX, Windows, and Mac OS.
 - It also provides tools to automatically install all packages as per user-requirement.
- 

Look at R!



The screenshot shows the R Commander application window. The menu bar includes File, Data, Statistics, Graphs, Models, Distributions, and Help. The 'Data' menu is open, showing options like Summaries, Contingency tables, Means, Proportions, Variances, Nonparametric tests, Dimensional analysis, and Fit models. The 'Fit models' sub-menu is also open, showing Linear regression..., Linear model..., and Generalized linear model... The main workspace shows a script with the following code:

```
View data set Log commands: Attach active data set:
ich/survey/data/psy3610/classsurvey.sav", x
```

At the bottom, the Model pane shows "(No active model)".

R-Commander Interface



The screenshot shows the R-Studio application window. The source editor contains the following R script:

```
1 library(maps)
2 library(ggplot2)
3 data(us.cities)
4
5 aplot(long, lat, data = choro, group = group,
6 fill = assault, geom = "polygon")
7 aplot(long, lat, data = choro, group = group,
8 fill = assault / murder, geom = "polygon")
```

The Environment pane on the right shows the following data objects:

Data	Dimensions
Capitals	2x47 double matrix
Distance	47x47 double matrix
Host	12x47 double matrix
LanguageDistance	47x47 double matrix
Multilingual	12x47 double matrix
Neighbours	47x47 double matrix

The Console pane shows the execution of the script:

```
> choro <- choro[order(choro$order), ]
> aplot(long, lat, data = choro, group = group,
+ fill = assault, geom = "polygon")
> aplot(long, lat, data = choro, group = group,
+ fill = assault / murder, geom = "polygon")
> aplot(long, lat, data = choro, group = group,
fill = assault, geom = "polygon")
```

The Plots pane shows a choropleth map of the United States with a legend on the right. The legend is labeled 'long' and has a color scale from 0 to 300.

R-Studio Interface

Understanding components of R

1. Data Type:

There are two types of data classified on very broad level. They are Numeric and Character data.


- Numeric Data: - It includes 0~9, “.” and “- ve” sign.
- Character Data: - Everything except Numeric data type is Character. For Example, Names, Gender etc.

Data is also classified as Quantitative and Qualitative.

For Example, “1,2,3...” are Quantitative Data while “Good”, “Bad” etc. are Qualitative Data.

Although we can convert Qualitative Data into Quantitative Data using Ordinal Values.

For Example, “Good” can be rated as 9 while “Average” can be rated as 5 and “Bad” can be rated as 0.



2. Data Frame:

A data frame is used for storing data tables. It is a list of vectors of equal length.

For example, here is a built-in data frame in R, called **mtcars**.

```
> mtcars
      mpg  cyl  disp  hp  drat   wt
Mazda RX4    21.0   6  160 110 3.90 2.62
Mazda RX4 Wag 21.0   6  160 110 3.90 2.88
Datsun 710    22.8   4  108  93 3.85 2.32
```

The top line of the table, called the header, contains the column names. Each horizontal line afterward denotes a data row, which begins with the name of the row, and then followed by the actual data. Each data member of a row is called a cell. To retrieve data in a cell, we would enter its row and column coordinates in the single square bracket "[" operator. The two coordinates are separated by a comma. In other words, the coordinates begins with row position, then followed by a comma, and ends with the column position. The order is important.

For Example,

Here is the cell value from the first row, second column of **mtcars**.

```
> mtcars[1, 2]
[1] 6
```

3. Array and Matrices:

We have two different options for constructing matrices or arrays. Either we use the creator functions **matrix ()** and

Array (), or you simply change the dimensions using the **dim ()** function.

For example, you make an array with four columns, three rows, and two “tables” like this:

```
[CODE] >my.array<- array(1:24, dim=c(3,4,2))
```

In the above example, “my.array” is the name of the array we have given. And “←” is the assignment operator.

There are 24 units in this array mentioned as “1:24” and are divided in three dimensions “(3, 4, 2)”.

Note: - Although the rows are given as the first dimension, the tables are filled column-wise. So, for arrays, R fills the columns, then the rows, and then the rest.

Alternatively, you could just add the dimensions using the **dim ()** function. This is a little hack that goes a bit faster than using the array () function; it’s especially useful if you have your data already in a vector. (This little trick also works for creating matrices, by the way, because a matrix is nothing more than an array with only two dimensions.)

Say you already have a vector with the numbers 1 through 24, like this:

```
[CODE] >my.vector<- 1:24
```

You can easily convert that vector to an array exactly like my.array simply by assigning the dimensions, like this:

```
[CODE] > dim(my.vector) <- c(3,4,2)
```

Reading Database using R

We can import Datasets from various sources having various files types for example,

- .csv format
- Big data tool – Impala
- **CSV File**

The sample data can also be in comma separated values (CSV) format. Each cell inside such data file is separated by a special character, which usually is a comma, although other characters can be used as well. The first row of the data file should contain the column names instead of the actual data. Here is a sample of the expected format.

```
Col1,Col2,Col3
100,a1,b1
200,a2,b2
300,a3,b3
```

After we copy and paste the data above in a file named "mydata.csv" with a text editor, we can read the data with the function read.csv.

After we copy and paste the data above in a file named "mydata.csv" with a text editor, we can read the data with the function read.csv.

[CODE]

```
> mydata = read.csv("mydata.csv") # read csv file
> mydata

  Col1 Col2 Col3
1  100  a1  b1
2  200  a2  b2
3  300  a3  b3
```

In various European locales, as the comma character serves as the decimal point, the function read.csv2 should be used instead. For further detail of the read.csv and read.csv2 functions, please consult the R documentation.

```
> help(read.csv)
➤ Big data tool – Impala
```

Cloudera 'Impala', which is a massively parallel processing (MPP) SQL query engine runs natively in Apache Hadoop.

R package, **RImpala**, connects Impala to R.

Introduction to Tools and Environment cont..



Python

Python is an open-source, object-oriented programming language that is easy to read, write, and maintain. It provides various machine learning and visualization libraries such as Scikit-learn, TensorFlow, Matplotlib, Pandas, Keras, etc. It also can be assembled on any platform like SQL server, a MongoDB database or JSON

Tableau Public

- This is a free software that connects to any data source such as Excel, corporate Data Warehouse, etc. It then creates visualizations, maps, dashboards etc with real-time updates on the web.

QlikView

- This tool offers in-memory data processing with the results delivered to the end-users quickly. It also offers data association and data visualization with data being compressed to almost 10% of its original size.



A programming language and environment for data manipulation and analytics, this tool is easily accessible and can analyze data from different sources.

Other Tools

- **Microsoft Excel** – This tool is one of the most widely used tools for data analytics. Mostly used for clients' internal data, this tool analyzes the tasks that summarize the data with a preview of pivot tables.
- **RapidMiner** – A powerful, integrated platform that can integrate with any data source types such as Access, Excel, Microsoft SQL, Tera data, Oracle, Sybase etc. This tool is mostly used for predictive analytics, such as data mining, text analytics, [machine learning](#)
- **KNIME** – Konstanz Information Miner (KNIME) is an open-source data analytics platform, which allows you to analyze and model data. With the benefit of visual programming, KNIME provides a platform for reporting and integration through its modular data pipeline concept.
- **OpenRefine** – Also known as GoogleRefine, this data cleaning software will help you clean up data for analysis. It is used for cleaning messy data, the transformation of data and parsing data from websites.

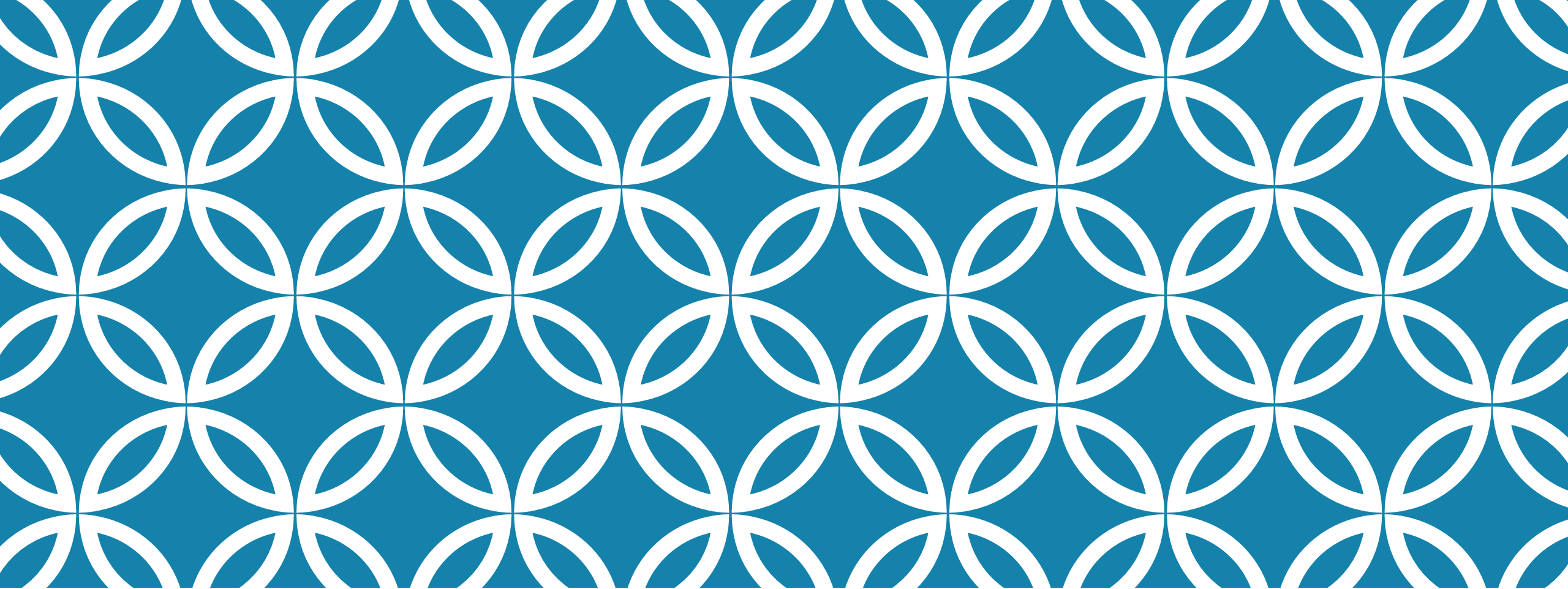
Other Tools

- [Apache Spark](#) – One of the largest large-scale data processing engine, this tool executes applications in Hadoop clusters 100 times faster in memory and 10 times faster on disk. This tool is also popular for data pipelines and machine learning model development.

Top Tools and Environment in Data Analytics

- **R programming** – This tool is the leading analytics tool used for statistics and data modeling. R compiles and runs on various platforms such as UNIX, Windows, and Mac OS. It also provides tools to automatically install all packages as per user-requirement.
- **Python** – Python is an open-source, **object-oriented programming** language which is easy to read, write and maintain. It provides various machine learning and visualization libraries such as **Scikit-learn**, **TensorFlow**, **Matplotlib**, **Pandas**, **Keras** etc. It also can be assembled on any platform like SQL server, a **MongoDB** database or **JSON**
- **Tableau Public** – This is a free software that connects to any data source such as Excel, **corporate Data Warehouse** etc. It then creates visualizations, maps, dashboards etc with real-time updates on the web.
- **QlikView** – This tool offers in-memory data processing with the results delivered to the end-users quickly. It also offers data association and data visualization with data being compressed to almost 10% of its original size.
- **SAS** – A programming language and environment for data manipulation and analytics, this tool is easily accessible and can analyze data from different sources.
- **Microsoft Excel** – This tool is one of the most widely used tools for data analytics. Mostly used for clients' internal data, this tool analyzes the tasks that summarize the data with a preview of pivot tables.

- **RapidMiner** – A powerful, integrated platform that can integrate with any data source types such as Access, Excel, Microsoft SQL, Tera data, Oracle, Sybase etc. This tool is mostly used for predictive analytics, such as data mining, text analytics, machine learning.
- **KNIME** – Konstanz Information Miner (KNIME) is an open-source data analytics platform, which allows you to analyze and model data. With the benefit of visual programming, KNIME provides a platform for reporting and integration through its modular data pipeline concept.
- **OpenRefine** – Also known as GoogleRefine, this data cleaning software will help you clean up data for analysis. It is used for cleaning messy data, the transformation of data and parsing data from websites.
- **Apache Spark** – One of the largest large-scale data processing engine, this tool executes applications in Hadoop clusters 100 times faster in memory and 10 times faster on disk. This tool is also popular for data pipelines and machine learning model development.



UNIT-2-DATA ANALYTICS

Data Analytics Modelling Techniques, Types of databases

DATA MODELLING

Data modeling is a set of tools and techniques used to understand and analyse how an organisation should collect, update, and store data.

It is a critical skill for the business analyst who is involved with discovering, analysing, and specifying changes to how software systems create and maintain information.

data modelling is nothing but a process through which data is stored structurally in a format in a database.

Data modelling is important because it enables organizations to make data-driven decisions and meet varied business goals.

DATA MODELING SOMETIMES NEEDS DATA ANALYSIS

Business Analysts often need to analyze data as part of making data modeling decisions, and this means that data modeling can include some amount of data analysis.

A lot can be accomplished with very basic technical skills, such as the ability to run simple database queries.

The primary goal of using data model are:

- Ensures that all data objects required by the database are accurately represented. Omission of data will lead to creation of faulty reports and produce incorrect results.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data Model structure helps to define the relational tables, primary and foreign keys and stored procedures.
- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.
- Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

WHY USE DATA MODEL?

- Data Modelling helps create a robust design with a data model that can show an organization's entire data on the same platform.
- The data model makes sure that all the data objects required by the database are represented or not.
- The database at the logical, physical, and conceptual levels can be designed with the help data model.
- The relation tables, foreign keys, and primary keys can be defined with the data model's help.
- Data Modelling Tools help in the improvement of data quality.
- Data Model gives the clear picture of business requirements.
- Redundant data and missing data can be identified with the help of data models.
- In data models, all the important data is accurately represented. The chances of incorrect results and faulty reports decreased as the data model reduces data omission.
- The data models create a visual representation of the data. With the help of it, the data analysis gets improved. We get the data picture, which can then be used by developers to create a physical database.
- Better consistency can be qualified with the help of a data model across all the projects.
- The data model is quite a time consuming, but it makes the maintenance cheaper and faster.

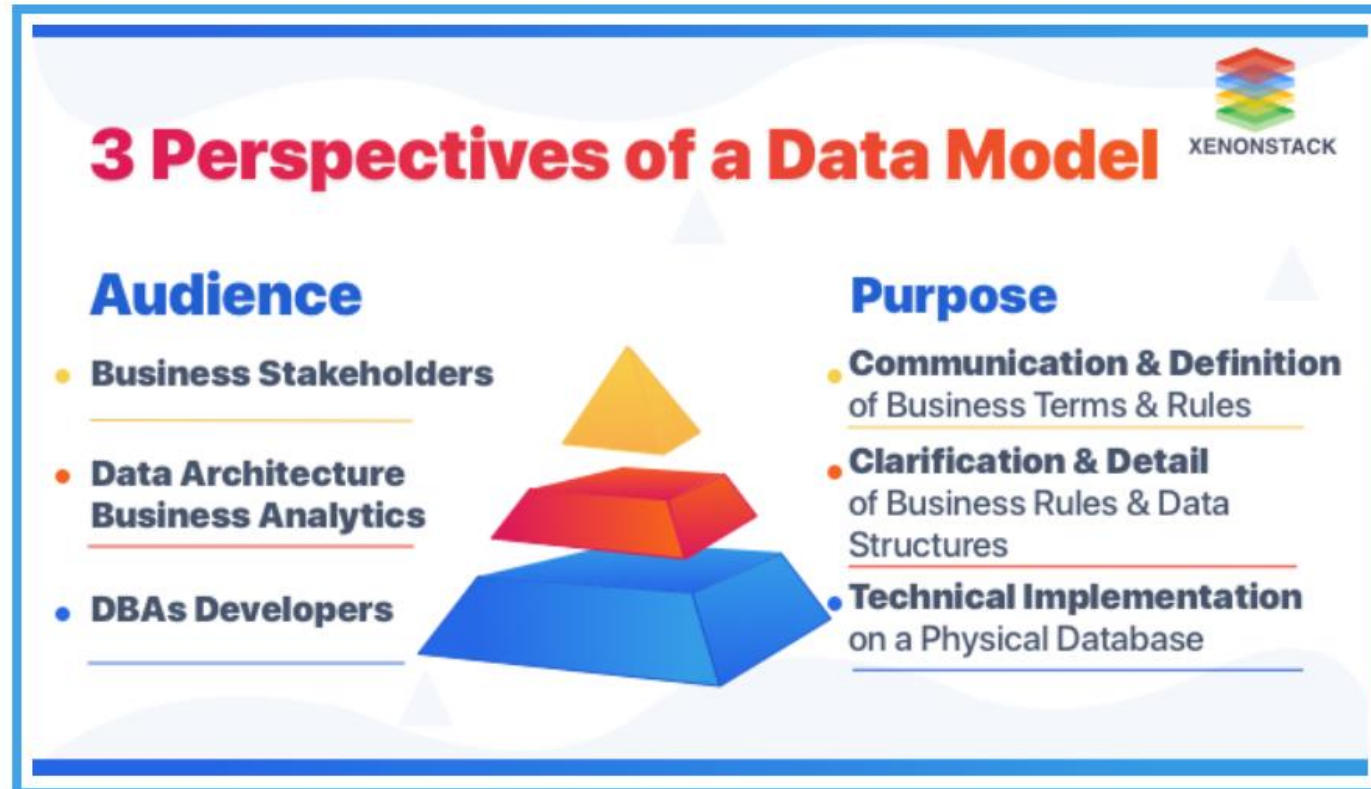
ROLES OF DATA MODELLER

- They create an entity relationship diagram to visualize relationships between key business concepts.
- They create a conceptual-level data dictionary to communicate data requirements that are important to business stakeholders.
- They create a data map to resolve potential data issues for a data migration or integration project.

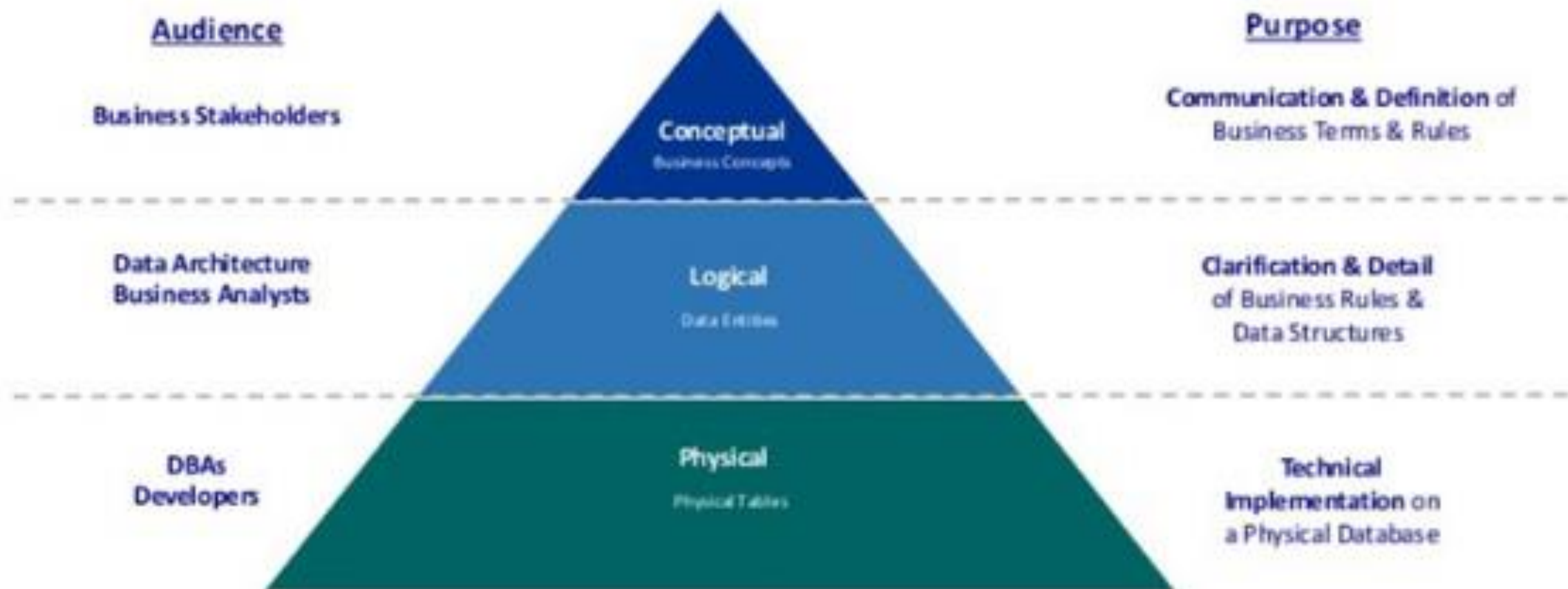
A data modeller would not necessarily query or manipulate data or become involved in designing or implementing databases or data repositories.

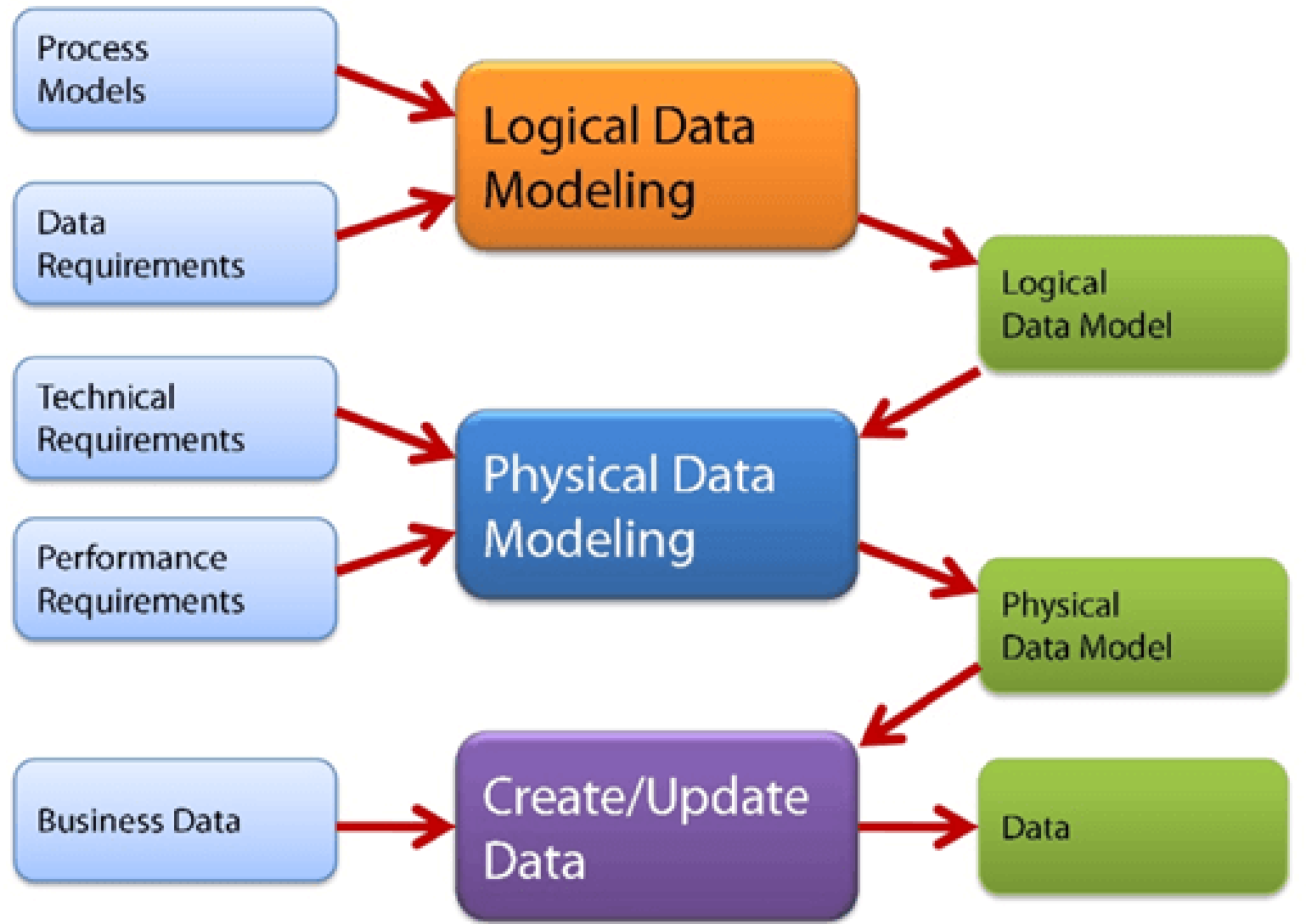
Three Perspectives of a Data Model

Data Modelling helps to create a conceptual model and create the relationship between the items. The basic data modelling techniques involve dealing with three perspectives of a data model.



Levels of Data Modeling





Types of Data Model

Data Model

The **Data Model** is defined as an abstract model that organizes data description, data semantics, and consistency constraints of data. The data model emphasizes on what data is needed and how it should be organized instead of what operations will be performed on data. Data Model is like an architect's building plan, which helps to build conceptual models and set a relationship between data items.

The two types of Data Modeling Techniques are

1. Entity Relationship (E-R) Model
2. UML (Unified Modelling Language)

Why use Data Model?

The primary goal of using data model are:

- Ensures that all data objects required by the database are accurately represented. Omission of data will lead to creation of faulty reports and produce incorrect results.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data Model structure helps to define the relational tables, primary and foreign keys and stored procedures.
- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.
- Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

TYPES OF DATA MODELS

1. conceptual data model
2. logical data model
3. physical data model

There are three different types of data models are produced while progressing from requirements to the actual database to be used for the information system.

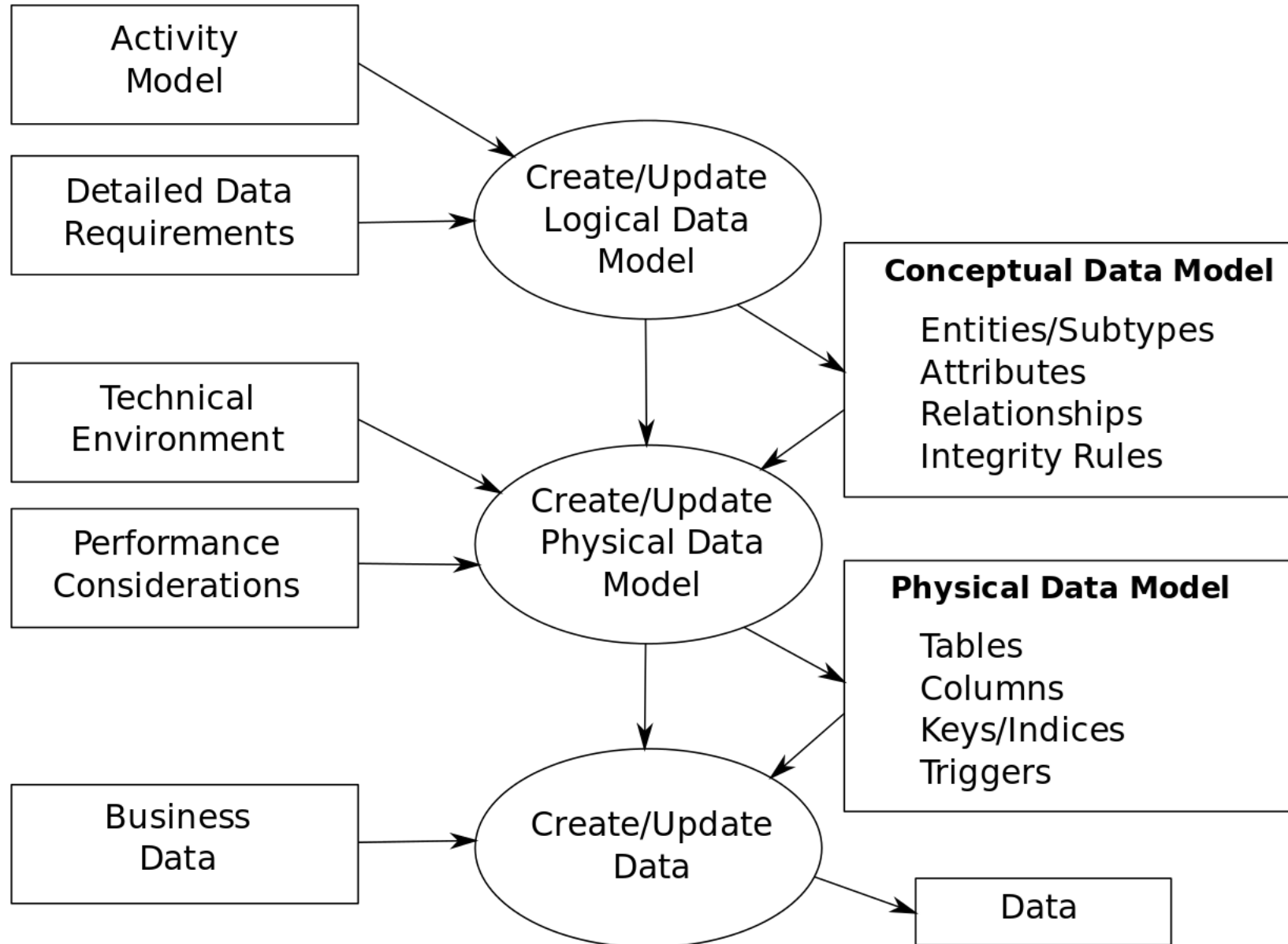
The data requirements are initially recorded as a **conceptual data model** which is essentially a set of technology independent specifications about the data and is used to discuss initial requirements with the business stakeholders.

The **conceptual model** is then translated into a **logical data model**, which documents structures of the data that can be implemented in databases.

Implementation of one conceptual data model may require multiple logical data models.

The last step in data modeling is transforming the logical data model to a **physical data model** that organizes the data into tables, and accounts for access, performance and storage details.

Data modeling defines not just data elements, but also their structures and the relationships between them.



Conceptual Data Model:

A **Conceptual Data Model** is an organized view of database concepts and their relationships. The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships. In this data modeling level, there is hardly any detail available on the actual database structure. Business stakeholders and data architects typically create a conceptual data model.

The 3 basic tenants of Conceptual Data Model are

- **Entity:** A real-world thing
- **Attribute:** Characteristics or properties of an entity
- **Relationship:** Dependency or association between two entities

Data model example:

- Customer and Product are two entities. Customer number and name are attributes of the Customer entity
- Product name and price are attributes of product entity
- Sale is the relationship between the customer and product



Characteristics of a conceptual data model:

- Offers Organisation-wide coverage of the business concepts.
 - This type of Data Models are designed and developed for a business audience.
 - The conceptual model is developed independently of hardware specifications like data storage capacity, location or software specifications like DBMS vendor and technology. The focus is to represent data as a user will see it in the “real world.”
- Conceptual data models known as Domain models create a common vocabulary for all stakeholders by establishing basic concepts and scope.

Logical Data Model:

The **Logical Data Model** is used to define the structure of data elements and to set relationships between them. The logical data model adds further information to the conceptual data model elements. The advantage of using a Logical data model is to provide a foundation to form the base for the Physical model. However, the modeling structure remains generic.



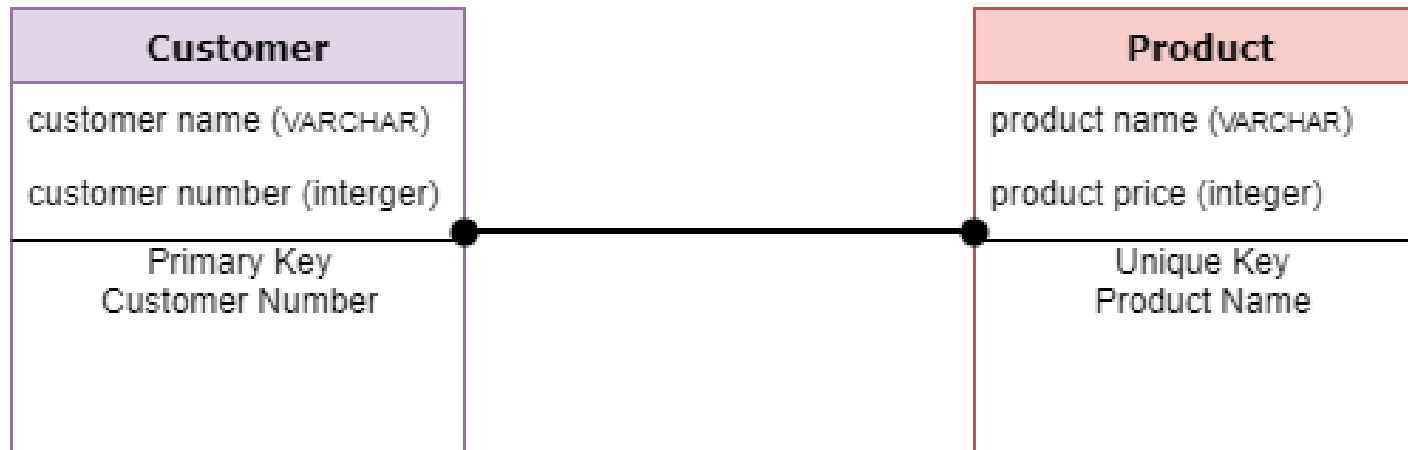
At this Data Modeling level, no primary or secondary key is defined. At this Data modeling level, you need to verify and adjust the connector details that were set earlier for relationships.

Characteristics of a Logical data model:

- Describes data needs for a single project but could integrate with other logical data models based on the scope of the project.
- Designed and developed independently from the DBMS.
- Data attributes will have datatypes with exact precisions and length.
- Normalization processes to the model is applied typically till 3NF.

Physical Data Model

A **Physical Data Model** describes a database-specific implementation of the data model. It offers database abstraction and helps generate the schema. This is because of the richness of meta-data offered by a Physical Data Model. The physical data model also helps in visualizing database structure by replicating database column keys, constraints, indexes, triggers, and other RDBMS features.



Characteristics of a physical data model:

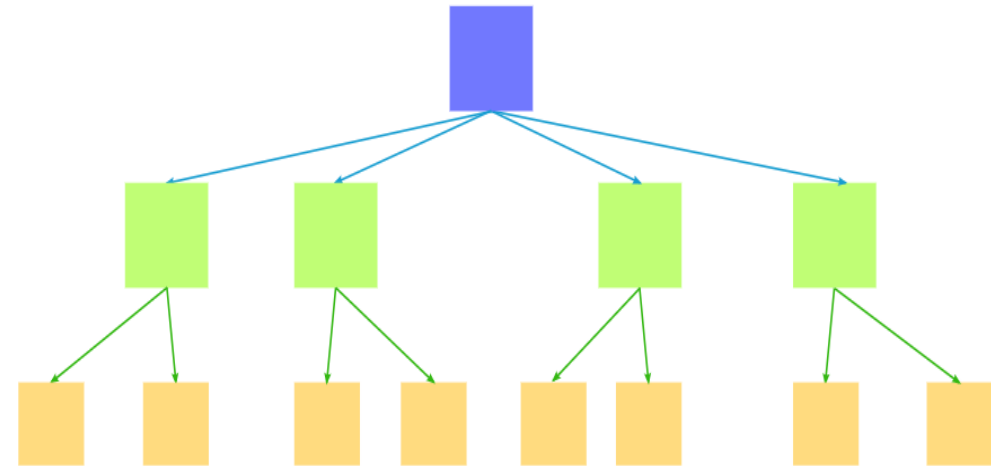
- The physical data model describes data need for a single project or application though it maybe integrated with other physical data models based on project scope.
- Data Model contains relationships between tables that which addresses cardinality and nullability of the relationships.
- Developed for a specific version of a DBMS, location, data storage or technology to be used in the project.
- Columns should have exact datatypes, lengths assigned and default values.
- Primary and Foreign keys, views, indexes, access profiles, and authorizations, etc. are defined.

CONVENTIONAL DATA MODELS

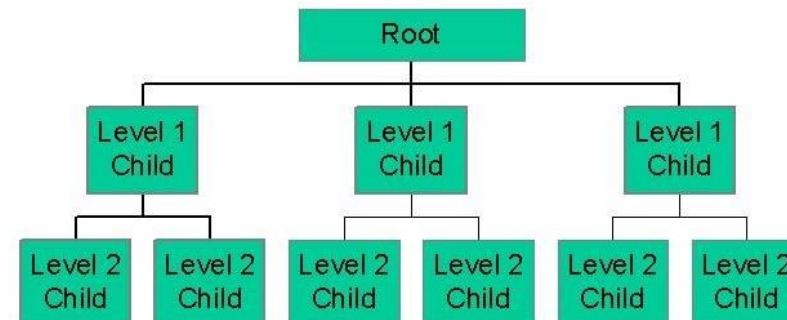
1. Hierarchical model
2. Relational model
3. Entity-relationship model
4. Network model
5. Object-oriented model

HIERARCHICAL MODEL

1. A **hierarchical database model** is a data model in which the data are organized into a tree-like structure.
2. The data are stored as **records** which are connected to one another through **links**.
3. A record is a collection of fields, with each field containing only one value. The **type** of a record defines which fields the record contains.
4. The hierarchical database model mandates that each child record has only one parent, whereas each parent record can have one or more child records.



Hierarchical database model



RELATIONAL MODEL

1. **RELATIONAL MODEL (RM)** represents the database as a collection of relations.
2. A relation is nothing but a table of values.
3. Every row in the table represents a collection of related data values.
4. These rows in the table denote a real-world entity or relationship.

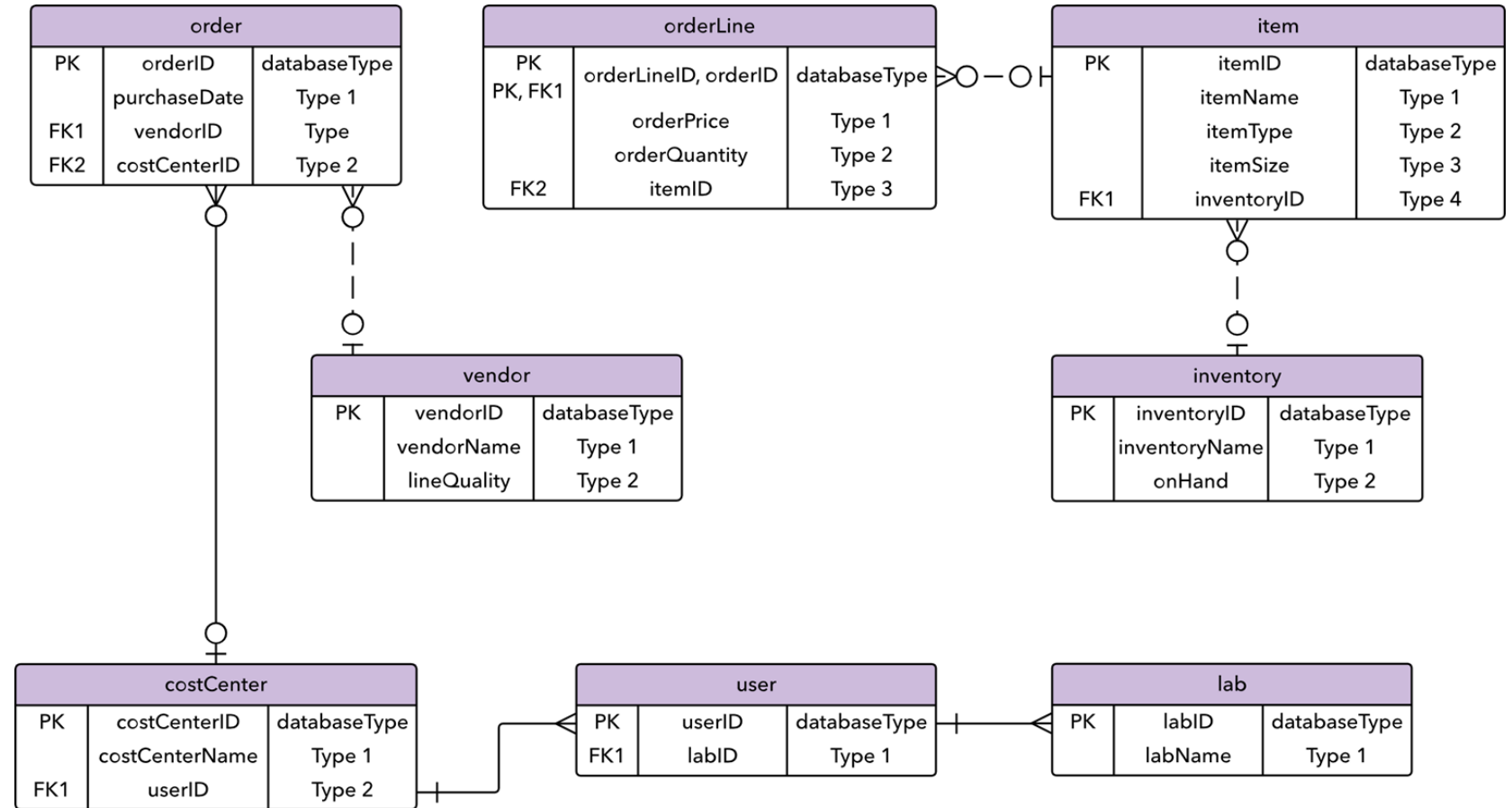
ID	First Name	Last Name
581-8463	Yan	Smith
962-6743	Marie	Johnston
826-272	Geoff	Lutter

Plan ID	Plan Provider
98374578	Provider A
82638367	Provider B
19274021	Provider C

ID	Plan ID	Type	Date
581-8463	98374578	R-5	12/04/2019
962-6743	82638367	M-9	09/08/2019
826-272	19274021	L-4	11/10/2019

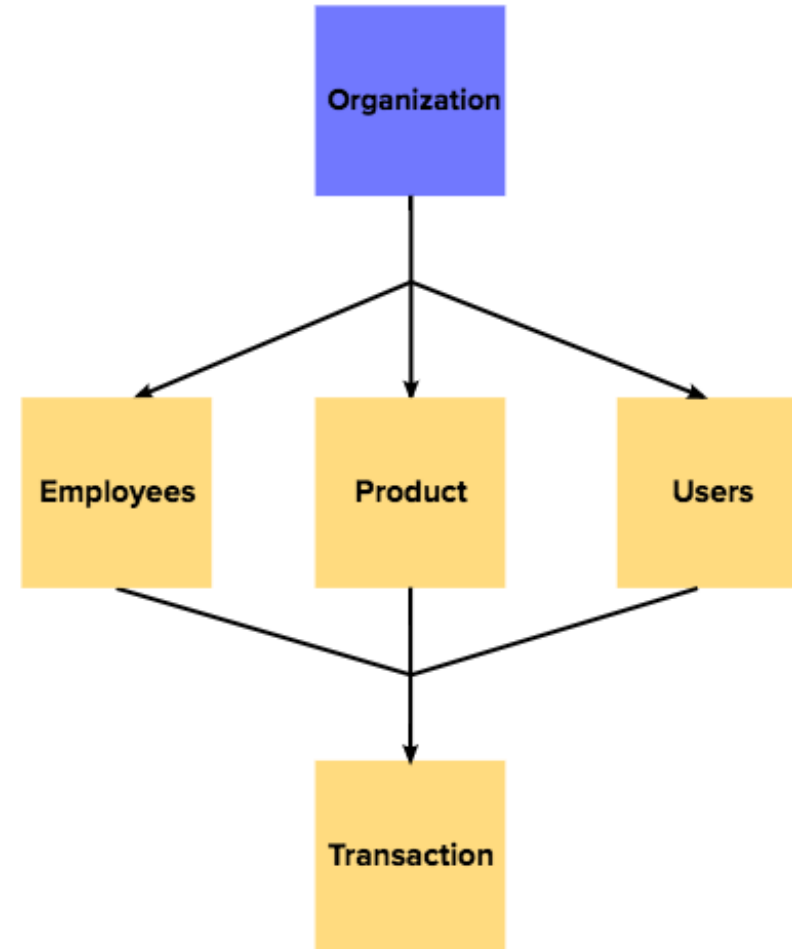
ENTITY-RELATIONSHIP MODEL

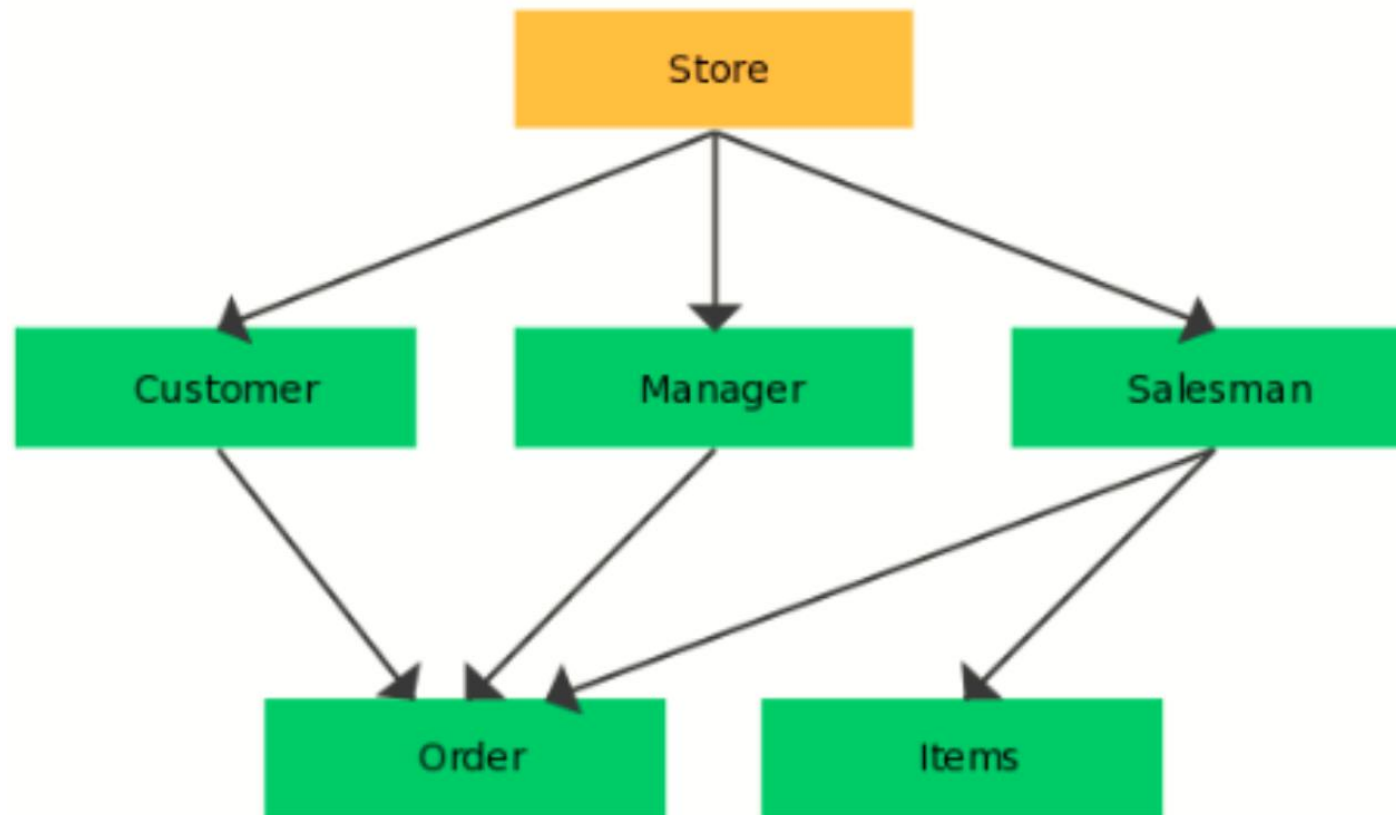
1. **ER model** stands for an **Entity-Relationship model**.
2. It is a high-level data **model**.
3. This **model** is used to define the data elements and **relationship** for a specified system.
4. It develops a conceptual design for the database.
5. It also develops a very simple and easy to design view of data.



NETWORK MODEL

1. The network model was created to solve the shortcomings of the hierarchical database model.
2. In this type of model, a child can be linked to multiple parents, a feature that was not supported by the hierarchical data model.
3. The parent nodes are known as owners and the child nodes are called members.





OBJECT-ORIENTED MODEL

Object oriented data model is based upon real world situations.

These situations are represented as objects, with different attributes.

All these object have multiple relationships between them.

Object-Oriented Model

Object 1: Maintenance Report

Date	
Activity Code	
Route No.	
Daily Production	
Equipment Hours	
Labor Hours	

Object 1 Instance

01-12-01
24
I-95
2.5
6.0
6.0

Object 2: Maintenance Activity

Activity Code	
Activity Name	
Production Unit	
Average Daily Production Rate	



IMPORTANCE OF DATA MODELING

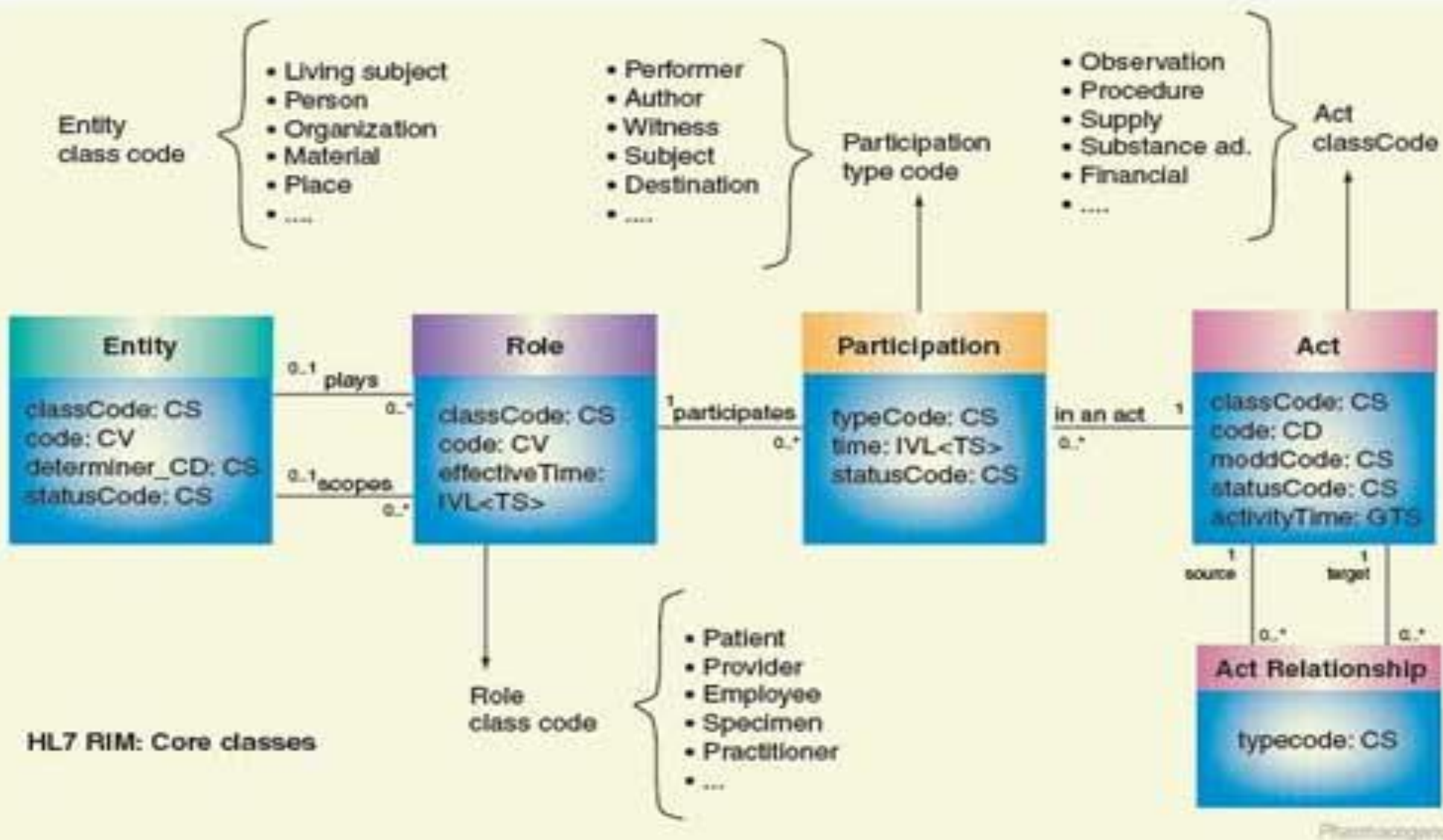
- A clear representation of data makes it easier to analyze the data properly. It provides a quick overview of the data which can then be used by the developers in varied applications.
- Data modeling represents the data properly in a model. It rules out any chances of data redundancy and omission. This helps in clear analysis and processing.
- Data modeling improves data quality and enables the concerned stakeholders to make data-driven decisions.

GENERIC DATA MODELING

A generic data model may define relation types such as a 'classification relation', being a **binary relation** between an individual thing and a kind of thing (a class) and a 'part-whole relation', being a binary relation between two things, one with the role of part, the other with the role of whole, regardless the kind of things that are related.

Conventional data models, on the other hand, have a fixed and limited domain scope, because the instantiation (usage) of such a model only allows expressions of kinds of facts that are predefined in the model.

Figure 1. The backbone of the HL7 Reference Information Model.



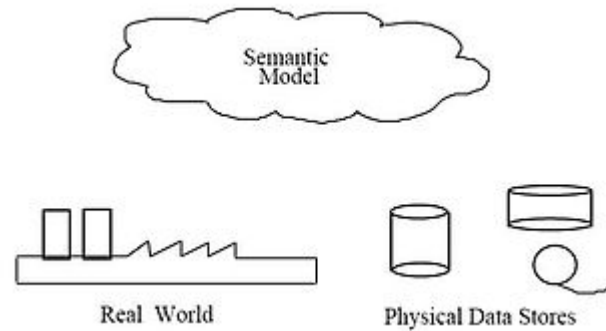
An entity plays a role that participates in an action. The RIM backbone is a generic set of classes from which health specific classes are derived, for example, observation, procedure and substance administration are all subclasses of act.

CD: Concept descriptor; CS: Coded simple value; CV: Coded value; GTS: General Timing Specification; HL7: Health Level Seven; IVL: Interval; RIM: Reference Information Model; TS: Point in time.

SEMANTIC DATA MODELING

To define the meaning of data within the context of its interrelationships with other data. As illustrated in the figure the real world, in terms of resources, ideas, events, etc., are symbolically defined within physical data stores. A semantic data model is an **abstraction** which defines how the stored symbols relate to the real world. Thus, the model must be a true representation of the real world.

SEMANTIC DATA MODELING



A semantic data model can be used to serve many purposes, such as:

- planning of data resources
- building of shareable databases
- evaluation of vendor software
- integration of existing databases

The overall goal of semantic data models is to capture more meaning of data by integrating relational concepts with more powerful **abstraction** concepts known from the **Artificial Intelligence** field. The idea is to provide high level modeling primitives as integral part of a data model in order to facilitate the representation of real world situations

KEY BUSINESS DECISIONS

Have a clear understanding of your organization's requirements and organize your data properly.

Keep your data models simple. The best data modeling practice here is to use a tool which can start small and scale up as needed.

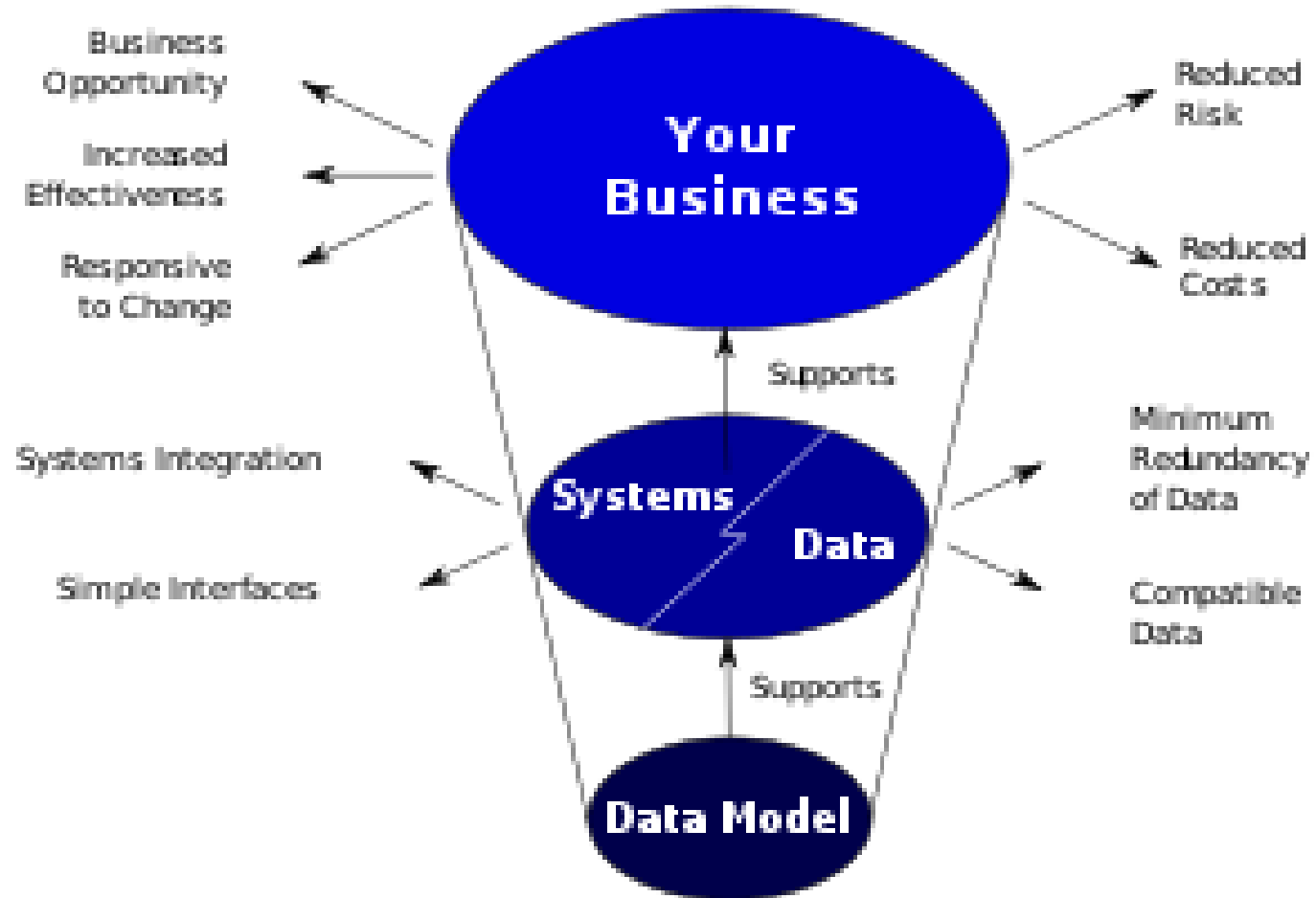
It is highly recommended to organize your data properly using individual tables for facts and dimensions to enable quick analysis.

Have a clear opinion on how much datasets you want to keep. Maintaining more than what is actually required wastes your data modeling, and leads to performance issues.

It is the best practice to maintain one-to-one or one-to-many relationships. The many-to-many relationship only introduces complexity in the system.

Data models become outdated quicker than you expect. It is necessary that you keep them updated from time to time.

APPLICATION OF MODELING IN BUSINESS



DEALING WITH MISSING DATA



MISSING DATA

- Missing data is a common problem and challenge for analysts.
- There are many reasons why data could be missing, including:



Respondents forgot to answer questions.

Respondents refused to answer certain questions.

Respondents failed to complete the survey.



A sensor failed.

Someone purposefully turned off recording equipment.

There was a power cut.

The method of data capture was changed.



An internet connection was lost.

A network went down.

A hard drive became corrupt.

A data transfer was cut short.

MISSING DATA

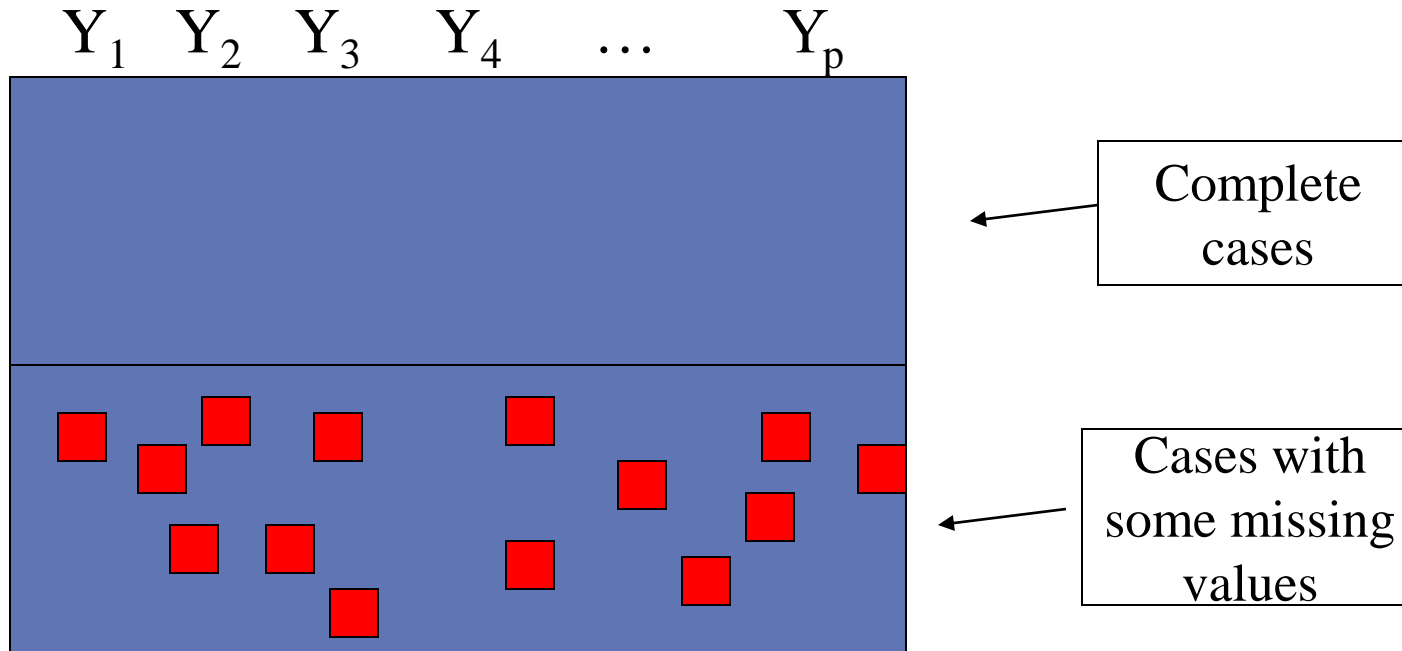
Missing data can usually be classified into:



- **Missing Completely at Random (MCAR):**
 - If missingness doesn't depend on the values of the data set.
 - **e.g.** a random sample of patients who had their blood pressure measured also had their weight measured.
- **Missing at Random (MAR):**
 - If missingness does not depend on the unobserved values of the data set but does depend on the observed.
 - **e.g.** patients with high blood pressure had their weight measured.
- **Not Missing at Random (NMAR):**
 - If missingness depends on the unobserved values of the data set.
 - **e.g.** overweight patients had their weight measured.

MORE GENERAL PROBLEM

Variables in
The data set



D_{obs} = Observed data: 

D_{miss} = Missing data: 

WHAT IS THE REASONS FOR MISSING DATA? (MISSING DATA MECHANISM)

X

Y_{obs}

Missing Completely at random (MCAR)
distribution
 $Y_{obs} = ?$

X

?

Missing at random (MAR)
distribution
 $Y_{obs} | X = x = ? | X = x$

Not Missing At random (NMAR)
distribution
 $Y_{obs} | X = x \neq ? | X = x$

MISSING DATA

Another example: **Survey data on drug use.**

- **Missing Completely at Random (MCAR):**
 - You removed 10% of the respondents data randomly.
- **Missing at Random (MAR):** (most common type)
 - People who come from poorer families might be less inclined to answer questions about drug use, and so the level of drug use is related to family income.
- **Not Missing at Random (NMAR):**
 - Students skipped the question on drug use because they feared that they would be expelled from school.

ANALYSIS

Most complete-case (available case) analyses are valid under MCAR assumption

- Default in most software packages
- Unreasonable assumption

MAR assumption is much weaker

- Depends on how good are the X as predictors of Y
- Non-testable assumption

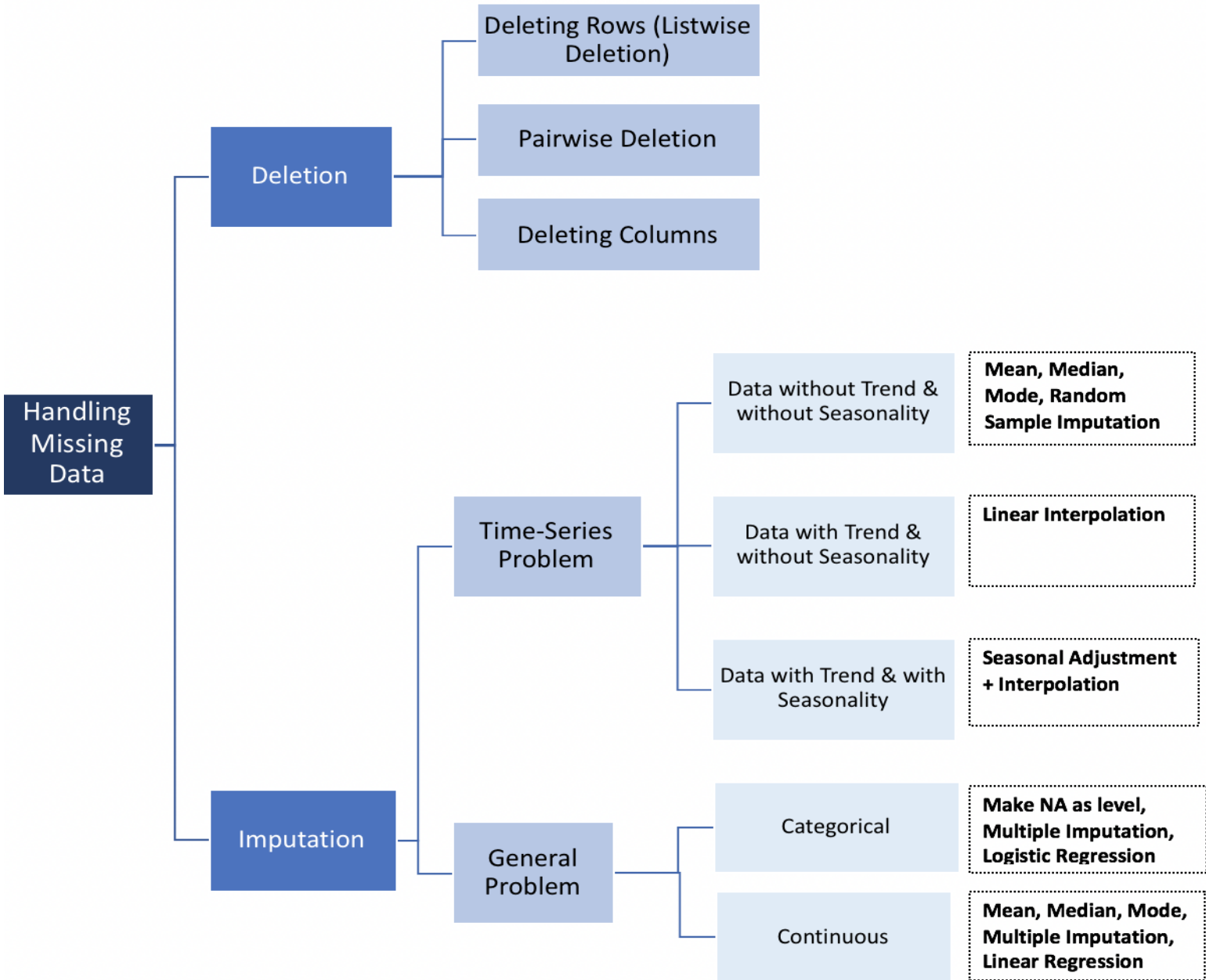
NMAR

- Need explicit formulation of differences between respondents and non-respondents
- Need External data
- Non-testable assumption

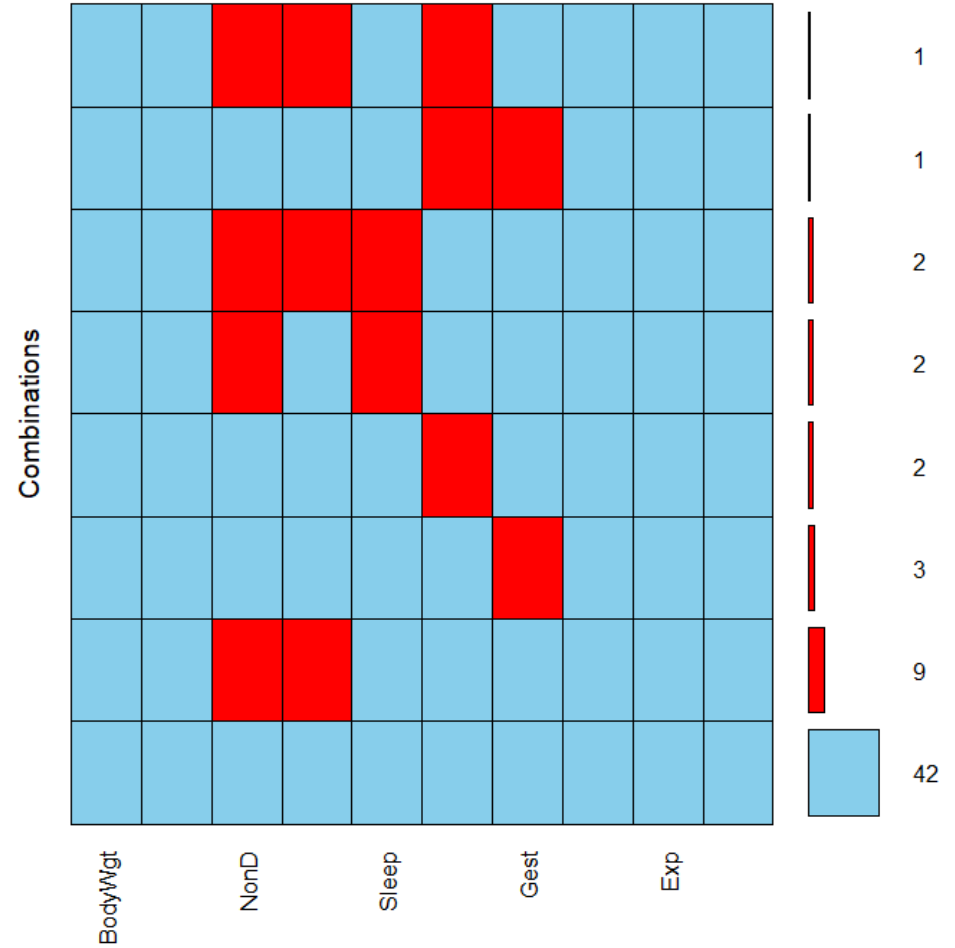
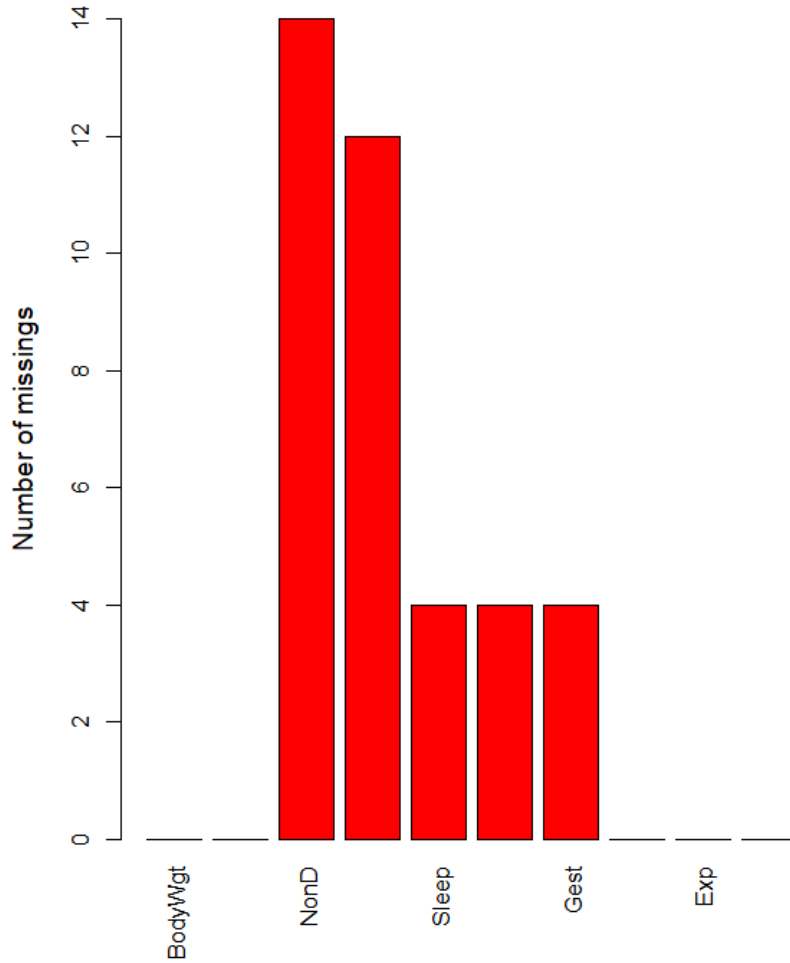
MISSING DATA

- Generally the procedure for dealing with missing data is:
 1. Identify the missing data.
 2. Identify the cause of the missing data.
 3. Either:
 - A. Remove the rows containing the missing data
 - Also called the **naïve approach**.
 - Make sure missing data isn't biased!
 - B. Replace missing values with alternative values.
 - **Impute** the missing values.
 - There are a number of approaches.

Deciding between A and B depends on which outcome you think will produce the **most reliable and accurate results**.



IDENTIFY MISSING DATA



IDENTIFY MISSING DATA

- Normally missing data is identified using summary in R:

```
> summary(sleep)
```

Bodywgt	Brainwgt	NonD	Dream	Sleep	Span
Min. : 0.005	Min. : 0.14	Min. : 2.100	Min. :0.000	Min. : 2.60	Min. : 2.000
1st Qu.: 0.600	1st Qu.: 4.25	1st Qu.: 6.250	1st Qu.:0.900	1st Qu.: 8.05	1st Qu.: 6.625
Median : 3.342	Median : 17.25	Median : 8.350	Median :1.800	Median :10.45	Median : 15.100
Mean : 198.790	Mean : 283.13	Mean : 8.673	Mean :1.972	Mean :10.53	Mean : 19.878
3rd Qu.: 48.203	3rd Qu.: 166.00	3rd Qu.:11.000	3rd Qu.:2.550	3rd Qu.:13.20	3rd Qu.: 27.750
Max. :6654.000	Max. :5712.00	Max. :17.900	Max. :6.600	Max. :19.90	Max. :100.000
		NA's :14	NA's :12	NA's :4	NA's :4

Gest	Pred	Exp	Danger
Min. : 12.00	Min. :1.000	Min. :1.000	Min. :1.000
1st Qu.: 35.75	1st Qu.:2.000	1st Qu.:1.000	1st Qu.:1.000
Median : 79.00	Median :3.000	Median :2.000	Median :2.000
Mean :142.35	Mean :2.871	Mean :2.419	Mean :2.613
3rd Qu.:207.50	3rd Qu.:4.000	3rd Qu.:4.000	3rd Qu.:4.000
Max. :645.00	Max. :5.000	Max. :5.000	Max. :5.000
NA's :4			

- There are also a number of different ways to visualize missing data in R..

Gender	Manpower	Sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
.	26	259
M	32	297

Gender	Manpower	Sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
.	26	259
M	32	297

	Listwise deletion (complete case analysis)	Pairwise deletion
Description:	Analyse the data rows where there is complete data for every column.	Analyse the data rows where the variables of interest have data present.
Advantages:	<ul style="list-style-type: none"> • Simple • Easily compare across analyses. 	<ul style="list-style-type: none"> • Uses all possible information.
Limitations:	<ul style="list-style-type: none"> • Could be biased (if the data is not MCAR). • Lower n, reduces statistical power. 	<ul style="list-style-type: none"> • Separate analyses cannot be compared as the data / sample will be different.

REMOVE MISSING DATA ROWS

- In R missing values can be represented by:

NA Not Available (placeholder for a missing value).

NULL Empty value.

Inf Infinity.

- It is possible to use `is.na()`, `is.null()` and `is.infinite()` functions in R to identify missing, empty and infinite values in datasets.
- The function `complete.cases()` can be used to identify the data rows in a matrix or data frame that are / aren't complete.
 - Only NA and NULL are regarded as missing, Inf is treated as valid.

EXAMPLE MISSING DATASET

- We will be using the following sleep dataset as an example.
- It contains the following data on 62 species of mammals:

Column	Description
Dream	Length of dreaming sleep
NonD	Non-dreaming sleep
Sleep	Sum of Dream and NonD
BodyWgt	Body weight (kg)
BrainWgt	Brain weight (g)
Span	Life span (yrs)
Gest	Gestation time in days
Pred	Degree to which species were preyed upon (1-low to 5-high scale)
Exp	Degree of their exposure while sleeping (1-low to 5-high scale)
Danger	Overall danger (1-low to 5-high scale)

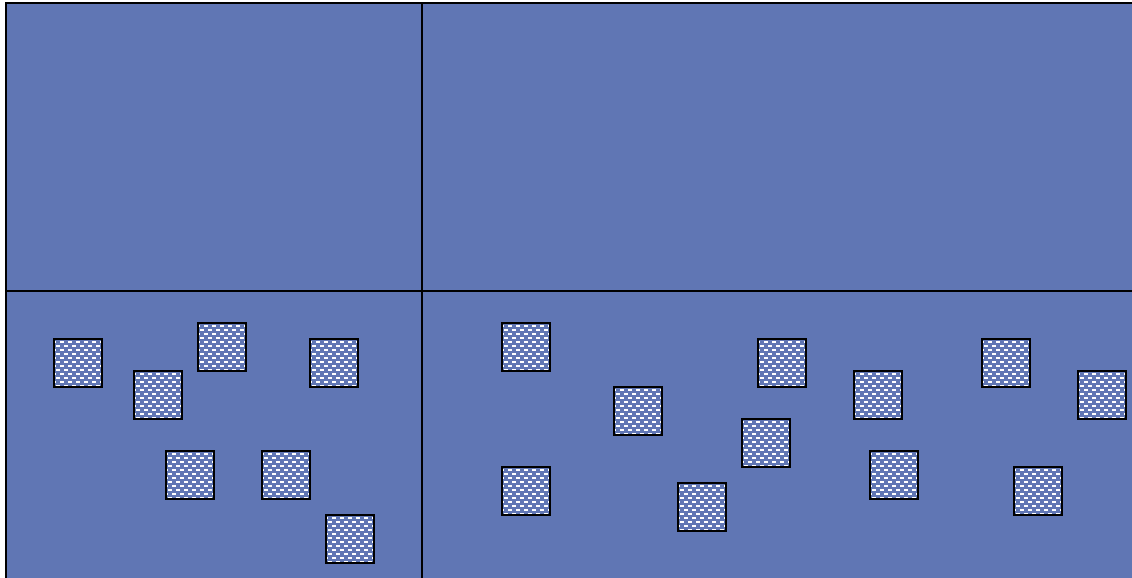
- Various data is missing in the dataset (NA values).

REPLACING MISSING DATA

- The two most common methods for replacing missing data are:

	Simple Imputation	Multiple Imputation
Description:	Missing values are replaced with the mean, median or mode value.	Estimates missing data through repeated simulations.
Stochastic:	No	Yes
Advantages:	<ul style="list-style-type: none">Simple.	<ul style="list-style-type: none">Variability more accurate.
Limitations:	<ul style="list-style-type: none">Could be biased (if the data is not MCAR).Underestimates standard errors.Could distort correlations among variables.	<ul style="list-style-type: none">Algorithms are more complex.Normally would require complex coding (R library available).

IMPUTATION



Imputation refers to filling in a value for each missing datum based on other information (e.g., a model and observed data)

Imputation :

Draws from predictive distribution $\Pr(D_{miss} | D_{obs})$

SIMPLE IMPUTATION

- Simply replace the missing values with the mean, median or mode:
 - `mean(x)`
 - `median(x)`
 - `names(sort(-table(x)))[1]`
- For example, using `mean(sleep$x, na.rm=TRUE)`:

```
> head(sleep)
```

	BodyWgt	BrainWgt	NonD	Dream	Sleep	Span	Gest	Pred	Exp	Danger
1	54.000	512.0	NA	NA	3.2	38.6	645	3	5	3
2	1.580	6.6	6.3	2.0	3.3	4.5	42	3	1	3
3	3.385	14.9	NA	NA	22.5	14.0	60	1	1	1
4	0.920	5.7	NA	NA	16.5	NA	25	5	2	3
5	2547.000	4603.0	2.1	1.8	3.9	69.0	624	3	5	4
6	10.550	179.5	9.1	0.7	9.8	27.0	180	4	4	4

8.672917 (Mean of NonD) →

1.972 (Mean of Dream) →

19.87759 (Mean of Span) →

- Replace NA values with `sleep$x[is.na(sleep$x)] <- value`
- Note:** If the value is **NA** the `is.na()` function return the value of true, otherwise, return to a value of false.

IMPUTATION

Typically used for item nonresponse

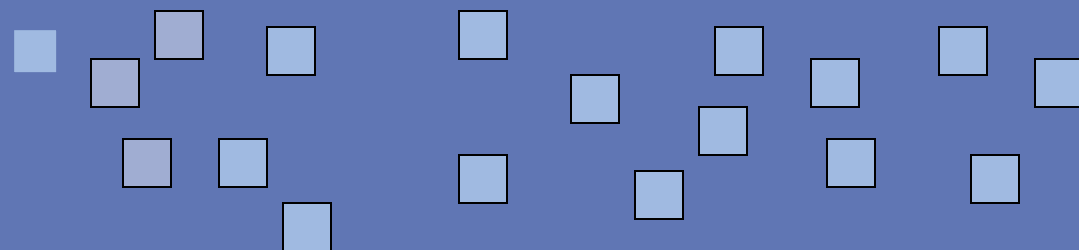
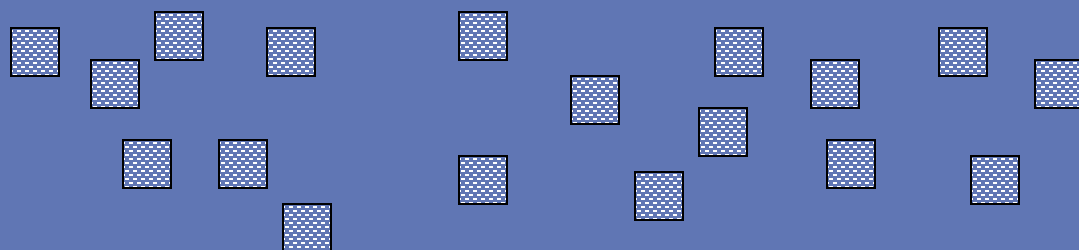
Benefits of imputation

- **Completes the data matrix**
- **If imputation is performed by a producer of public-use data:**
 - **Missing data are handled comparably across secondary data analyses**
 - **Information available to the data producer but not the public can be used in creating imputations**

MULTIPLE IMPUTATION

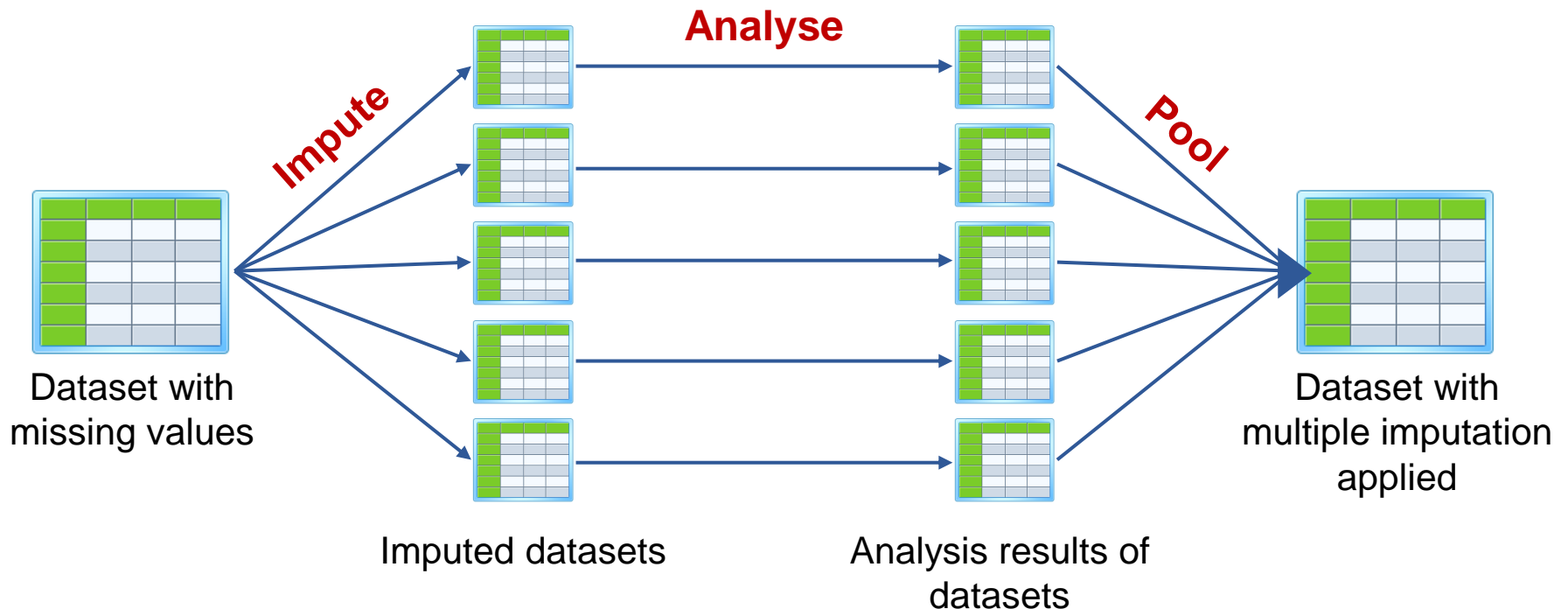
Repeat Imputation process several times (say M times)

Uncertainty due to imputation is captured by the “between Imputed Data” Variation



MULTIPLE IMPUTATION

- The idea of Multiple Imputation is to replace each missing value with multiple acceptable values that represent a distribution of possibilities.
- This results in a number of complete datasets (usually 3-10):



MULTIPLE IMPUTATION

The general procedure for the **chained equation approach** to multiple imputation (used in `mice()`) is:

1. A simple imputation is performed for every missing value.
2. One of the missing variables are set back to missing.
3. Regression is performed (linear, logistic, polynomial etc.), the missing variable being the forecast variable and all other variables in the dataset being the predictor variables.
4. Missing values are replaced with predictions (imputations) from the regression.
5. Repeat steps 2-4 for each variable that has missing data (one cycle).
6. Repeat for a number of cycles then retain results as one imputed dataset.

Mice: Multivariate Imputation By Chained Equations

MICE (Multivariate Imputation via Chained Equations) is one of the commonly used package by R users.

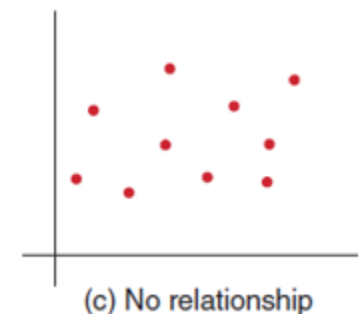
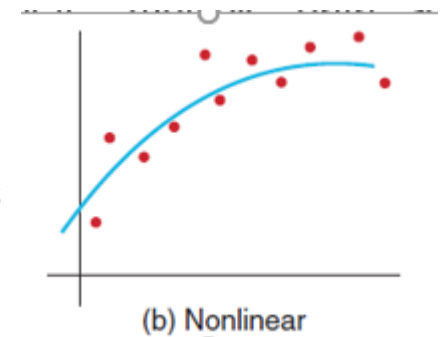
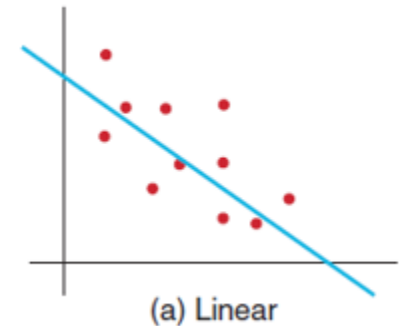
Creating multiple imputations as compared to a single imputation (such as mean) takes care of uncertainty in missing values.

MICE assumes that the missing data are Missing at Random (MAR), which means that the probability that a value is missing depends only on observed value and can be predicted using them.

It imputes data on a variable by variable basis by specifying an imputation model per variable.

- **For example:**

1. Suppose we have X_1, X_2, \dots, X_k variables.
2. If X_1 has missing values, then it will be regressed on other variables X_2 to X_k .
3. The missing values in X_1 will be then replaced by predictive values obtained. Similarly, if X_2 has missing values, then X_1, X_3 to X_k variables will be used in prediction model as independent variables.
4. Later, missing values will be replaced with predicted values.
5. By default, linear regression is used to predict continuous missing values. Logistic regression is used for categorical missing values.
6. Once this cycle is complete, multiple data sets are generated.
7. These data sets differ only in imputed missing values.
8. Generally, it's considered to be a good practice to build models on these data sets separately and combining their results.



Reference:- <https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/>

ANALYSIS OF MULTIPLY IMPUTED DATA

Analyze each imputed data separately

$$\textit{Estimate} : e_1, e_2, \dots, e_M$$

$$\textit{Variance}(= SE^2) : v_1, v_2, \dots, v_M$$

Combine Estimates

$$\bar{e} = (e_1 + e_2 + \dots + e_M) / M$$

Combine variances

$$\left. \begin{aligned} \bar{v} &= (v_1 + v_2 + \dots + v_M) / M \\ b &= \text{var}(e_1, e_2, \dots, e_M) \end{aligned} \right\} T = \bar{v} + (1 + 1/M)b$$

SOFTWARE FOR CREATING IMPUTATIONS

SAS

- PROC MI
- User-developed IVEWARE (www.isr.umich.edu/src/smp/ive)

Stata

- ICE

R

- MICE
- MI

Another good source:
www.multiple-imputation.com

SOLAS

AMELIA

SPSS

Stand-Alone

- SRCWARE (www.isr.umich.edu/src/smp/ive)
- NORM
- PAN (www.stat.psu.edu/~jls)
- CAT

MULTIPLE IMPUTATION

- We will focus on using the *mice* package.
- The *mice* package has many built in imputation techniques including:

Method	Description	Scale type
pmm	Predictive mean matching	numeric
norm	Bayesian linear regression	numeric
norm.nob	Linear regression, non-Bayesian	numeric
mean	Unconditional mean imputation	numeric
2l.norm	Two-level linear model	numeric
logreg	Logistic regression	factor, 2 levels
polyreg	Polytomous (unordered) regression	factor, >2 levels
lda	Linear discriminant analysis	factor
sample	Random sample from the observed data	any

- Example: `mice(data, meth=c('sample','pmm','logreg','norm'))`

Data Analytics:

Unit 3

Regression

- A statistical measure that attempts to determine the strength of the relationship between one dependent variable (usually denoted by Y) and a series of other changing variables (known as independent variables).
- Forecast value of a dependent variable (Y) from the value of independent variables (X_1, X_2, \dots).

Regression Analysis

- In statistics, regression analysis includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables
- Regression analysis is widely used for prediction and forecasting,

Dependent & independent variable

- **Independent variables are regarded as inputs to a system and may take on different values freely.**
- **Dependent variables are those values that change as a consequence of changes in other values in the system.**
- **Independent variable is also called as predictor or explanatory variable and it is denoted by X.**
- **Dependent variable is also called as response variable and it is denoted by Y.**

Linear regression

- The simplest mathematical relationship between two variables x and y is a linear relationship.
- In a cause and effect relationship, the independent variable is the cause, and the dependent variable is the effect.
- Least squares linear regression is a method for predicting the value of a dependent variable Y , based on the value of an independent variable X .

The first order linear model

$$Y = b_0 + b_1 X + \epsilon$$

Y = dependent variable

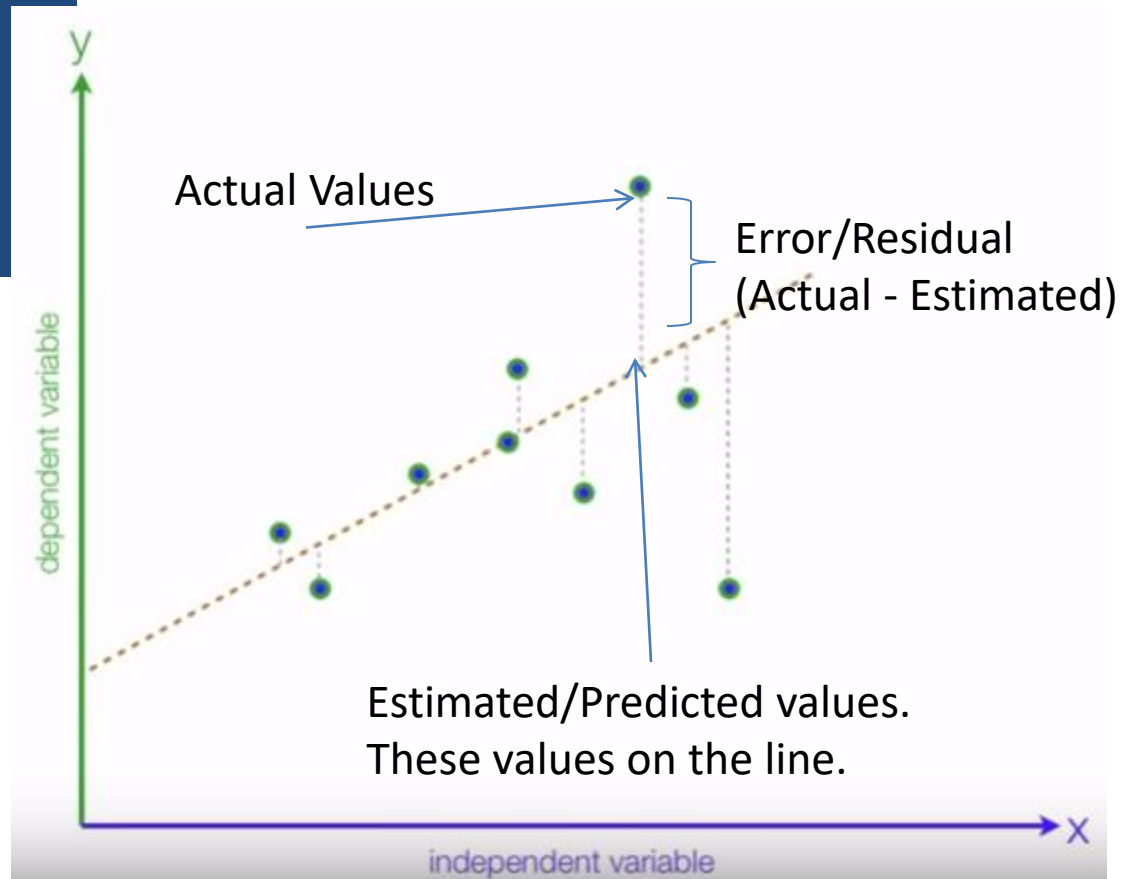
X = independent variable

b_0 = Y-intercept

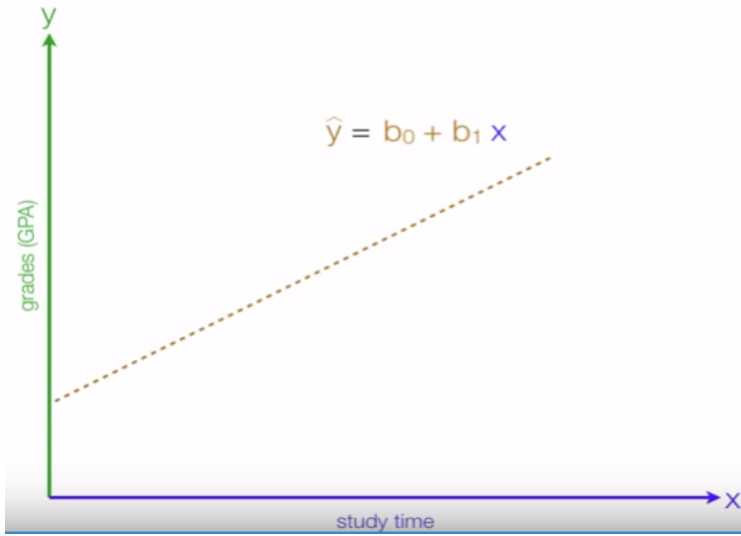
b_1 = slope of the line

ϵ = error variable

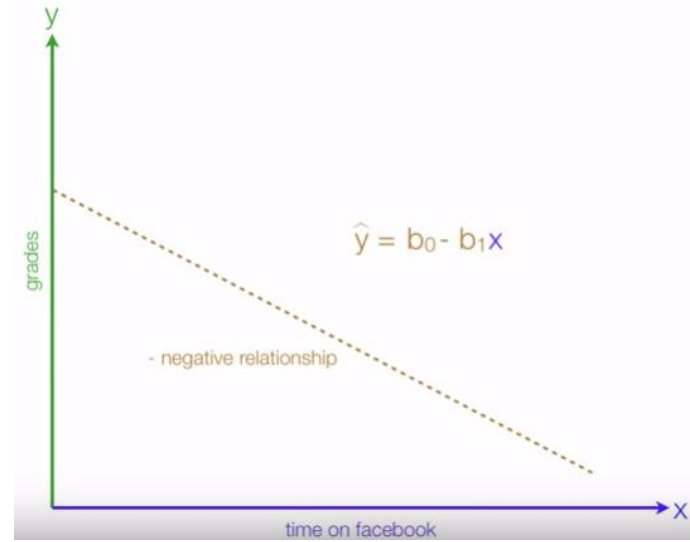
Note: Very important.
Linear regression Equation,
The graph below.



Positive Regression



Negative Regression

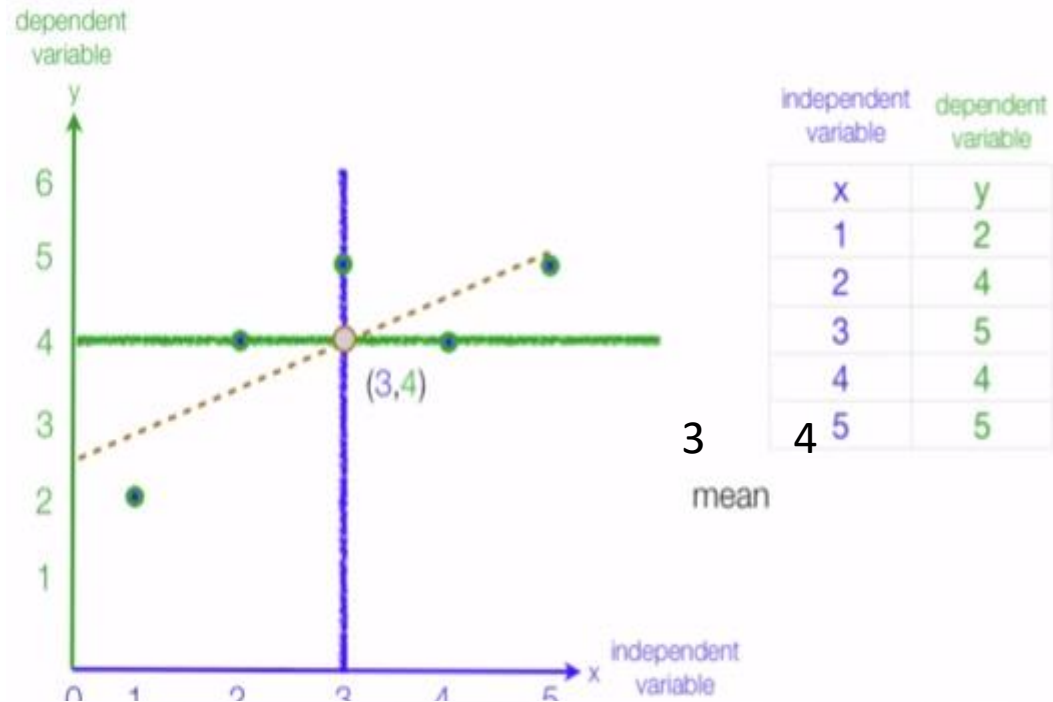


Building/Constructing Linear Regression using Ordinary least Square method

Step 1: Calculate the mean of independent(X) and dependent variable (Y).

Mean is (\bar{X}, \bar{Y})

See the example



Step 2: Calculate the distance of all Points from mean

$x - \bar{x}$	$y - \bar{y}$
---------------	---------------

x	y	$x - \bar{x}$	$y - \bar{y}$
1	2	-2	-2
2	4	-1	0
3	5	0	1
4	4	1	0
5	5	2	1

mean 3 4

Step 3: Calculate the

$$(x - \bar{x})^2$$

And

$$(x - \bar{x})(y - \bar{y})$$

x	y	$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})^2$	$(x - \bar{x})(y - \bar{y})$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2
Mean	3	4		10	6

Step 4: find the slope value using the equation

$$\hat{y} = b_0 + b_1 x$$

$$b_1 = \frac{6}{10} = .6 = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

Step 5: Using the slope value calculate the intercept b_0 value.

$$4 = b_0 + .6(3)$$

$$b = \frac{\sum y - n \bar{y}}{N}$$

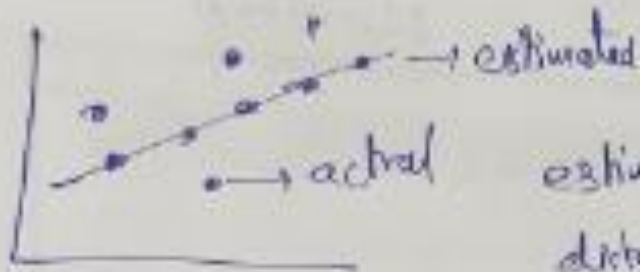
Step 6: Arrange the values to get the linear Regression equation

$$\hat{y} = 2.2 + .6x$$

$$Y = 2.2 + 0.6 * X$$

Standard Error Estimate

Standard Error of the estimate : (Mean Squared Error)



estimated distance b/w actual & estimated, & distance is Error.

$$\text{Stand Err} = \sqrt{\frac{\sum (\hat{Y} - Y)^2}{n-2}}$$

↑
no. of observations

$$= 0.89 = \sqrt{\frac{2.4}{5-2}}$$

X	Y	\hat{Y}	$\hat{Y} - Y$	$(\hat{Y} - Y)^2$
1	2	2.8	.8	.64
2	4	3.4	-.6	.36
3	5	4	-1	1
4	4	4.6	-.6	0.36
5	5	5.2	-.2	0.04
				2.4

Model fit Statistics/Goodness of fit for Linear Regression:

This will tell us how good the Regression Line is

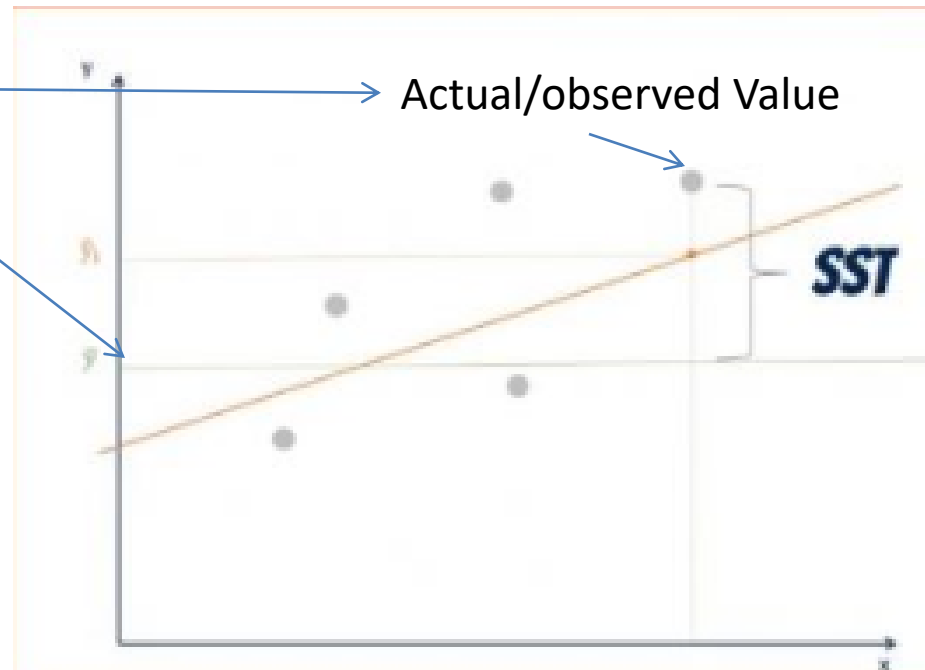
To find out we need 4 things

1.SST: The **sum of squares total**, denoted **SST**, is the squared differences between the observed *dependent variable* and its **mean**. You can think of this as the dispersion of the observed variables around the **mean** – **much like the variance in descriptive statistics**.

$$\sum_{i=1}^n (y_i - \bar{y})^2$$

Mean

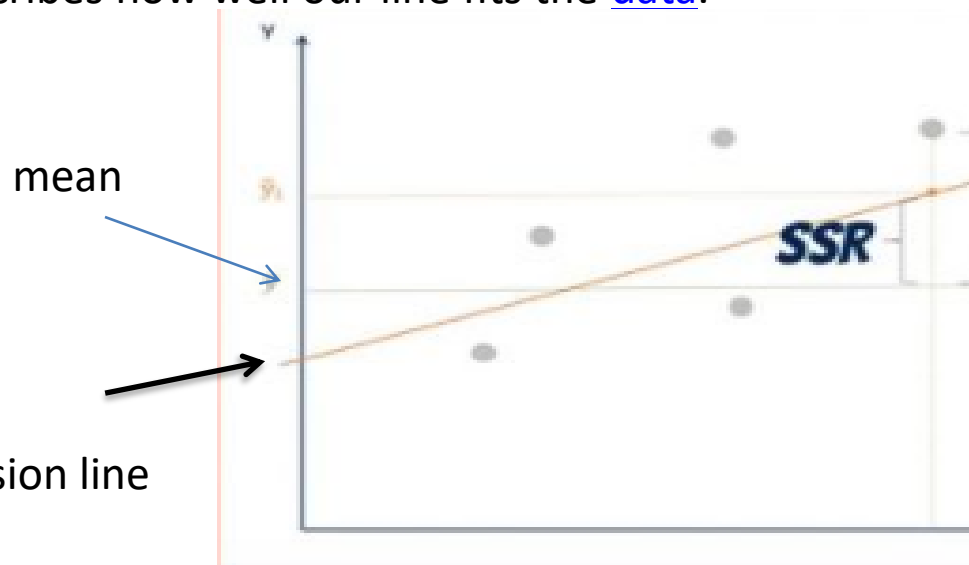
x	y	$y - \bar{y}$	$(y - \bar{y})^2$
1	2	-2	4
2	4	0	0
3	5	1	1
4	4	0	0
5	5	1	1
mean	4		6



2. SSR: The second term is the **sum of squares due to regression**, or **SSR**. It is the sum of the differences between the *predicted* value and the **mean** of the *dependent variable*. Think of it as a measure that describes how well our line fits the [data](#).

$$\sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

Predicted values,
Nothing but Regression line



If this value of **SSR** is equal to the **sum of squares total**, it means our **regression model** captures all the observed variability and is perfect. Once again, we have to mention that another common notation is **ESS** or **explained sum of squares**.

x	y	$y - \bar{y}$	$(y - \bar{y})^2$	\hat{y}	$\hat{y} - \bar{y}$	$(\hat{y} - \bar{y})^2$
1	2	-2	4	2.8	-1.2	1.44
2	4	0	0	3.4	-.6	.36
3	5	1	1	4	0	0
4	4	0	0	4.6	.6	.36
5	5	1	1	5.2	1.2	1.44
mean	4		6			3.6

3. SSE: The last term is the **sum of squares error**, or **SSE**. The error is the difference between the *observed* value and the *predicted* value.

$$SSE = \sum (y_i - \hat{y}_i)^2$$

We can say that

$$SST = SSR + SSE$$

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2$$



4. Calculate R Square

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$SST = \sum (y_i - \bar{y})^2$$

$$SSR = \sum (\hat{y}_i - \bar{y})^2$$

$$SSE = \sum (y_i - \hat{y}_i)^2$$

$$SST = SSR + SSE$$

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

Solved Examples

Question: Find linear regression equation for the following two sets of data:

x	2	4	6	8
y	3	7	5	10

Solution:

Construct the following table:

x	y	x^2	xy
2	3	4	6
4	7	16	28
6	5	36	30
8	10	64	80
$\sum x$ = 20	$\sum y$ = 25	$\sum x^2$ = 120	$\sum xy$ = 144

$$b = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$$

$$b = \frac{4 \times 144 - 20 \times 25}{4 \times 120 - 400}$$

$$b = 0.95$$

$$a = \frac{\sum y \sum x^2 - \sum x \sum xy}{n(\sum x^2) - (\sum x)^2}$$

$$a = \frac{25 \times 120 - 20 \times 144}{4(120) - 400}$$

$$a = 1.5$$

Linear regression is given by:

$$y = a + bx$$

$$y = 1.5 + 0.95x$$

What does R Square tells

What Is R-squared?

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

How To Interpret R-squared in Regression Analysis

R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scale.

The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

$$\text{R-squared} = \frac{\text{Explained variation}}{\text{Total variation}}$$

R-squared is always between 0 and 100%:

0% indicates that the model explains none of the variability of the response data around its mean.

100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

Some Problems with R-squared

Problem 1: Every time you add a predictor to a model, the R-squared increases, even if due to chance alone. It never decreases. Consequently, a model with more terms may appear to have a better fit simply because it has more terms.

Problem 2: If a model has too many predictors and higher order polynomials, it begins to model the random noise in the data. This condition is known as [overfitting the model](#) and it produces misleadingly high R-squared values and a lessened ability to make predictions.

What Is the Adjusted R-squared?

The adjusted R-squared compares the explanatory power of regression models that contain different numbers of predictors.

Suppose you compare a five-predictor model with a higher R-squared to a one-predictor model. Does the five predictor model have a higher R-squared because it's better? Or is the R-squared higher because it has more predictors? Simply compare the adjusted R-squared values to find out!

The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance. The adjusted R-squared can be negative, but it's usually not. It is always lower than the R-squared.

In the simplified Best Subsets Regression output below, you can see where the adjusted R-squared peaks, and then declines. Meanwhile, the R-squared continues to increase.

Ordinary Least Squares (OLS) is the most common estimation method for linear models—and that's true for a good reason. As long as your model satisfies the OLS assumptions for linear regression, you can rest easy knowing that you're getting the best possible estimates.

Regression is a powerful analysis that can analyze multiple variables simultaneously to answer complex research questions.

However, if you don't satisfy the OLS assumptions, you might not be able to trust the results.

What Does OLS Estimate and What are Good Estimates?

- Regression analysis is like other inferential methodologies.
- Our goal is to draw a random sample from a population and use it to estimate the properties of that population.
- In regression analysis, the coefficients in the regression equation are estimates of the actual population parameters. We want these coefficient estimates to be the best possible estimates!

Example to understand OLS

Suppose you request an estimate—say for the cost of a service that you are considering. How would you define a reasonable estimate?

The estimates should tend to be right on target.

They should not be systematically too high or too low. In other words, they should be unbiased or correct on average.

Recognizing that estimates are almost never exactly correct, you want to minimize the discrepancy between the estimated value and actual value. Large differences are bad!

These two properties are exactly what we need for our coefficient estimates!

Remember:

When your linear regression model satisfies the OLS assumptions, the procedure generates unbiased coefficient estimates that tend to be relatively close to the true population values (minimum variance).

In fact, the Gauss-Markov theorem states that OLS **produces** estimates that are better than estimates from all other linear model estimation methods when the assumptions hold true.

The Seven Classical OLS Assumptions

Like many statistical analyses, [ordinary least squares](#) (OLS) regression has underlying assumptions. When these classical assumptions for linear regression are true, ordinary least squares produces the best estimates. However, if some of these assumptions are not true, you might need to employ remedial measures or use other estimation methods to improve the results.

Many of these assumptions describe properties of the error term. Unfortunately, the error term is a population value that we'll never know. Instead, we'll use the next best thing that is available—the [residuals](#). [Residuals](#) are the [sample](#) estimate of the error for each observation.

In fact, the defining characteristic of linear regression is this functional form of the *parameters* rather than the ability to model curvature. Linear models can model curvature by including nonlinear *variables* such as polynomials and transforming exponential functions. To satisfy this assumption, the correctly specified model must fit the linear pattern.

OLS Assumption 2: The error term has a population mean of zero

The error term accounts for the variation in the dependent variable that the independent variables do not explain. Random chance should determine the values of the error term. For your model to be unbiased, the average value of the error term must equal zero.

Suppose the average error is +7. This non-zero average error indicates that our model systematically underpredicts the observed values. [Statisticians](#) refer to systematic error like this as bias, and it signifies that our model is inadequate because it is not correct on average.

Stated another way, we want the expected value of the error to equal zero. If the expected value is +7 rather than zero, part of the error term is predictable, and we should add that information to the regression model itself. We want only random error left for the error term.

You don't need to worry about this assumption when you include the constant in your regression model because it forces the mean of the residuals to equal zero. For more information about this assumption, read my post about the [regression constant](#).

OLS Assumption 3: All independent variables are uncorrelated with the error term

If an independent variable is correlated with the error term, we can use the independent variable to predict the error term, which violates the notion that the error term represents unpredictable random error. We need to find a way to incorporate that information into the regression model itself.

This assumption is also referred to as exogeneity. When this type of [correlation](#) exists, there is endogeneity. Violations of this assumption can occur because there is simultaneity between the independent and dependent variables, omitted variable bias, or measurement error in the independent variables.

Violating this assumption biases the coefficient estimate. To understand why this bias occurs, keep in mind that the error term always explains some of the variability in the dependent variable. However, when an independent variable correlates with the error term, OLS incorrectly attributes some of the variance that the error term actually explains to the independent variable instead. For more information about violating this assumption, read my post about [confounding variables and omitted variable bias](#).

OLS Assumption 4: Observations of the error term are uncorrelated with each other

One observation of the error term should not predict the next observation. For instance, if the error for one observation is positive and that systematically increases the probability that the following error is positive, that is a positive correlation. If the subsequent error is more likely to have the opposite sign, that is a negative correlation. This problem is known both as serial correlation and autocorrelation. Serial correlation is most likely to occur in time series models.

For example, if sales are unexpectedly high on one day, then they are likely to be higher than average on the next day. This type of correlation isn't an unreasonable expectation for some subject areas, such as inflation rates, GDP, unemployment, and so on.

Assess this assumption by graphing the residuals in the order that the data were collected. You want to see randomness in the plot. In the graph for a sales model, there is a cyclical pattern with a positive correlation.

OLS Assumption 5: The error term has a constant variance (no heteroscedasticity)

The variance of the errors should be consistent for all observations. In other words, the variance does not change for each observation or for a range of observations. This preferred condition is known as homoscedasticity (same scatter). If the variance changes, we refer to that as heteroscedasticity (different scatter).

The easiest way to check this assumption is to create a residuals versus fitted value plot. On this type of graph, heteroscedasticity appears as a cone shape where the spread of the residuals increases in one direction. In the graph below, the spread of the residuals increases as the fitted value increases.

OLS Assumption 6: No independent variable is a perfect linear function of other explanatory variables

Perfect correlation occurs when two variables have a [Pearson's correlation coefficient](#) of +1 or -1. When one of the variables changes, the other variable also changes by a completely fixed proportion. The two variables move in unison.

Perfect correlation suggests that two variables are different forms of the same variable. For example, games won and games lost have a perfect negative correlation (-1). The temperature in Fahrenheit and Celsius have a perfect positive correlation (+1).

[Ordinary least squares](#) cannot distinguish one variable from the other when they are perfectly correlated. If you specify a model that contains independent variables with perfect correlation, your statistical software can't fit the model, and it will display an error message. You must remove one of the variables from the model to proceed.

Perfect correlation is a show stopper. However, your statistical software can fit OLS regression models with imperfect but strong relationships between the independent variables. If these correlations are high enough, they can cause problems. Statisticians refer to this condition as multicollinearity, and it reduces the precision of the estimates in OLS linear regression.

OLS Assumption 7: The error term is normally distributed (optional)

OLS does not require that the error term follows a [normal distribution](#) to produce unbiased estimates with the minimum variance. However, satisfying this assumption allows you to perform statistical hypothesis testing and generate reliable [confidence intervals](#) and [prediction intervals](#). The easiest way to determine whether the residuals follow a normal distribution is to assess a normal probability plot. If the residuals follow the straight line on this type of graph, they are normally distributed. They look good on the plot below!

Why You Should Care About the Classical OLS Assumptions

In a nutshell, your linear model should produce residuals that have a mean of zero, have a constant variance, and are not correlated with themselves or other variables.

If these assumptions hold true, the OLS procedure creates the best possible estimates. In statistics, [estimators](#) that produce unbiased estimates that have the smallest variance are referred to as being “efficient.” Efficiency is a statistical concept that compares the quality of the estimates calculated by different procedures while holding the sample size constant. OLS is the most efficient linear regression [estimator](#) when the assumptions hold true.

Property 4: Asymptotic Unbiasedness

This property of OLS says that as the sample size increases, the biasedness of OLS estimators disappears.

Property 5: Consistency

An estimator is said to be consistent if its value approaches the actual, true parameter (population) value as the sample size increases. An estimator is consistent if it satisfies two conditions:

- a. It is asymptotically unbiased
- b. Its variance converges to 0 as the sample size increases.

Both these hold true for OLS estimators and, hence, they are consistent estimators. For an estimator to be useful, consistency is the minimum basic requirement. Since there may be several such estimators, asymptotic efficiency also is considered. Asymptotic efficiency is the sufficient condition that makes OLS estimators the best estimators.

What is OLS regression?

- OLS, or ordinary least squares regression, is a method that statisticians use to approximate the unspecified parameters in a linear regression model.
- It's important to note that while OLS isn't a model itself, it's an estimator for the parameters of a linear regression model.
- Whenever a linear regression model accurately fulfills its assumptions, statisticians can observe coefficient estimates that are close to the actual population values.

Properties of OLS Estimator

- 1. Un-biasedness**
- 2. Minimum variance**
- 3. Efficiency**
- 4. Linear estimator**
- 5. BLUE (Best, linear, unbiased and efficient)**

PROPERTIES OF LEAST SQUARE ESTIMATES

1) Un-biasedness

- The estimator should ideally be an unbiased estimator of true parameter/population values.

Mathematically,

$$E(b_0) = \beta_0$$

$$E(b_i) = \beta_i$$

- For example, if we take out samples of 50, out of 1000 repeatedly, then after some repeated attempts, you would find that the average of all the betas; β_0 and β_i from the samples will equal to the actual (or the population) values of beta.
- If your estimator is biased, then the average will not equal the true parameter value in the population

PROPERTIES OF LEAST SQUARE ESTIMATES

2) Minimum variance

- An estimator is best when it has a small variance as compared with any other estimators obtained from other Econometrics methods
 - $\text{Var}(b) < \text{Var}\hat{\alpha}$
 - $E[b - E(b)]^2 < E[\hat{\alpha} - E(\hat{\alpha})]^2$
- Where $\hat{\alpha}$ is any estimator (not necessarily unbiased) of the true parameters α .
- The estimator that has less variance will have individual data points closer to the mean.

3) Efficiency

An estimator is efficient when it possess both the previous properties that is **unbiased and has minimum variance** as compare with any other unbiased estimator

Symbolically b is efficient if

a. $E(b) = b$

b. $E[b - E(b)]^2 < E[\hat{\alpha} - E(\hat{\alpha})]^2$

4) Linear Estimator

An estimator is linear if it is a linear function of the sample observation, which is determined by the linear combination of the sample data.

- The *linear* property of OLS estimator means that OLS belongs to that class of estimators, which are linear in Y , the dependent variable

5) **BLUE (Best, linear, unbiased and efficient)**

An estimator \hat{b} is blue if it is linear and has small variance as compare with all other unbiased estimator of the true b .The blue estimator has minimum variance with the class of linear unbiased

Assumptions Linear regression Model or (OLS)

1) The regression model is linear in the parameters

$$Y_i = \beta_1 + \beta_2 X_i + u_i$$

2) Zero mean value of disturbance u_i

Given the value of X , the mean, or expected, value of the random disturbance term u_i is zero.

Symbolically we have, $E(u_i | X_i) = 0$

3) Homoscedasticity or equal variance of u

Given the value of X , the variance of u_i is the same for all observations. $\text{var}(u_i | X_i) = \sigma$

Assumptions Linear regression Model or (OLS)

4) **No autocorrelation between the disturbances**

Given any two X values, X_i and X_j ($i \neq j$), the correlation between any two u_i and u_j ($i \neq j$) is zero. Symbolically, $\text{Cov}(u_i, u_j | X_i, X_j) = 0$

5) **Zero covariance between u_i and X_i**

Assumption 6 states that the disturbance u and explanatory variable are uncorrelated

Assumptions Linear regression Model or (OLS)

- 6) The number of observations n must be greater than the number of parameters to be estimated
- 7) Variability in X values.
The X values in a given sample must not all be the same
- 8) The regression model is correctly specified
- 9) There is no perfect Multicollinearity.
That is, there are no perfect linear relationships among the explanatory variables.

Unit 3

Logistic Regression

Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1.**

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems.**

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

$\logit(p) = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$
In logistic regression, the dependent variable is in fact a logit, which is a log of odds,

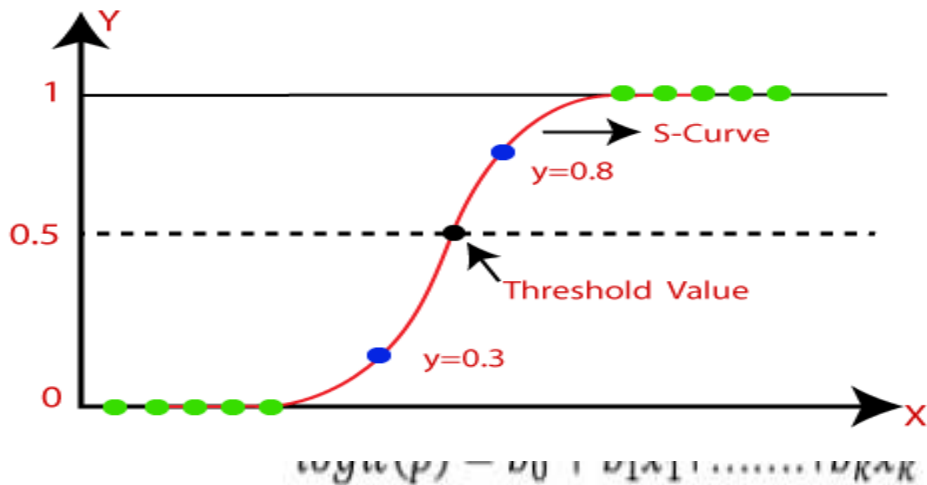
$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$



Very Important

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



In logistic regression, the dependent variable is in fact a logit, which is a log of odds,

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

$$p = \frac{e^{\text{logit}(p)}}{1 + e^{\text{logit}(p)}}$$

Values on Y-axis,
Dependent variable
Important equations

Logistic Regression is used when the dependent variable (target) is categorical.

For example,



- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

Linear Regression vs Logistic Regression Comparison Table

Let's discuss the top comparison between Linear Regression vs Logistic Regression

Linear Regression	Logistic Regression
It is used to solve regression problems	It is used to solve classification problems
It models the relationship between a dependent variable and one or more independent variable	It predicts the probability of an outcome that can only have two values at the output either 0 or 1
The predicted output is a continuous variable	The predicted output is a discrete variable
Predicted output Y can exceed 0 and 1 range	Predicted output Y lies within 0 and 1 range
	
Predicted output Y can exceed 0 and 1 range $Y = b_0 + b_1 \times X + e$	Predicted output $\ln \left(\frac{P}{1-P} \right) = b_0 + b_1 \times X_1 + b_2 \times X_2 + \dots + b_K \times X_K$

Application of Logistic Regression:

- Logistic regression is used when the response you want to predict/measure is categorical with two or more levels. Some examples are gender of a person , outcome of a football match
 - **Marketing:**
 - A marketing consultant wants to predict if the subsidiary of his company will make profit, loss or just break even depending on the characteristic of the subsidiary operations.
 - **Human Resources:**
 - The HR manager of a company wants to predict the absenteeism pattern of his employees based on their individual characteristic.
 - **Finance:**
 - A bank wants to predict if his customers would default based on the previous transactions and history.
-

Application of Logistic Regression:

- **Image Segmentation and Categorization**
- **Geographic Image Processing**
- **Handwriting recognition**
- **Healthcare :**
 - Analyzing a group of over million people for myocardial infarction within a period of 10 years is an application area of logistic regression.
 - Prediction whether a person is depressed or not based on bag of words from the corpus seems to be conveniently solvable using logistic regression and SVM.
 - It is one of the best tools used by statisticians, researchers and data scientists in predictive analytics.
- It is one of the best tools used by statisticians, researchers and data scientists in predictive analytics.

Logistic Regression Assumptions

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little or no multicollinearity.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

Lets understand What are Odds and Odds Ratio

Odds

- Let's begin with probability. Let's say that the probability of success is .8, thus
 - $p = .8$
- Then the probability of failure is
 - $q = 1 - p = .2$
- The odds of success are defined as
 - $\text{odds}(\text{success}) = p/q = .8/.2 = 4,$
 - that is, the odds of success are 4 to 1.
- We can also define the odds of failure
 - $\text{odds}(\text{failure}) = q/p = .2/.8 = .25,$
 - that is, the odds of failure are 1 to 4.

Odds Ratio

- Next, let's compute the odds ratio by
- $OR = \text{odds}(\text{success})/\text{odds}(\text{failure}) = 4/.25 = 16$
- The interpretation of this odds ratio would be that the odds of success are 16 times greater than for failure.
- Now if we had formed the odds ratio the other way around with odds of failure in the numerator, we would have gotten
- $OR = \text{odds}(\text{failure})/\text{odds}(\text{success}) = .25/4 = .0625$
- Here the interpretation is that the odds of failure are one-sixteenth the odds of success.

Logit

- Logit

- Natural log (e) of an odds
- Often called a *log odds*
 - *The logit scale is linear*

- Logits are continuous and are centered on zero (kind of like z-scores)

- $p = 0.50$, odds = 1, then logit = 0
- $p = 0.70$, odds = 2.33, then logit = 0.85
- $p = 0.30$, odds = .43, then logit = -0.85

- So conceptually putting things in our standard regression form:

- Log odds = $b_0 + b_1X$

- Now a one unit change in X leads to a b_1 change in the log odds

- In terms of odds: $odds(Y = 1) = e^{b_0 + b_1X}$

- In terms of probability: $Pr(Y = 1) = \frac{e^{b_0 + b_1X}}{1 + e^{b_0 + b_1X}}$ $Ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1X_1 + \beta_2X_2 + \dots + \beta_kX_k$

- Thus the logit, odds and probability are different ways of expressing the same thing

Model Construction/Best Curve fit for the data points

Model construction is nothing but finding the best fit curve that covers all the points with minimum cost/error.

The following 3 steps are needed for finding the best curve.

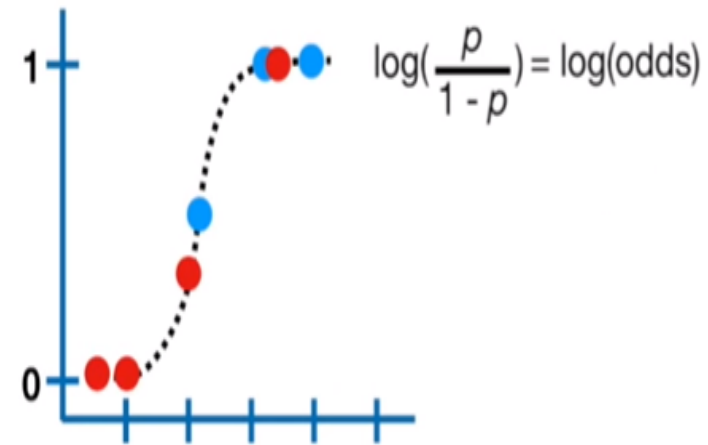
1. Find the logit values using logit function for the dependent variable

Formally, the model logistic regression model is that

$$\log \frac{p(x)}{1-p(x)} = \beta_0 + x \cdot \beta$$

Solving for p , this gives

$$p(x; b, w) = \frac{e^{\beta_0 + x \cdot \beta}}{1 + e^{\beta_0 + x \cdot \beta}} = \frac{1}{1 + e^{-(\beta_0 + x \cdot \beta)}}$$



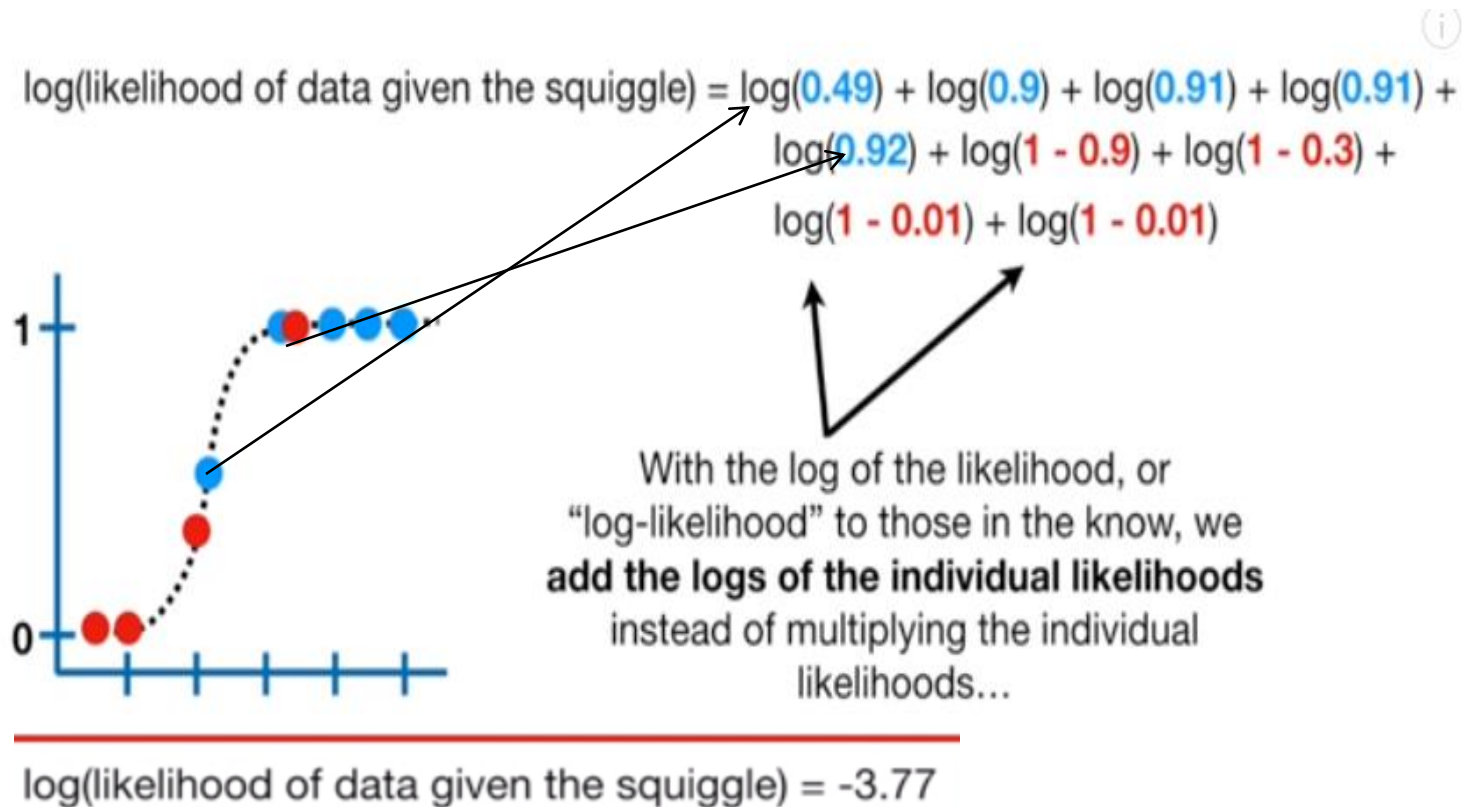
2. Calculate the Maximum likelihood:

Maximum likelihood is calculating the value using the below equations for the data points which fits the curve

$$L(\beta_0, \beta) = \prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

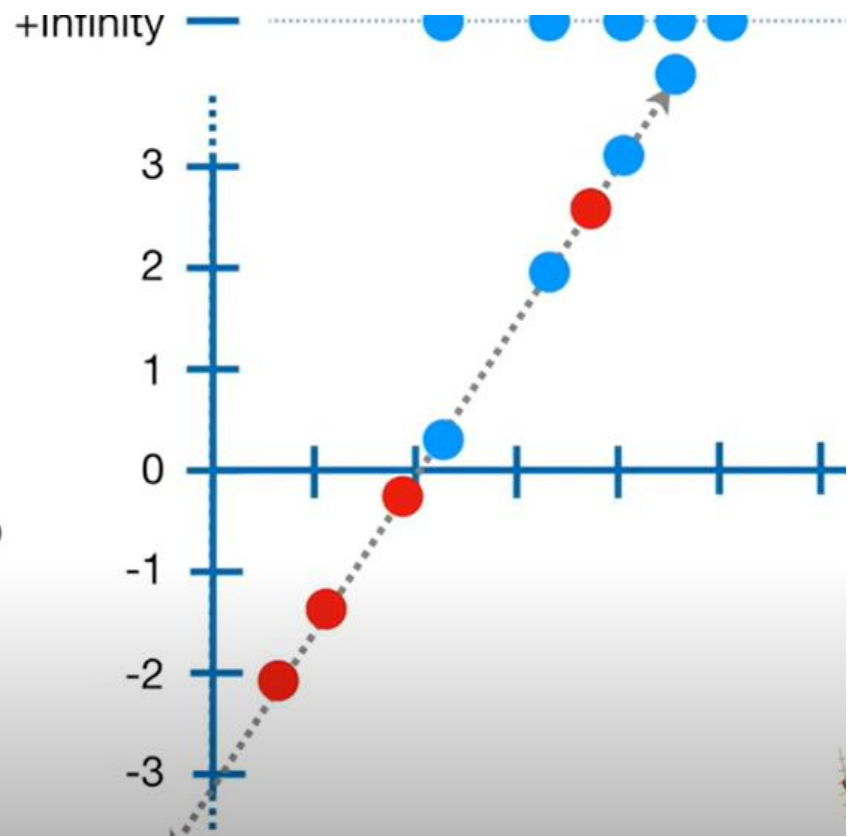
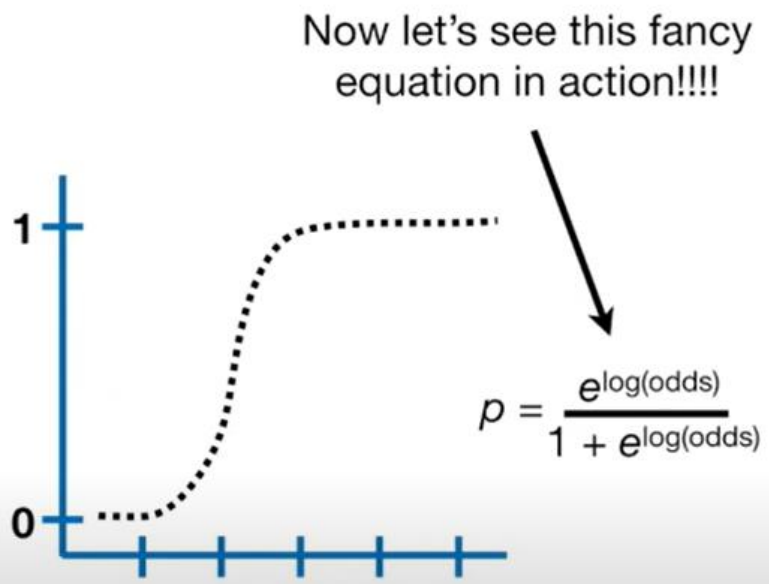
$$\ell(\beta_0, \beta) = \sum_{i=1}^n y_i \log p(x_i) + (1 - y_i) \log 1 - p(x_i)$$

The log likelihood of the data points are calculate for the s-shaped curve or squiggle.

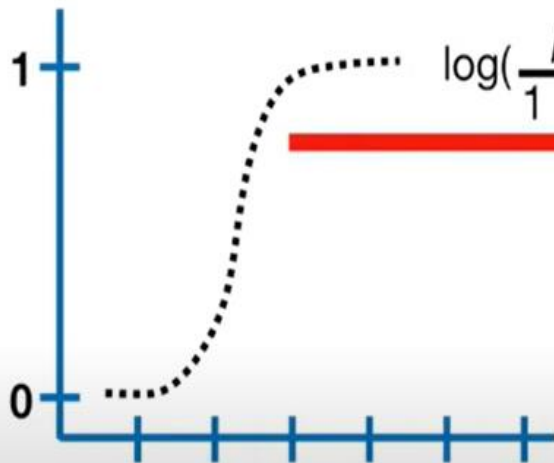


3. Repeat the same process for different curves

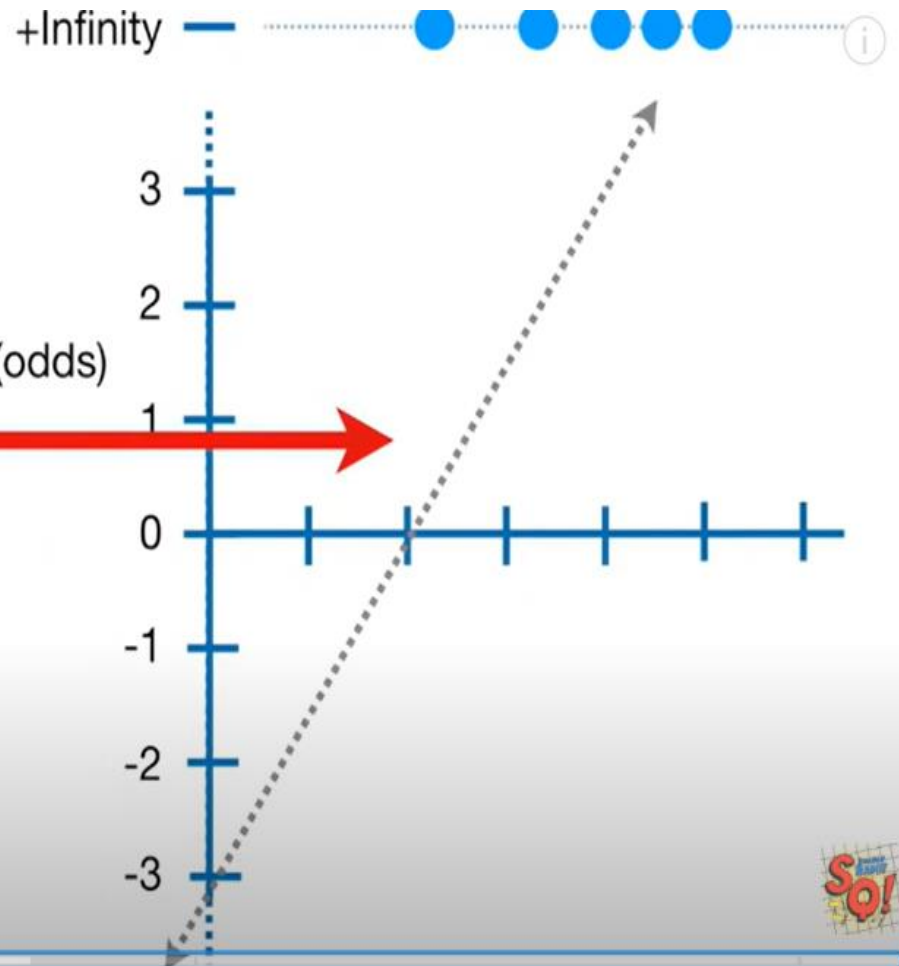
Note: The curve which gives maximum values is selected as fit curve

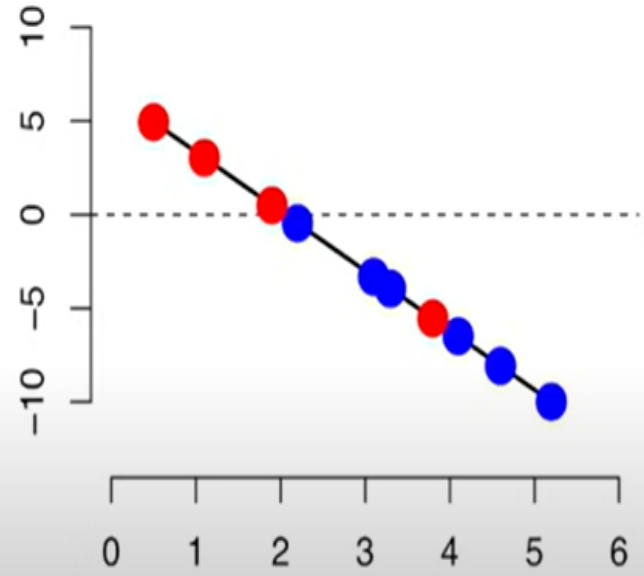
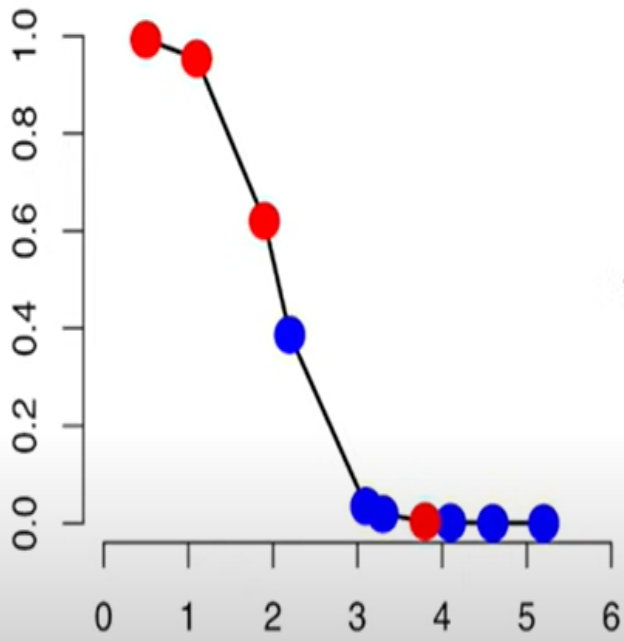


...and the new y-axis transforms the squiggly line into a straight line.

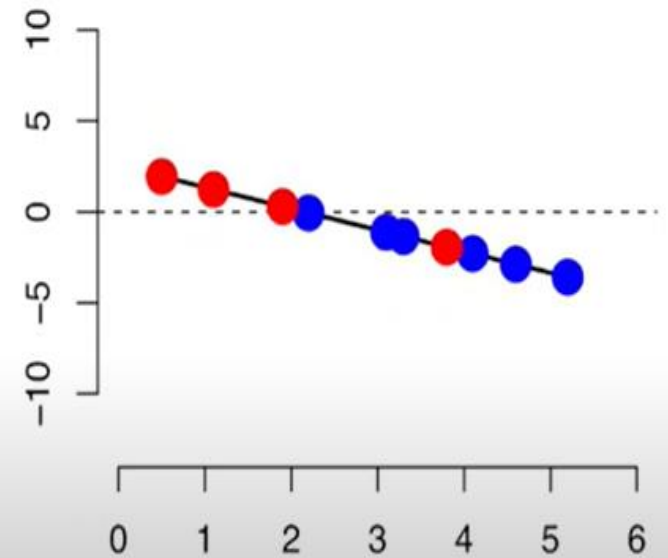
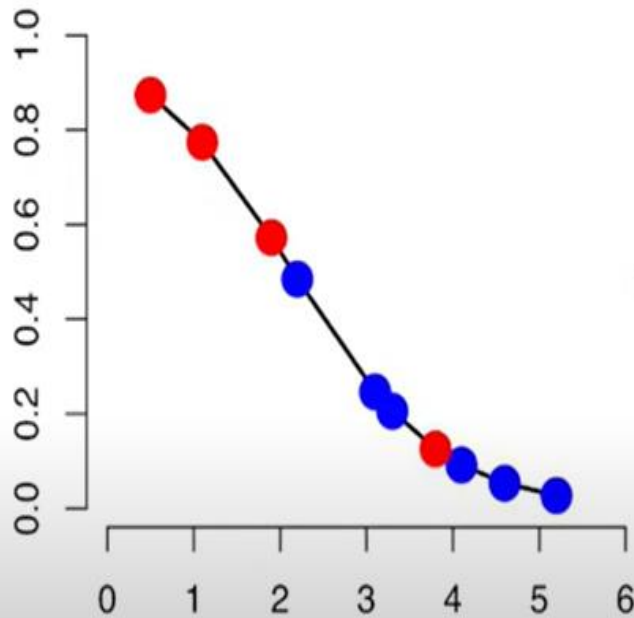


$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

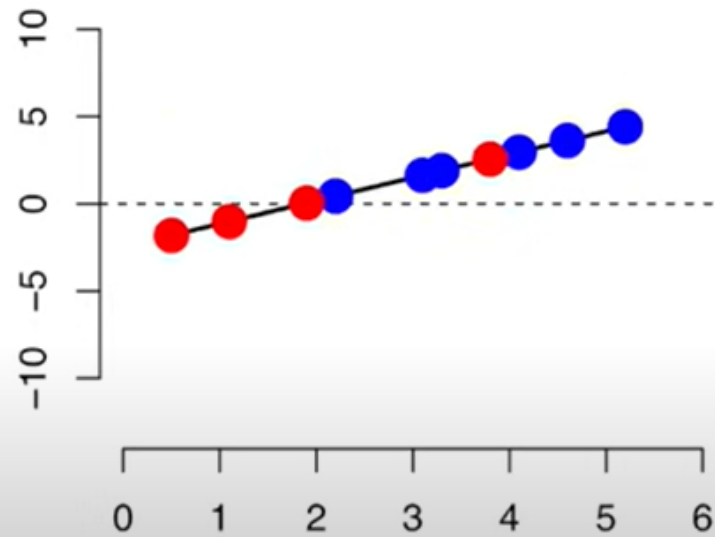
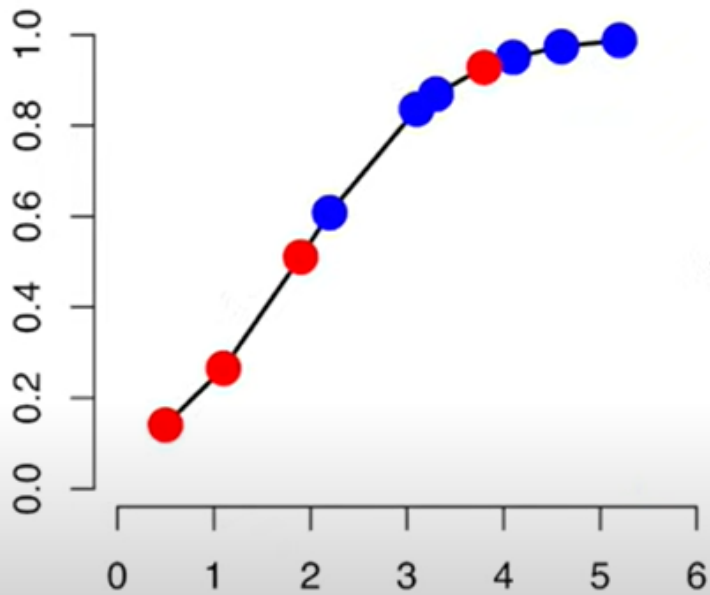




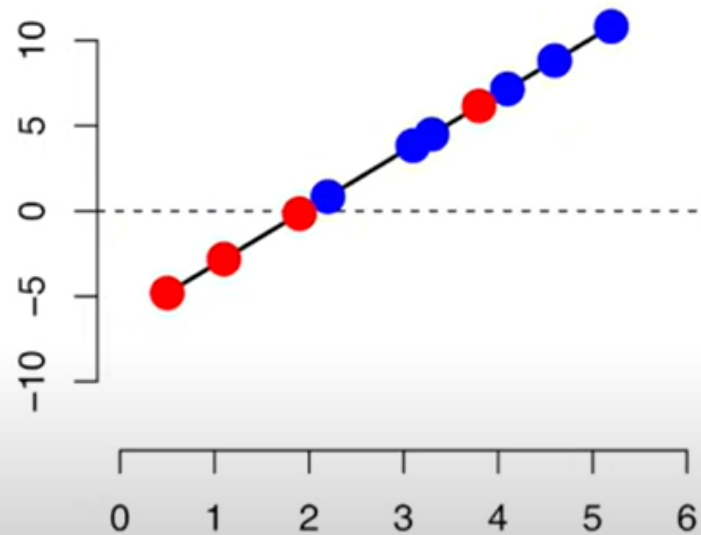
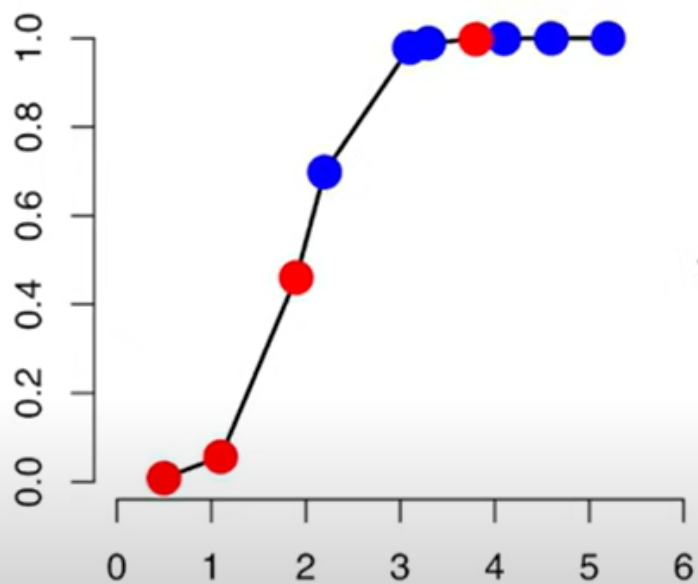
...and transforming it to probabilities and calculating the log-likelihood.



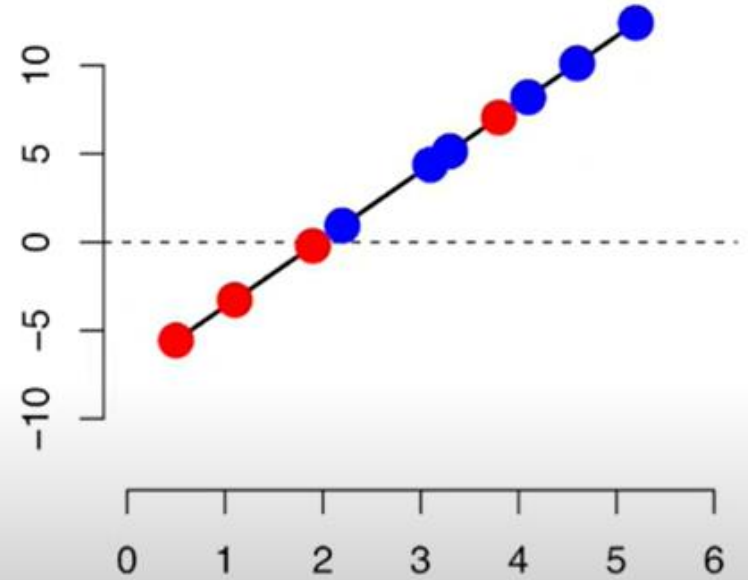
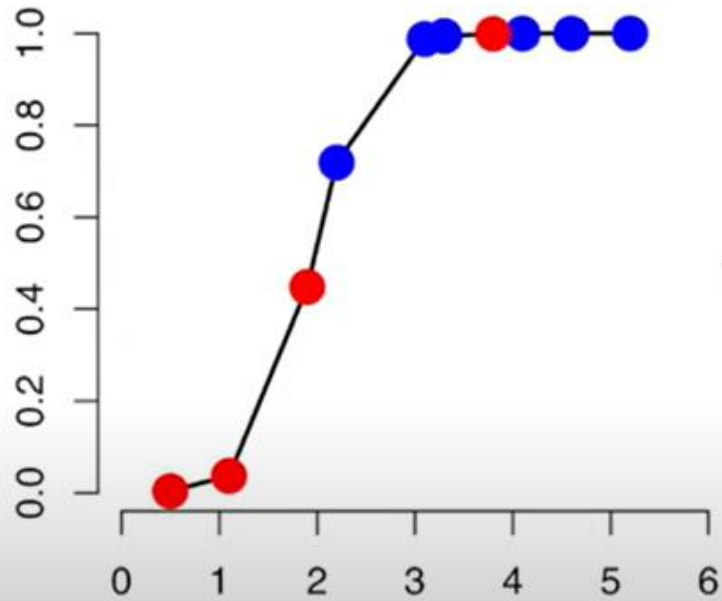
...and transforming it to probabilities and calculating the log-likelihood.



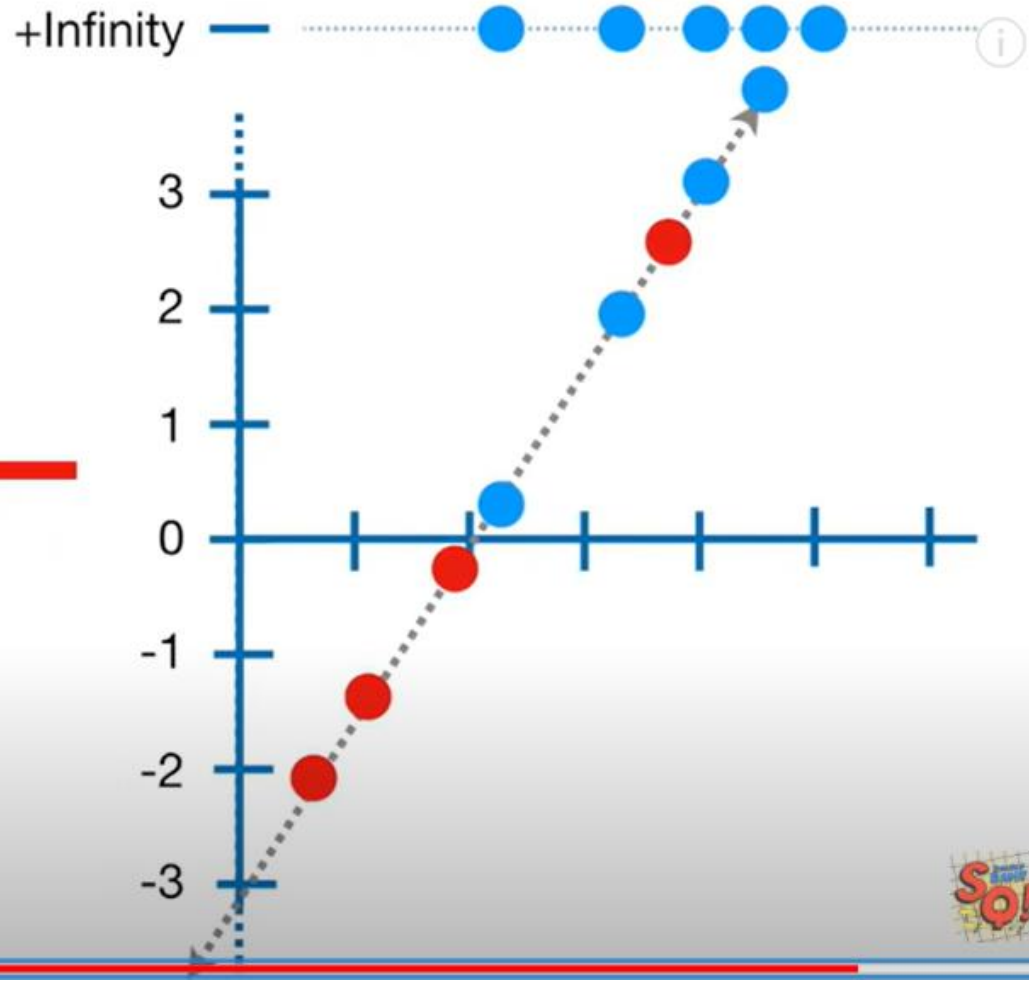
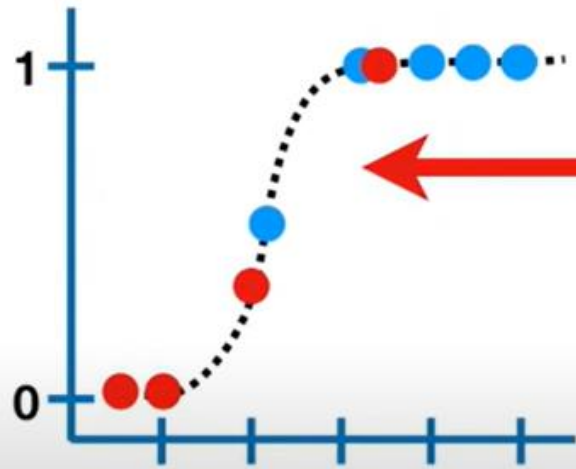
...and transforming it to probabilities and calculating the log-likelihood.



...and transforming it to probabilities and calculating the log-likelihood.



Ultimately we get a line that maximizes the likelihood and that's the one chosen to have the best fit.



Model fit statistics of Logistic Regression/:

It tells us whether the best fit curve we got is good enough or not or how much it is good.

1. Calculate R Square: R Square is calculated using the equation

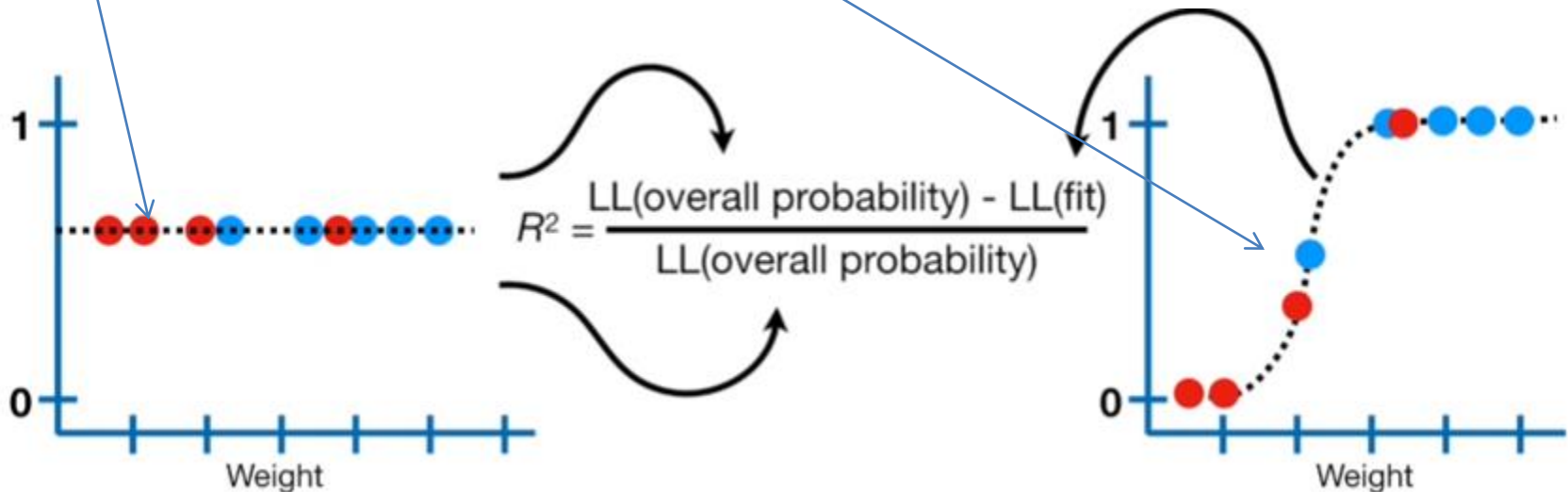
$$R^2 = \frac{LL(\text{overall probability}) - LL(\text{fit})}{LL(\text{overall probability})}$$

R Square like between 0 & 1.

0 for poor model, 1 for good model

LL(Overall probability): it means calculating the values using odds of happening and odds of not happening without taking the independent variable (X axis) into consideration.

LL(fit): it means calculating the values using odds of happening and odds of not happening taking the independent variable into consideration.



2. Calculate P-Value

A p-value is the probability that random chance generated the data, or something else that is equal or rarer.

For example: The probability of getting 2 heads when two coins are flipped is 0.25

$$\frac{\text{HH}}{\text{HH, HT, TH, TT}} = \frac{1}{4} = 0.25$$

The P-value for getting 2 heads when two coins are flipped is 0.5. It takes into consideration the chance of getting two tails also. As it is similar to getting two heads

$$\frac{\text{HH}}{\text{HH, HT, TH, TT}} = \frac{1}{4} = 0.25$$

the p-value for **HH** = 0.5

+

$$\frac{\text{TT}}{\text{HH, HT, TH, TT}} = \frac{1}{4} = 0.25$$

P-Value for logistic regression is calculated using the equation

$$2(LL(\text{fit}) - LL(\text{overall probability})) = \text{A Chi-squared value with degrees of freedom equal to the difference in the number of parameters in the two models.}$$

LL(fit) has 2 parameters (intercept and slope) as it takes into account the independent Variable X.

LL(overall probability) has 1 parameter as it takes only the Y values.

Therefore,

$$\text{Degree of freedom} = 2 - 1 = 1$$

Note: A low p-value means a predictor that has a low p-value is likely to be meaningful addition to our model because the change in the predictor's value are related to the change in response

A high p value suggests that changes in the predictor are not associated with changes in responses

3. ROC and AUC curves

The concept of ROC and AUC builds upon the knowledge of Confusion Matrix

		Actual	
		Has Heart Disease	Doesnot have Heart Disease
Predicted	Has Heart Disease	True Positives	False Positives
	Doesnot have Heart Disease	False Negatives	True Negatives

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}$$

- **True Positives (TP):** People who *had heart disease* and were also predicted to have heart disease.
- **True negatives (TN):** People who *did not have heart disease* and were also predicted to not have heart disease.
- **False negatives (FN):** People who have heart disease but the prediction says they don't.
- **False positives (FP):** People who *did not have heart disease* but the prediction says they do.

ROC Graphs

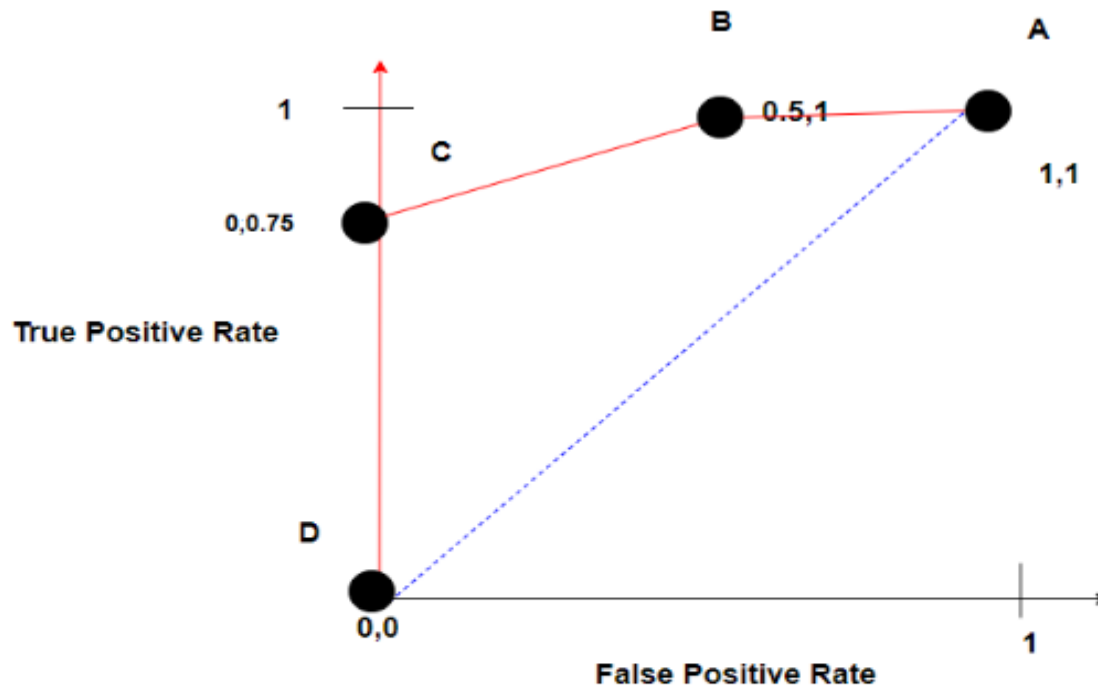
ROC(Receiver Operator Characteristic Curve) can help in deciding the best threshold value. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis).

$$\text{True Positive Rate} = \text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{False Positive Rate} = (1 - \text{Specificity}) = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

True Positive Rate indicates what proportion of people 'with heart disease' were correctly classified.

False Positive Rate indicates the proportion of people classified as 'not having heart disease', that are False Positives.

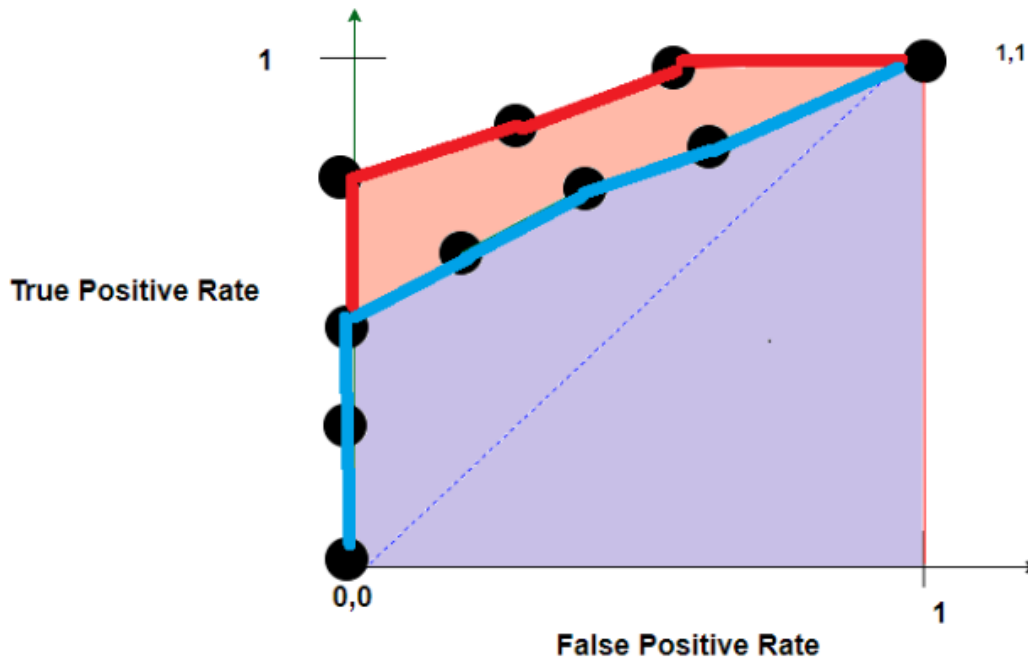


Just by glancing over the graph, we can conclude that threshold C is better than threshold B and depending on how many False Positives that we are willing to accept, we can choose the optimal threshold.

Note: The intention of ROC curve is to find the best threshold value that indicates how many False positives that we are willing to accept.

AUC

AUC stands for **Area under the curve**. AUC gives the rate of successful classification by the logistic model. The AUC makes it easy to compare the ROC curve of one model to another.



The **AUC** for the red **ROC** curve is greater than the **AUC** for the blue **ROC** curve. This means that the Red curve is better. If the Red ROC curve was generated by say, a Random Forest and the Blue ROC by Logistic Regression we could conclude that the Random classifier did a better job in classifying the patients.

Summary

- Logistic Regression Def. & Equation to Represent it. The model graph
- Building Model/Constructing model for Logistic regression.
 1. Logit function (Odds)
 2. Maximum likelihood. (equation)
- Model fit statistics
 1. R Square value
 2. P value
 3. Roc & AUC curve
- Difference b/w linear and logistic Regression
- Application of Logistic Regression

Introduction to Properties of OLS Estimators

Linear regression models have several applications in real life. In econometrics, [Ordinary Least Squares \(OLS\)](#) method is widely used to estimate the parameters of a linear regression model. For the validity of OLS estimates, there are assumptions made while running linear regression models.

A1. The linear regression model is “linear in parameters.”

A2. There is a random sampling of observations.

A3. The conditional mean should be zero.

A4. There is no multi-collinearity (or perfect collinearity).

A5. Spherical errors: There is homoscedasticity and no auto-correlation

A6: Optional Assumption: Error terms should be normally distributed.

These assumptions are extremely important because violation of any of these assumptions would make OLS estimates unreliable and incorrect. Specifically, a violation would result in incorrect signs of OLS estimates, or the variance of OLS estimates would be unreliable, leading to confidence intervals that are too wide or too narrow.

This being said, it is necessary to investigate why OLS estimators and its assumptions gather so much focus. In this article, the properties of OLS model are discussed. First, the famous Gauss-Markov Theorem is outlined. Thereafter, a detailed description of the properties of the OLS model is described. In the end, the article briefly talks about the applications of the properties of OLS in econometrics.

The Gauss-Markov Theorem

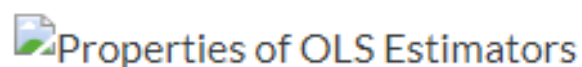
The Gauss-Markov Theorem is named after Carl Friedrich Gauss and Andrey Markov.

Let the regression model be: $Y = \beta_0 + \beta_i X_i + \varepsilon$

Let $\hat{\beta}_0$ and $\hat{\beta}_i$ be the OLS estimators of β_0 and β_i

According to the Gauss-Markov Theorem, under the assumptions A_1 to A_5 of the linear regression model, the OLS estimators $\hat{\beta}_0$ and $\hat{\beta}_i$ are the Best Linear Unbiased Estimators (BLUE) of β_0 and β_i .

In other words, the OLS estimators β_0 and β_i have the minimum variance of all linear and unbiased estimators of β_0 and β_i . BLUE summarizes the properties of OLS regression. These properties of OLS in econometrics are extremely important, thus making OLS estimators one of the strongest and most widely used estimators for unknown parameters. This theorem tells that one should use OLS estimators not only because it is unbiased but also because it has minimum variance among the class of all linear and unbiased estimators.



Properties of OLS Regression Estimators in Detail

Property 1: Linear

This property is more concerned with the estimator rather than the original equation that is being estimated. In assumption A_1 , the focus was that the linear regression should be “linear in parameters.” However, the *linear* property of OLS estimator means that OLS belongs to that class of estimators, which are linear in Y , the dependent variable. Note that OLS estimators are linear only with respect to the dependent variable and not necessarily with respect to the independent variables. The *linear* property of OLS estimators doesn’t depend only on assumption A_1 but on all assumptions A_1 to A_5 .

Property 2: Unbiasedness

If you look at the regression equation, you will find an error term associated with the regression equation that is estimated. This makes the dependent variable also random. If an estimator uses the dependent variable, then that estimator would also be a random number. Therefore, before describing what unbiasedness is, it is important to mention that unbiasedness property is a property of the estimator and not of any sample.

Unbiasedness is one of the most desirable properties of any estimator. The estimator should ideally be an unbiased estimator of true parameter/population values.

Consider a simple example: Suppose there is a population of size 1000, and you are taking out samples of 50 from this population to estimate the population parameters. Every time you take a sample, it will have the different set of 50 observations and, hence, you would estimate different values of β_0 and β_i . The unbiasedness property of OLS method says that when you take out samples of 50 repeatedly, then after some repeated attempts, you would find that the average of all the β_0 and β_i from the samples will equal to the actual (or the population) values of β_0 and β_i .

Mathematically,

$$E(b_0) = \beta_0$$

$$E(b_i) = \beta_i$$

Here, 'E' is the expectation operator.

In layman's term, if you take out several samples, keep recording the values of the estimates, and then take an average, you will get very close to the correct population value. If your estimator is biased, then the average will not equal the true parameter value in the population.

The unbiasedness property of OLS in Econometrics is the basic minimum requirement to be satisfied by any estimator. However, it is not sufficient for the reason that most times in real-life applications, you will not have the luxury of taking out repeated samples. In fact, only one sample will be available in most cases.

Property 3: Best: Minimum Variance

First, let us look at what efficient estimators are. The efficient property of any estimator says that the estimator is the *minimum variance unbiased* estimator. Therefore, if you take all the unbiased estimators of the unknown population parameter, the estimator will have the least variance. The estimator that has less variance will have individual data points closer to the mean. As a result, they will be more likely to give better and accurate results than other estimators having higher variance. In short:

1. If the estimator is unbiased but doesn't have the least variance – it's not the best!
2. If the estimator has the least variance but is biased – it's again not the best!
3. If the estimator is both unbiased and has the least variance – it's the best estimator.

Now, talking about OLS, OLS estimators have the *least variance* among the class of all *linear unbiased* estimators. So, this property of OLS regression is less strict than efficiency property. Efficiency property says least variance among all unbiased estimators, and OLS estimators have the least variance among all linear and unbiased estimators.

Let b_o be the OLS estimator, which is linear and unbiased. Let b_o^* be any other estimator of β_o , which is also linear and unbiased. Then,

$$\text{Var}(b_o) < \text{Var}(b_o^*)$$

Let b_i be the OLS estimator, which is linear and unbiased. Let b_i^* be any other estimator of β_i , which is also linear and unbiased. Then,

$$\text{Var}(b_i) < \text{Var}(b_i^*)$$


The above three properties of OLS model makes OLS estimators BLUE as mentioned in the Gauss-Markov theorem.

It is worth spending time on some other estimators' properties of OLS in econometrics. The properties of OLS described below are asymptotic properties of OLS estimators. So far, finite sample properties of OLS regression were discussed. These properties tried to study the behavior of the OLS estimator under the assumption that you can have several samples and, hence, several estimators of the same unknown population parameter. In short, the properties were that the average of these estimators in different samples should be equal to the true population parameter (unbiasedness), or the average distance to the true parameter value should be the least (efficient). However, in real life, you will often have just one sample. Hence, asymptotic properties of OLS model are discussed, which studies how OLS estimators behave as sample size increases. Keep in mind that sample size should be large.

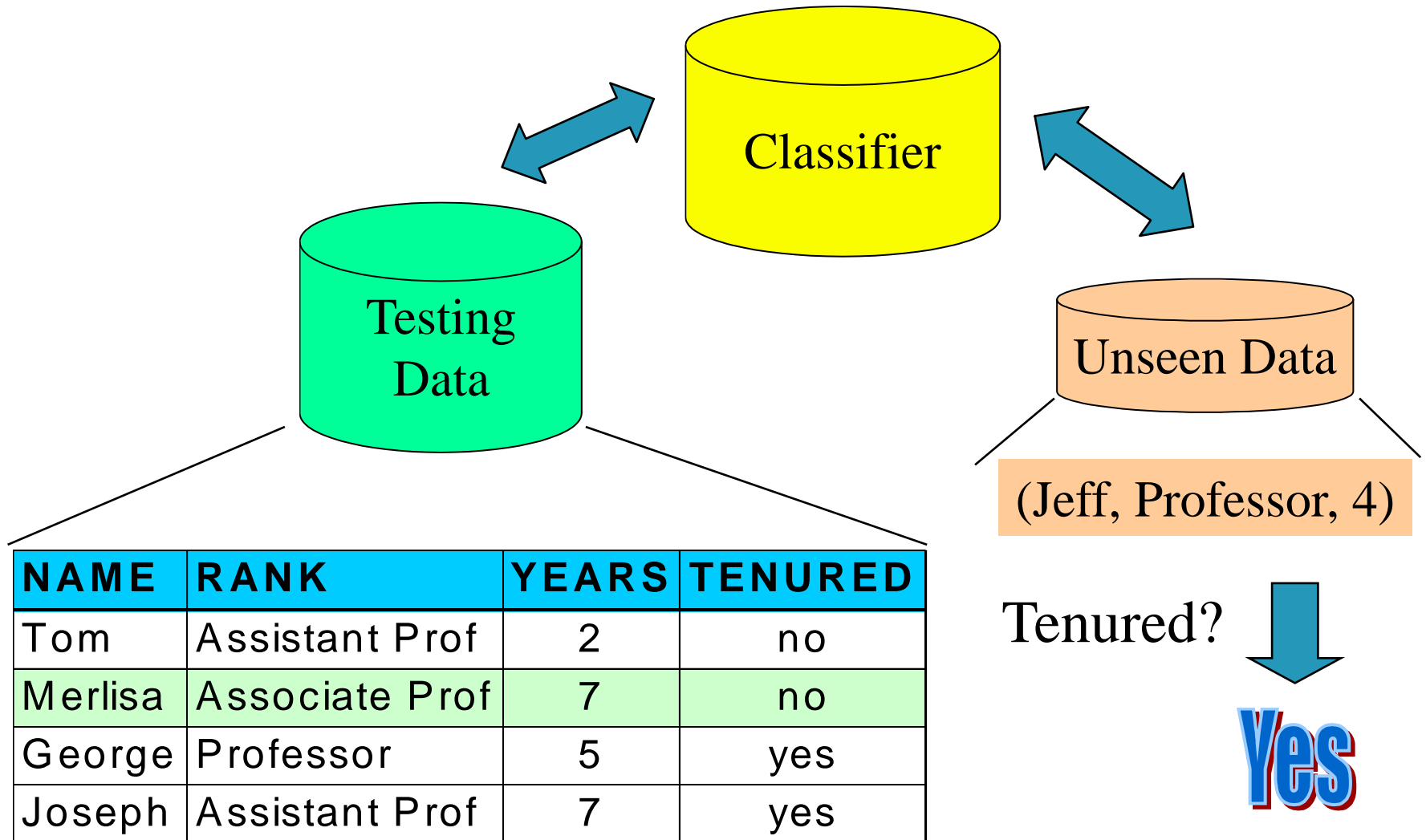
Conclusion

To conclude, linear regression is important and widely used, and OLS estimation technique is the most prevalent. In this article, the properties of OLS estimators were discussed because it is the most widely used estimation technique. OLS estimators are BLUE (i.e. they are linear, unbiased and have the least variance among the class of all linear and unbiased estimators). Amidst all this, one should not forget the Gauss-Markov Theorem (i.e. the estimators of OLS model are BLUE) holds only if the assumptions of OLS are satisfied. Each assumption that is made while studying OLS adds restrictions to the model, but at the same time, also allows to make stronger statements regarding OLS. So, whenever you are planning to use a linear regression model using OLS, always check for the OLS assumptions. If the OLS assumptions are satisfied, then life becomes simpler, for you can directly use OLS for the best results – thanks to the Gauss-Markov theorem!


Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts 
- Decision Tree Induction
- Bayes Classification Methods
- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy:
Ensemble Methods
- Summary

Process (2): Using the Model in Prediction

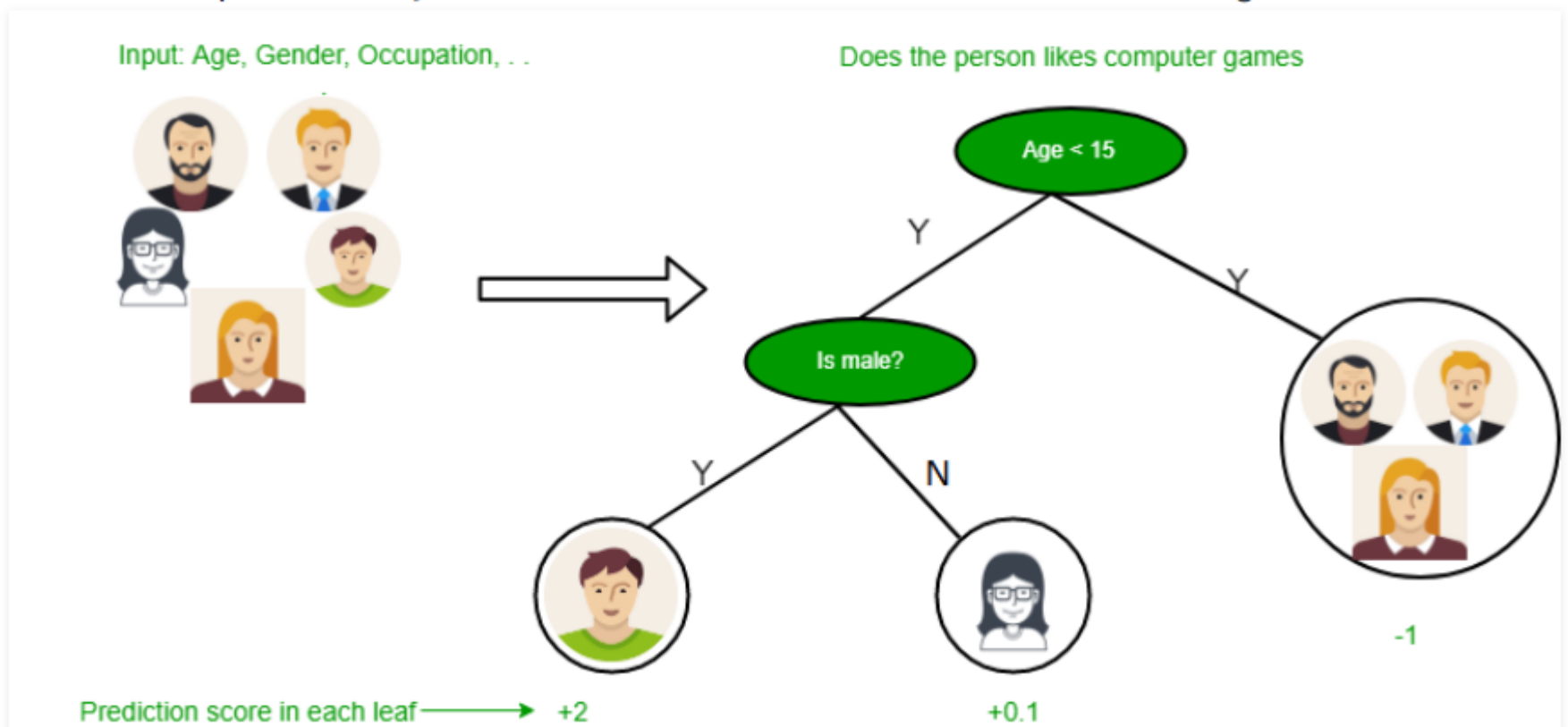


Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- Decision Tree Induction 
- Bayes Classification Methods
- Rule-Based Classification
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy:
Ensemble Methods
- Summary

Decision Tree Def.

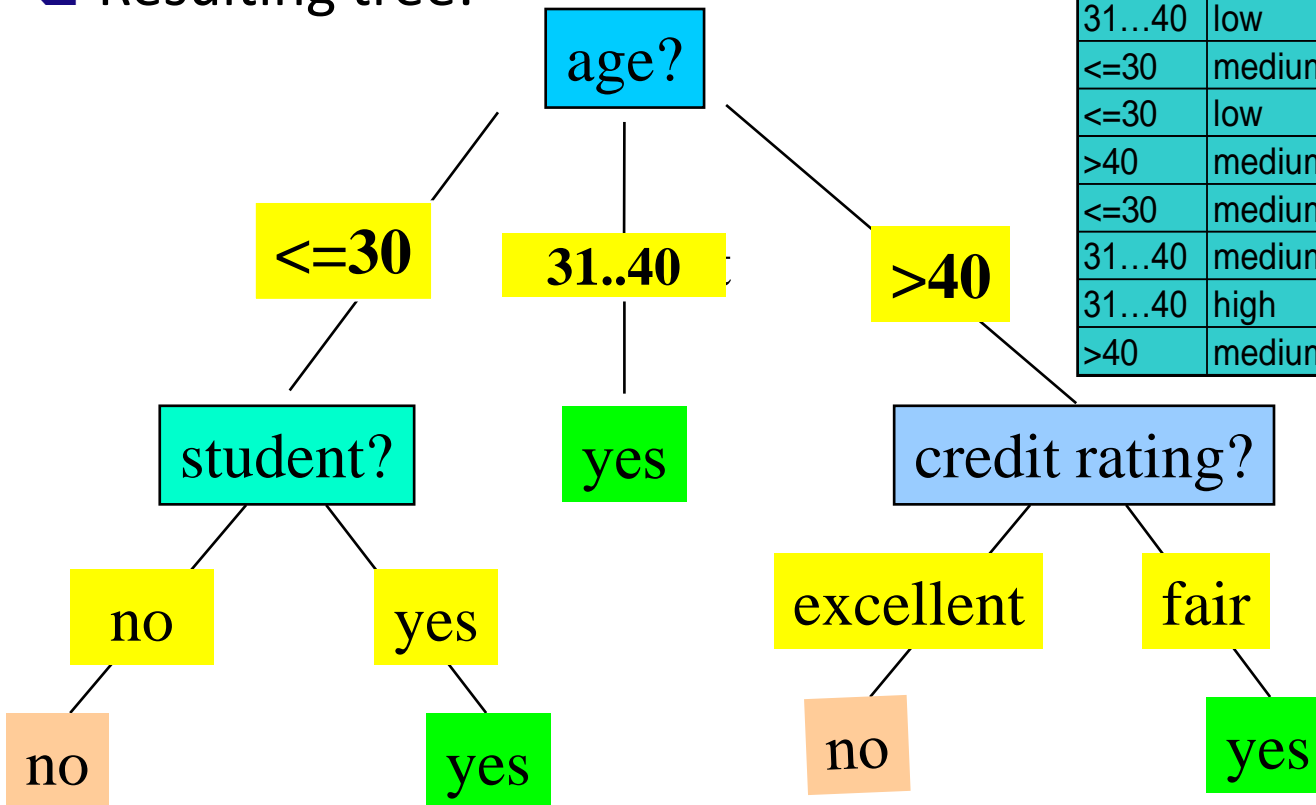
- Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
- Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
- We can represent any boolean function on discrete attributes using the decision tree.



Decision Tree Induction: An Example

- ❑ Training data set: Buys_computer
- ❑ The data set follows an example of Quinlan's ID3 (Playing Tennis)
- ❑ Resulting tree:

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no



Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
 - Tree is constructed in a **top-down recursive divide-and-conquer manner**
 - At start, all the training examples are at the root
 - Attributes are categorical (if continuous-valued, they are discretized in advance)
 - Examples are partitioned recursively based on selected attributes
 - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., **information gain**)
- Conditions for stopping partitioning
 - All samples for a given node belong to the same class
 - There are no remaining attributes for further partitioning – **majority voting** is employed for classifying the leaf
 - There are no samples left

Brief Review of Entropy

■ Entropy (Information Theory)

- A measure of uncertainty associated with a random variable

- Calculation: For a discrete random variable Y taking m distinct values $\{y_1, \dots, y_m\}$,

- $H(Y) = -\sum_{i=1}^m p_i \log(p_i)$, where $p_i = P(Y = y_i)$

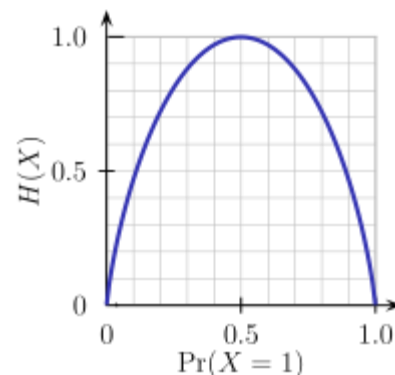
- Interpretation:

- Higher entropy => higher uncertainty

- Lower entropy => lower uncertainty

■ Conditional Entropy

- $H(Y|X) = \sum_x p(x)H(Y|X = x)$



m = 2

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i , estimated by $|C_{i,D}|/|D|$

- **Expected information** (entropy) needed to classify a tuple in D :

$$Info(D) = -\sum_{i=1}^m p_i \log_2(p_i)$$

- **Information** needed (after using A to split D into v partitions) to classify D :

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$

- **Information gained** by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection: Information Gain

■ Class P: buys_computer = “yes”

■ Class N: buys_computer = “no”

$$Info(D) = I(9,5) = -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
≤ 30	2	3	0.971
31...40	4	0	0
> 40	3	2	0.971

age	income	student	credit_rating	buys_computer
≤ 30	high	no	fair	no
≤ 30	high	no	excellent	no
31...40	high	no	fair	yes
> 40	medium	no	fair	yes
> 40	low	yes	fair	yes
> 40	low	yes	excellent	no
31...40	low	yes	excellent	yes
≤ 30	medium	no	fair	no
≤ 30	low	yes	fair	yes
> 40	medium	yes	fair	yes
≤ 30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
> 40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$ means “age ≤ 30 ” has 5 out of 14 samples, with 2 yes’es and 3 no’s. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit_rating) = 0.048$$

Computing Information-Gain for Continuous-Valued Attributes

- Let attribute A be a continuous-valued attribute
- Must determine the *best split point* for A
 - Sort the value A in increasing order
 - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
 - $(a_i + a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
 - The point with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
 - D1 is the set of tuples in D satisfying $A \leq \text{split-point}$, and D2 is the set of tuples in D satisfying $A > \text{split-point}$

Training Set

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Algorithm

- Calculate the **Entropy** of every attribute using the data set.
- Split the set into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum).
- Make a decision tree node containing that attribute.
- Recurse on subsets using remaining attributes.

Entropy

- In order to define **Information Gain** precisely, we need to discuss **Entropy** first.
- A formula to calculate the homogeneity of a sample.
- A completely homogeneous sample has entropy of 0 (leaf node).
- An equally divided sample has entropy of 1.
- The formula for entropy is:

$$\text{Entropy}(S) = -\sum p(I) \log_2 p(I)$$

- where $p(I)$ is the proportion of S belonging to class I . \sum is over total outcomes.

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

• Example 1

If S is a collection of 14 examples with 9 YES and 5 NO examples then

$$\begin{aligned} \text{Entropy}(S) &= - (9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) \\ &= 0.940 \end{aligned}$$

Information Gain

- The information gain is based on the decrease in entropy after a dataset is split on an attribute.
- The formula for calculating information gain is:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \left(\left(\frac{|S_v|}{|S|} \right) * \text{Entropy}(S_v) \right)$$

Where, S_v = subset of S for which attribute A has value v

- $|S_v|$ = number of elements in S_v
- $|S|$ = number of elements in S

Procedure

- First the entropy of the total dataset is calculated.
- The dataset is then split on the different attributes.
- The entropy for each branch is calculated.
- Then it is added proportionally, to get total entropy for the split.
- The resulting entropy is subtracted from the entropy before the split.
- The result is the Information Gain, or decrease in entropy.
- The attribute that yields the largest IG is chosen for the decision node.

Our Problem :

Name	Gender	Car Ownership	Travel Cost	Income Level	Transportation
Abhi	Male	1	Standard	High	?
Pavi	Male	0	Cheap	Medium	?
Ammu	Female	1	Cheap	High	?

Calculate the entropy of the total dataset

- First compute the Entropy of given training set.

Probability

Bus : $4/10 = 0.4$

Train : $3/10 = 0.3$

Car : $3/10 = 0.3$

$$E(S) = -P(I) \log_2 P(I)$$

$$E(S) = -(0.4) \log_2 (0.4) - (0.3) \log_2 (0.3) - (0.3) \log_2 (0.3) = 1.571$$

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car



Attributes	Classes
Gender	Transportation
Male	Bus
Male	Bus
Female	Train
Female	Bus
Male	Bus
Male	Train
Female	Train
Female	Car
Male	Car
Female	Car

Split the dataset on 'Gender' attribute

Attributes	Classes
Gender	Transportation
Male	Bus
Male	Bus
Male	Bus
Male	Train
Male	Car

Probability
Bus: $3/5 = 0.6$
Train: $1/5 = 0.2$
Car: $1/5 = 0.2$

Attributes	Classes
Gender	Transportation
Female	Train
Female	Bus
Female	Train
Female	Car
Female	Car

Probability
Bus: $1/5 = 0.2$
Train: $2/5 = 0.4$
Car: $2/5 = 0.4$

$$\begin{aligned} \text{Gain}(S,A) &= E(S) - I(S,A) \\ I(S,A) &= 1.522 * (5/10) + \\ & 1.371 * (5/10) \end{aligned}$$

$$\begin{aligned} \text{Gain}(S,A) &= 1.571 - 1.447 \\ &= 0.12 \end{aligned}$$

$$\begin{aligned} E(S_1) &= -0.6 \log_2(0.6) - 0.2 \log_2(0.2) - 0.2 \log_2(0.2) \\ &= 1.522 \end{aligned}$$

$$\begin{aligned} E(S_2) &= -0.2 \log_2(0.2) - 0.4 \log_2(0.4) - 0.4 \log_2(0.4) \\ &= 1.371 \end{aligned}$$

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car



Attributes	Classes
Car Ownership	Transportation
0	Bus
1	Bus
1	Train
0	Bus
1	Bus
0	Train
1	Train
1	Car
2	Car
2	Car

Split the dataset on 'ownership' attribute

Attributes	Classes
Car	Transportation
Ownership	
0	Bus
0	Bus
0	Train

Attributes	Classes
Car	Transportation
Ownership	
1	Bus
1	Train
1	Bus
1	Train
1	Car

Attributes	Classes
Car	Transportation
Ownership	
2	Car
2	Car

$$\begin{aligned} \text{Gain}(S,A) &= E(S) - I(S,A) \\ I(S,A) &= 0.918 * (3/10) + \\ & 1.522 * (5/10) + 0 * (2/10) \end{aligned}$$

$$\begin{aligned} \text{Gain}(S,A) &= 1.571 - 1.0364 \\ &= 0.534 \end{aligned}$$

Probability
Bus : $2/3 = 0.6$
Train: $1/3 = 0.3$
Car: $0/3 = 0$
Entropy = 0.918

Probability
Bus : $2/5 = 0.4$
Train: $2/5 = 0.4$
Car: $1/5 = 0.2$
Entropy = 1.522

Probability
Bus : $0/2 = 0$
Train: $0/2 = 0$
Car: $2/2 = 1$
Entropy = 0

- If we choose **Travel Cost** as splitting attribute,

-Entropy for Cheap = 0.722

Standard = 0

Expensive = 0

IG = 1.21

- If we choose **Income Level** as splitting attribute,

-Entropy for Low = 0

Medium = 1.459

High = 0

IG = 0.695

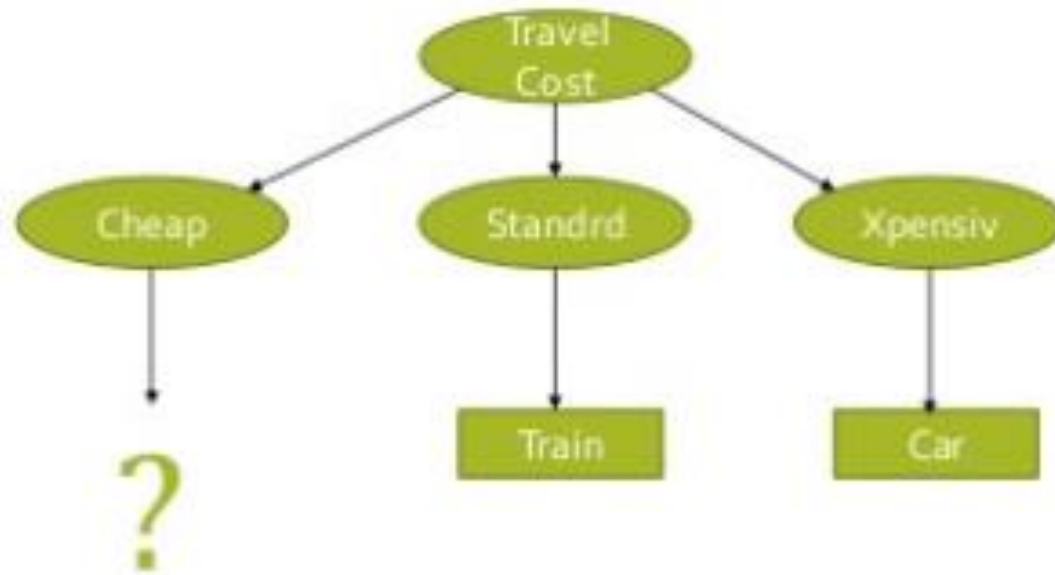
Attribute	Information Gain
Gender	0.125
Car Ownership	0.534
Travel Cost	1.21
Income Level	0.695

Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Cheap	Male	0	Low	Bus
Cheap	Male	1	Medium	Bus
Cheap	Female	1	Medium	Train
Cheap	Female	0	Low	Bus
Cheap	Male	1	Medium	Bus

Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Standard	Male	0	Medium	Train
Standard	Female	1	Medium	Train

Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Expensive	Female	1	High	Car
Expensive	Male	2	Medium	Car
Expensive	Female	2	High	Car

Diagram : Decision Tree



Iteration on Subset of Training Set

Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Cheap	Male	0	Low	Bus
Cheap	Male	1	Medium	Bus
Cheap	Female	1	Medium	Train
Cheap	Female	0	Low	Bus
Cheap	Male	1	Medium	Bus

Probability

Bus : $4/5 = 0.8$

Train: $1/5 = 0.2$

Car: $0/5 = 0$

$$E(S) = -P(I) \log_2 P(I)$$

$$E(S) = -(0.8) \log_2 (0.8) - (0.2) \log_2 (0.2) = 0.722$$

- If we choose **Gender** as splitting attribute,

-Entropy for Male = 0

Female = 1

IG = 0.322

- If we choose **Car Ownership** as splitting attribute,

-Entropy for 0 = 0

1 = 0.918

IG = 0.171

- If we choose **Income Level** as splitting attribute,

-Entropy for Low = 0

Medium = 0.918

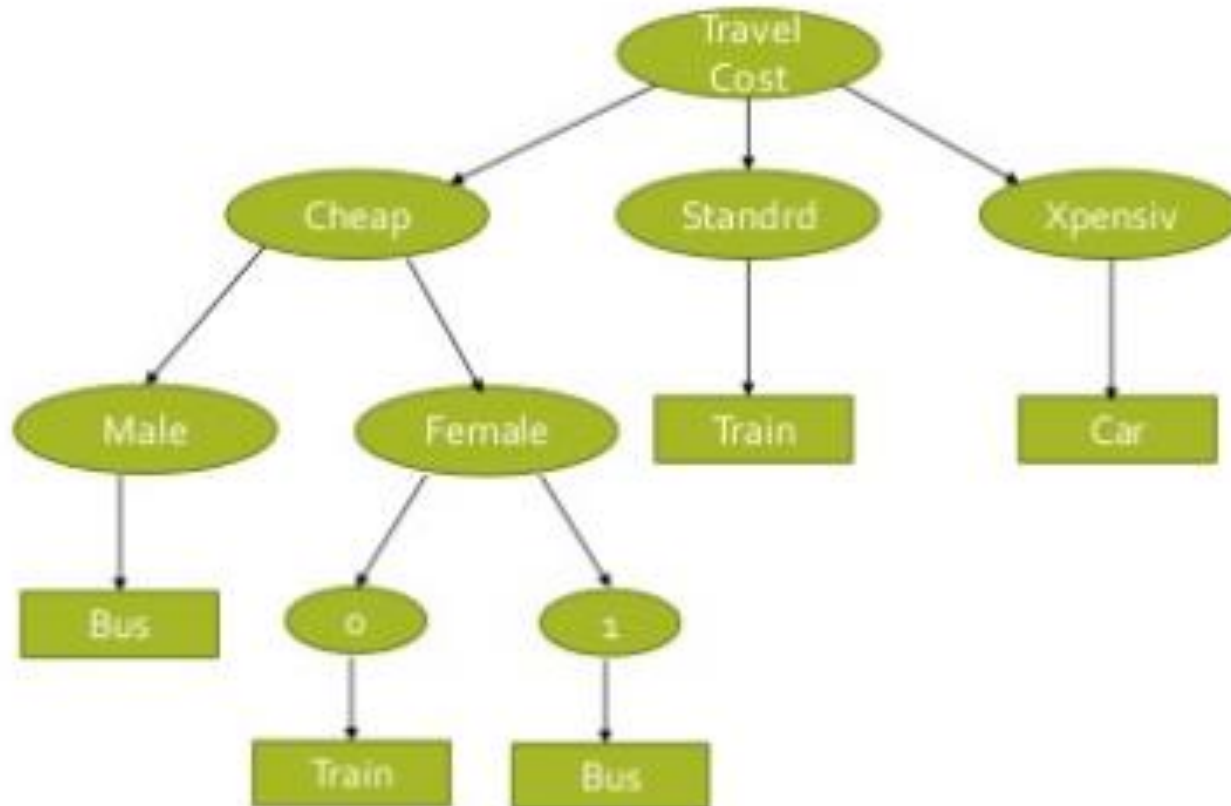
IG = 0.171

Attributes	Information Gain
Gender	0.322
Car Ownership	0.171
Income Level	0.171

Attributes			Classes
Gender	Car Ownership	Income Level	Transportation
Male	0	Low	Bus
Male	1	Medium	Bus
Male	1	Medium	Bus

Attributes			Classes
Gender	Car Ownership	Income Level	Transportation
Female	1	Medium	Train
Female	0	Low	Bus

Diagram : Decision Tree



Solution to Our Problem :

Name	Gender	Car Ownership	Travel Cost	Income Level	Transportation
Abhi	Male	1	Standard	High	Train
Pavi	Male	0	Cheap	Medium	Bus
Ammu	Female	1	Cheap	High	Bus

Gain Ratio for Attribute Selection (C4.5)

- Information gain measure is biased towards attributes with a large number of values
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- $\text{GainRatio}(A) = \text{Gain}(A)/\text{SplitInfo}(A)$
- Ex.
$$\text{SplitInfo}_{\text{income}}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$
 - $\text{gain_ratio}(\text{income}) = 0.029/1.557 = 0.019$
- The attribute with the maximum gain ratio is selected as the splitting attribute

Gini Index (CART, IBM IntelligentMiner)

- If a data set D contains examples from n classes, gini index, $gini(D)$ is defined as

$$gini(D) = 1 - \sum_{j=1}^n p_j^2$$

where p_j is the relative frequency of class j in D

- If a data set D is split on A into two subsets D_1 and D_2 , the gini index $gini_A(D)$ is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Comparing Attribute Selection Measures

- The three measures, in general, return good results but
 - **Information gain:**
 - biased towards multivalued attributes
 - **Gain ratio:**
 - tends to prefer unbalanced splits in which one partition is much smaller than the others
 - **Gini index:**
 - biased to multivalued attributes
 - has difficulty when # of classes is large
 - tends to favor tests that result in equal-sized partitions and purity in both partitions

Overfitting in Machine Learning

Before diving further let's understand two important terms:

- **Bias:** Assumptions made by a model to make a function easier to learn.
- **Variance:** If you train your data on training data and obtain a very low error, upon changing the data and then training the same previous model you experience a high error, this is variance.

A statistical model is said to be overfitted if it can't generalize well with unseen data.

Overfitting:

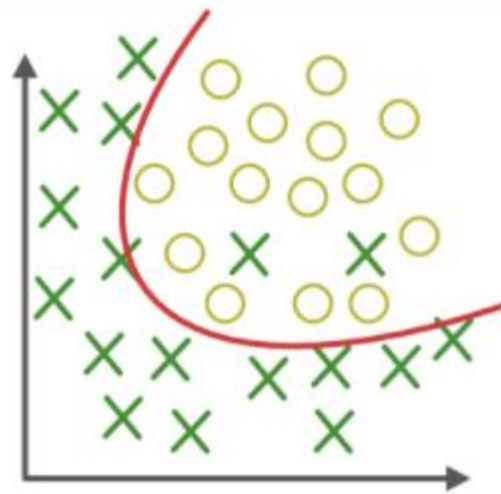
A statistical model is said to be overfitted when we train it with a lot of data (*just like fitting ourselves in oversized pants!*).

When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. Then the model does not categorize the data correctly, because of too many details and noise.

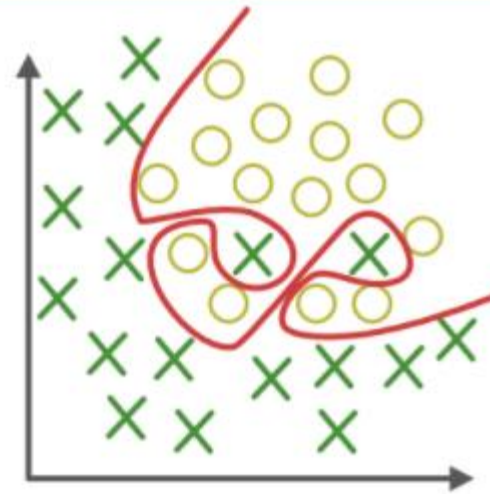
The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.

A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

In a nutshell, **Overfitting – High variance and low bias**



Appropriate-fitting



Over-fitting

(forcefitting--too good to be true)



Reasons for overfitting:

1. Noisy Data.
2. Training set is very large.
3. Large number of features

Methods to avoid overfittings:

1. Limit the number of hidden nodes.
2. Stop training early to avoid a perfect explanation of training set and,
3. Apply weight decay to limit the size of the weights and thus of the function class implemented by the network.

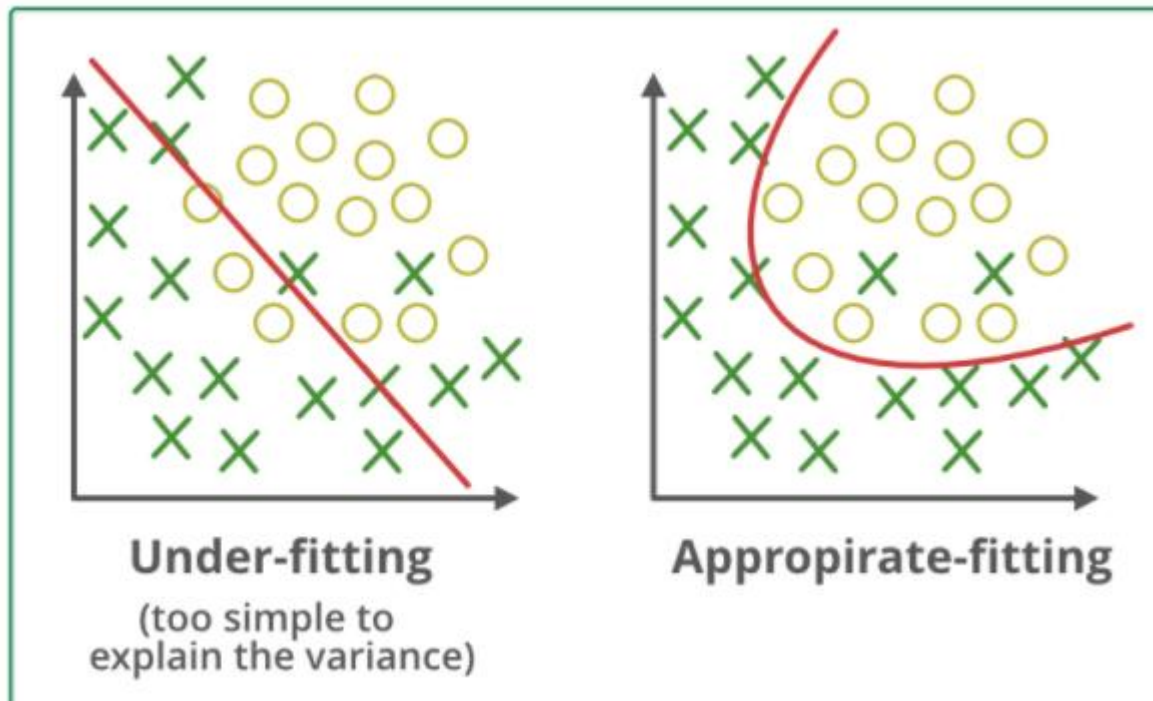
Underfitting:

A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data. *(It's just like trying to fit undersized pants!)* Underfitting destroys the accuracy of our machine learning model.

Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have fewer data to build an accurate model and also when we try to build a linear model with fewer non-linear data.

In such cases, the rules of the machine learning model are too easy and flexible to be applied on such minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection.

In a nutshell, **Underfitting – High bias and low variance**

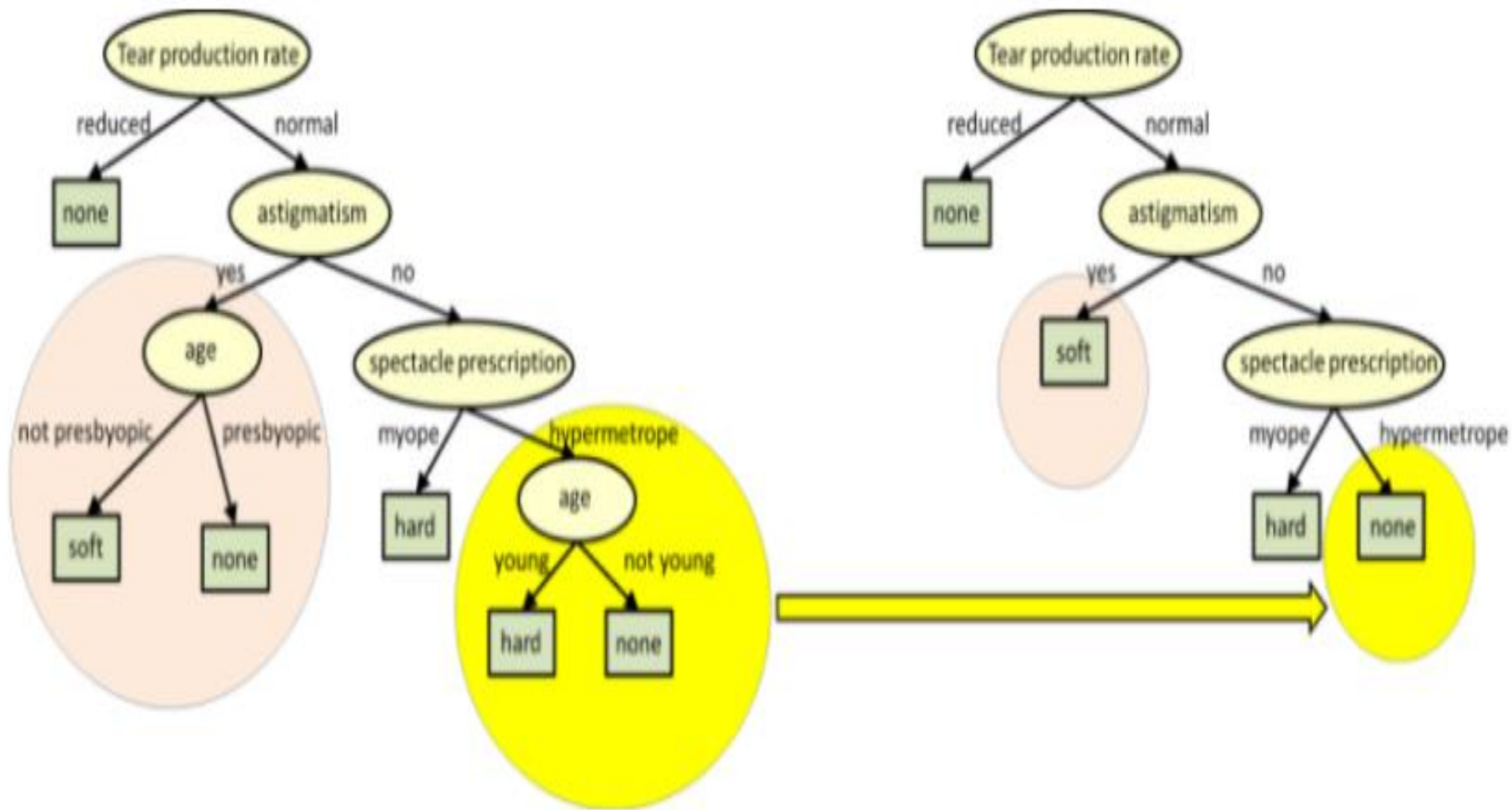


Overfitting and Tree Pruning

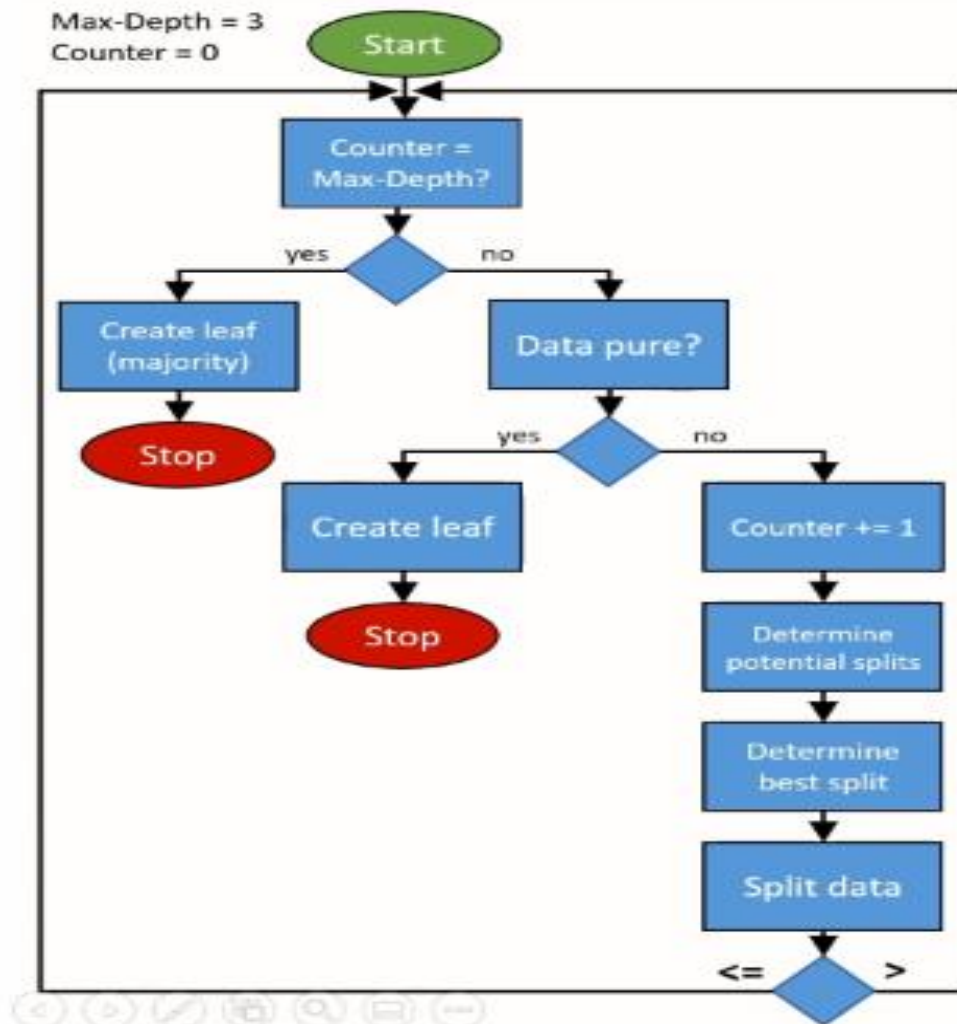
- Overfitting: An induced tree may overfit the training data
 - Too many branches, some may reflect anomalies due to noise or outliers
 - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
 - Prepruning: *Halt tree construction early*-do not split a node if this would result in the goodness measure falling below a threshold
 - Difficult to choose an appropriate threshold
 - Postpruning: *Remove branches* from a “fully grown” tree—get a sequence of progressively pruned trees
 - Use a set of data different from the training data to decide which is the “best pruned tree”

Pruning

- A tree that has too many branches and layers can result in overfitting of the training data.
- **Pruning** a decision tree helps to prevent overfitting the training data so that our model generalizes well to unseen data.
- Pruning a decision tree means to remove a subtree that is redundant and not a useful split and replace it with a leaf node. Decision tree pruning can be divided into two types: *pre-pruning* and *post-pruning*.



Pre-pruning



Pre-Pruning: Based on limiting the depth of the tree

-
- Pre-pruning, also known as Early Stopping Rule, is the method where the subtree construction is halted at a particular node after evaluation of some measure.
 - These measures can be the Gini Impurity or the Information Gain. In pre-pruning, we evaluate the pruning condition based on the above measures at each node.
 - Examples of pruning conditions include $\text{informationGain}(\text{Attr}) > \text{minGain}$ or $\text{treeDepth} == \text{MaxDepth}$.
 - If the condition is satisfied, we prune the subtree. That means we replace the decision node with a leaf node. Otherwise, we continue building the tree using our decision tree algorithm.
 - Pre-pruning has the advantage of being faster and more efficient as it avoids generating overly complex subtrees which overfit the training data. However, in pre-pruning, the growth of the tree is stopped prematurely by our stopping condition.

Post-pruning

As the name suggests, **post-pruning** means to prune after the tree is built. You grow the tree entirely using your decision tree algorithm and then you prune the subtrees in the tree in a bottom-up fashion.

You start from the bottom decision node and, based on measures such as Gini Impurity or Information Gain, you decide whether to keep this decision node or replace it with a leaf node.

For example, say we want to prune out subtrees that result in least information gain. When deciding the leaf node, we want to know what leaf our decision tree algorithm would have created if it didn't create this decision node.

Pruning algorithms

There are many pruning algorithms out there; below are three examples of pruning algorithms.

Pruning by information gain

We can prune our decision tree by using **information gain** in both post-pruning and pre-pruning. In pre-pruning, we check whether information gain at a particular node is greater than minimum gain. In post-pruning, we prune the subtrees with the least information gain until we reach a desired number of leaves.

Reduced Error Pruning (REP)

REP belongs to the Post-Pruning category. In **REP**, pruning is performed with the help of a [validation set](#). In REP, all nodes are evaluated for pruning in a bottom up fashion. A node is pruned if the resulting pruned tree performs no worse than the original tree on the validation set. The subtree at the node is replaced with a leaf node which is assigned the most common class.

Cost-complexity pruning

Cost-complexity pruning also falls under the post-pruning category. **Cost-complexity pruning** works by calculating a *Tree Score* based on Residual Sum of Squares (RSS) for the subtree, and a *Tree Complexity Penalty* that is a function of the number of leaves in the subtree.

The Tree Complexity Penalty compensates for the difference in the number of leaves. Numerically, Tree Score is defined as follows:

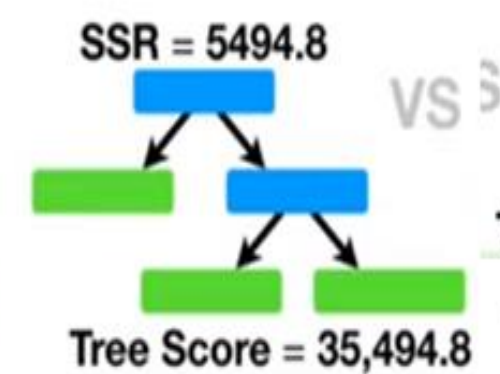
$$\text{Tree Score} = \text{SSR} + \alpha T$$

Take $\alpha = 10,000$

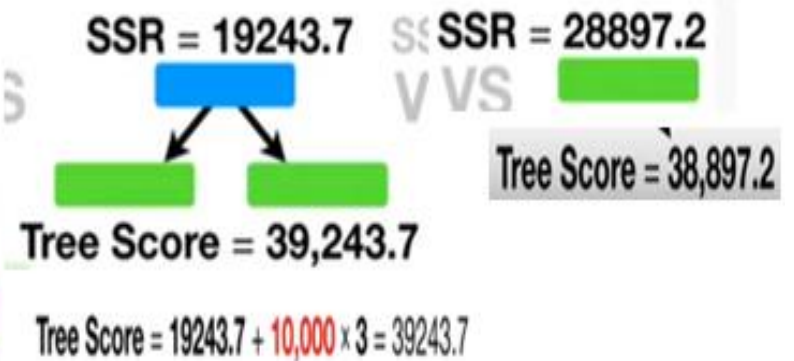
Then we will get



$$\text{Tree Score} = 543.8 + 10,000 \times 4 = 40,543.8$$



$$\text{Tree Score} = 5494.8 + 10,000 \times 3 = 35,494.8$$



$$\text{Tree Score} = 19243.7 + 10,000 \times 3 = 39,243.7$$

$$\text{Tree Score} = 38,897.2$$

Why to go for Regression Trees ?

Take an example where we are plotting data points of Drug Dosage and Drug effectiveness.

If points are falling like this (as shown in fig. 1) then we can go for Simple linear regression

Drug effectiveness

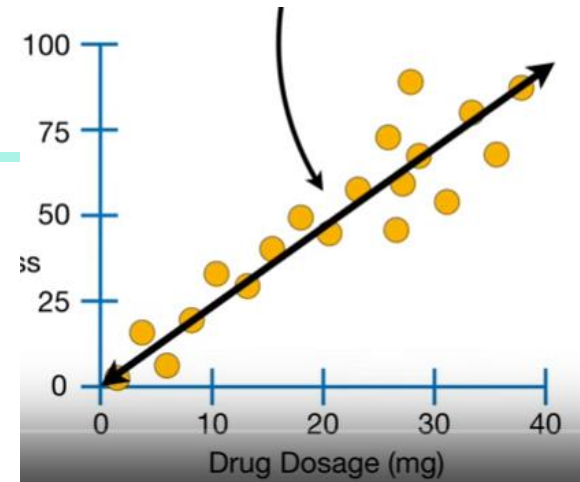


fig. 1

If data points are like as shown in fig.2 in such cases we can-not go for linear regression

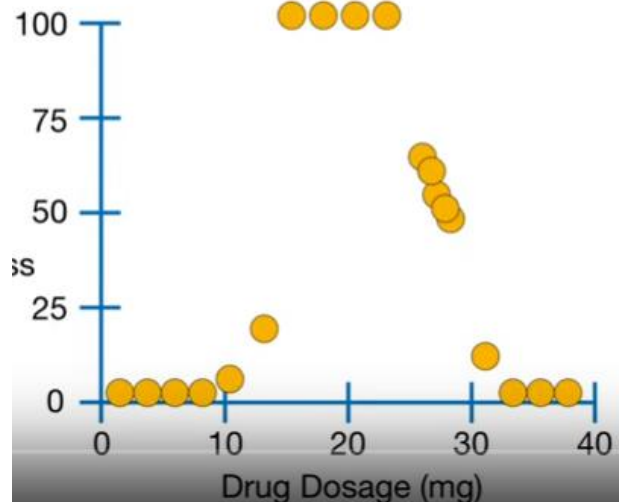


fig.2

We cannot draw a line that best fits all the points. In scenarios like this we go for Regression Tree. That looks like fig.3

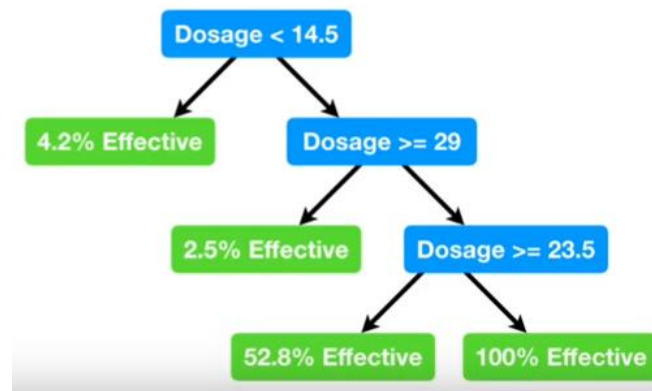
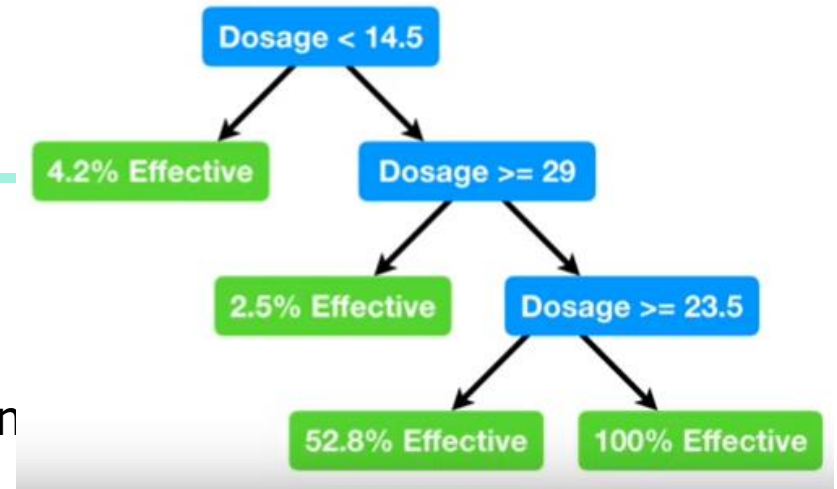


fig.3

Regression Trees

Regression Trees are a type of decision trees where each leaf node represent a numeric

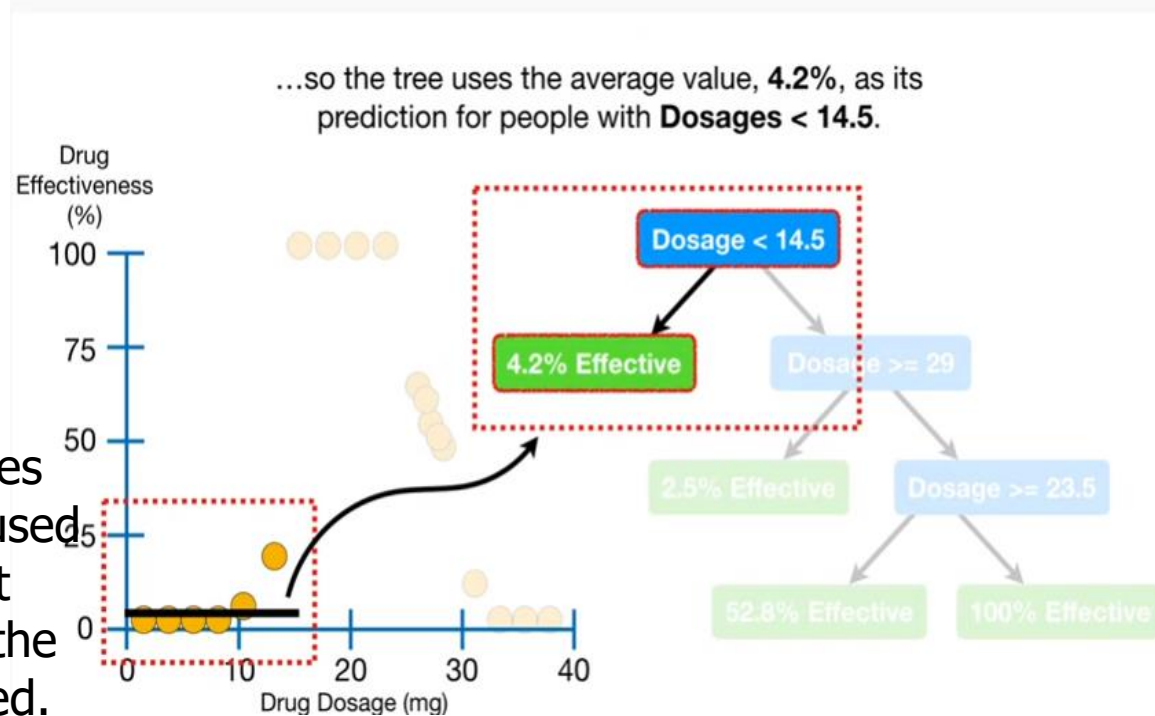
Values in contrast to decision/classification trees which uses True or False in the leaf nodes or Some other discrete category.



In this example the points highlighted

on X-axis are less then 14.5 and their average value on Y axis is 4.2

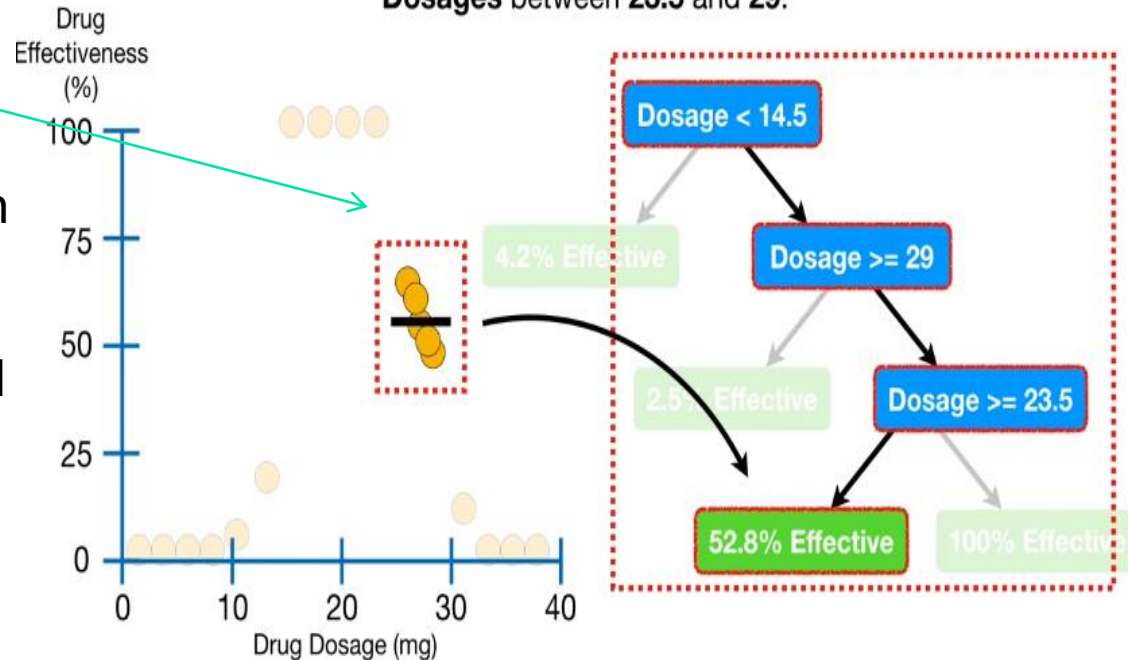
Note: I'll be using lot of pictures here. Don't worry as they are used To explain things. You need not Draw all these in exams. Only the Important diagrams are required.



In the same ways the average Value of the these data points greater than 23.5 points on X-axis

And there average value all these Data points on Y-axis is 52.8

...so the tree uses the average value, **52.8%**, as its prediction for people with **Dosages** between **23.5** and **29**.



Here each leaf node corresponds to average Drug effectiveness

Enhancements to Basic Decision Tree Induction

- Allow for **continuous-valued attributes**
 - Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals
- Handle **missing attribute values**
 - Assign the most common value of the attribute
 - Assign probability to each of the possible values
- **Attribute construction**
 - Create new attributes based on existing ones that are sparsely represented
 - This reduces fragmentation, repetition, and replication

Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class \ Predicted class	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

Example of Confusion Matrix:

Actual class \ Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

- Given m classes, an entry, $CM_{i,j}$ in a **confusion matrix** indicates # of tuples in class i that were labeled by the classifier as class j
- May have extra rows/columns to provide totals

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	-C	
C	TP	FN	P
-C	FP	TN	N
	P'	N'	All

- **Classifier Accuracy**, or recognition rate: percentage of test set tuples that are correctly classified

$$\text{Accuracy} = (TP + TN)/All$$

- **Error rate**: $1 - \text{accuracy}$, or
Error rate = $(FP + FN)/All$

- **Class Imbalance Problem:**

- One class may be *rare*, e.g. fraud, or HIV-positive
- Significant *majority of the negative class* and minority of the positive class
- **Sensitivity**: True Positive recognition rate
 - **Sensitivity** = TP/P
- **Specificity**: True Negative recognition rate
 - **Specificity** = TN/N

Classifier Evaluation Metrics:

Precision and Recall, and F-measures

- **Precision:** exactness – what % of tuples that the classifier labeled as positive are actually positive

$$\textit{precision} = \frac{TP}{TP + FP}$$

- **Recall:** completeness – what % of positive tuples did the classifier label as positive?

$$\textit{recall} = \frac{TP}{TP + FN}$$

- Perfect score is 1.0
- Inverse relationship between precision & recall
- **F measure (F_1 or F-score):** harmonic mean of precision and recall,

$$F = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$$

- F_β : weighted measure of precision and recall
 - assigns β times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times \textit{precision} \times \textit{recall}}{\beta^2 \times \textit{precision} + \textit{recall}}$$

Classifier Evaluation Metrics: Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	90	210	300	30.00 (<i>sensitivity</i>)
cancer = no	140	9560	9700	98.56 (<i>specificity</i>)
Total	230	9770	10000	96.40 (<i>accuracy</i>)

■ $Precision = 90/230 = 39.13\%$

$Recall = 90/300 = 30.00\%$

Time taken to build model: 0.73 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	705	70.5	%
Incorrectly Classified Instances	295	29.5	%
Kappa statistic	0.2467		
Mean absolute error	0.3467		
Root mean squared error	0.4796		
Relative absolute error	82.5233	%	
Root relative squared error	104.6565	%	
Total Number of Instances	1000		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
	0.840	0.610	0.763	0.840	0.799	0.251	0.639	0.746
	0.390	0.160	0.511	0.390	0.442	0.251	0.639	0.449
Weighted Avg.	0.705	0.475	0.687	0.705	0.692	0.251	0.639	0.657

=== Confusion Matrix ===

```
a  b  <-- classified as
588 112 |  a = good
183 117 |  b = bad
```

Evaluating Classifier Accuracy: Holdout & Cross-Validation Methods

■ Holdout method

- Given data is randomly partitioned into two independent sets
 - Training set (e.g., 2/3) for model construction
 - Test set (e.g., 1/3) for accuracy estimation
- Random sampling: a variation of holdout
 - Repeat holdout k times, accuracy = avg. of the accuracies obtained

■ Cross-validation (k -fold, where $k = 10$ is most popular)

- Randomly partition the data into k *mutually exclusive* subsets, each approximately equal size
- At i -th iteration, use D_i as test set and others as training set
- Leave-one-out: k folds where $k = \#$ of tuples, for small sized data
- *Stratified cross-validation*: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

Measuring Time series Analysis

Measuring Time Series Forecasting Performance

- The fact that the future is wholly unknown and can only be predicted from what has already occurred is a significant distinction in forecasting. The ability of a time series forecasting model to predict the future is defined by its performance
- Time series prediction performance measurements provide a summary of the forecast model's skill and capability in making the forecasts.
- There are numerous performance metrics from which to pick. Knowing which metric to use and how to interpret the data might be difficult. .

Evaluation Metrics to Measure Performance

Now, let us have a look at the popular evaluation metrics used to measure the performance of a time-series forecasting model.

R-Squared

The stationary R-squared is used in time series forecasting as a measure that compares the stationary part of the model to a simple mean model. It is defined as,

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

Where,

SS_{res} denotes the sum of squared residuals from expected values .

SS_{tot} denotes the sum of squared deviations from the dependent variable's sample mean.

It denotes the proportion of the dependent variable's variance that may be explained by the independent variable's variance. A high R^2 value shows that the model's variance is similar to that of the true values, whereas a low R^2 value suggests that the two values are not strongly related.

Mean Absolute Error (MAE) :

The MAE is defined as the average of the absolute difference between forecasted and true values. Where y_i is the expected value and x_i is the actual value (shown below formula). The letter n represents the total number of values in the test set.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

The MAE shows us how much inaccuracy we should expect from the forecast on average.

MAE = 0 means that the anticipated values are correct, and the error statistics are in the original units of the forecasted values.

The lower the MAE value, the better the model; a value of zero indicates that the forecast is error-free. In other words, the model with the lowest MAE is deemed superior when comparing many models.

However, because MAE does not reveal the proportional scale of the error, it can be difficult to distinguish between large and little errors.

It can be combined with other measures to see if the errors are higher (see Root Mean Square Error below). Furthermore, MAE might obscure issues related to low data volume; for more information, check the last two metrics in this article.

Mean Absolute Percentage Error (MAPE)

MAPE is the proportion of the average absolute difference between projected and true values divided by the true value. The anticipated value is F_t , and the true value is A_t . The number n refers to the total number of values in the test set.

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

It works better with data that is free of zeros and extreme values because of the in-denominator. The MAPE value also takes an extreme value if this value is exceedingly tiny or huge.

The model is better if the MAPE is low. Remember that MAPE works best with data that is devoid of zeros and extreme values. MAPE, like MAE, understates the impact of big but rare errors caused by extreme values.

Mean Square Error can be utilized to address this issue. This statistic may obscure issues related to low data volume; for more information, check the last two metrics in this article.

Root Mean Squared Error(RMSE)

This measure is defined as the square root of mean square error and is an extension of MSE. Where y' denotes the predicted value and y denotes the actual value. The number n refers to the total number of values in the test set. This statistic, like MSE, penalizes greater errors more.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$$

This statistic is likewise always positive, with lower values indicating higher performance. The RMSE number is in the same unit as the projected value, which is an advantage of this technique. In comparison to MSE, this makes it easier to comprehend.

The RMSE can also be compared to the MAE to see whether there are any substantial but uncommon inaccuracies in the forecast. The wider the gap between RMSE and MAE, the more erratic the error size. This statistic can mask issues with low data volume.

Normalized Root Mean Squared Error (NRMSE)

The normalized RMSE is used to calculate NRMSE, which is an extension of RMSE. The mean or the range of actual values are the two most used methods for standardizing RMSE (difference of minimum and maximum values). The maximum true value is y_{max} , while the smallest true value is y_{min} .

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}} \text{ or } \text{NRMSE} = \frac{\text{RMSE}}{\bar{y}}$$

NRMSE is frequently used to compare datasets or forecasting models with varying sizes (units and gross revenue, for example).

The smaller the value, the better the model's performance. When working with little amounts of data, this metric can be misleading.

However, Weighted Absolute Percentage Error and Weighted Mean Absolute Percentage Error can help.

Weighted Mean Absolute Percentage Error (WMAPE)

WMAPE (sometimes called wMAPE) is an abbreviation for Weighted Mean Absolute Percentage Error. It is a measure of a forecasting method's prediction accuracy. It is a MAPE version in which errors are weighted by real values (e.g. in the case of sales forecasting, errors are weighted by sales volume).

$$\text{WMAPE} = \frac{\sum_{t=1}^n |A_t - F_t|}{\sum_{t=1}^n |A_t|}$$

where A is the current data vector and F is the forecast This metric has an advantage over MAPE in that it avoids the 'infinite error' problem.

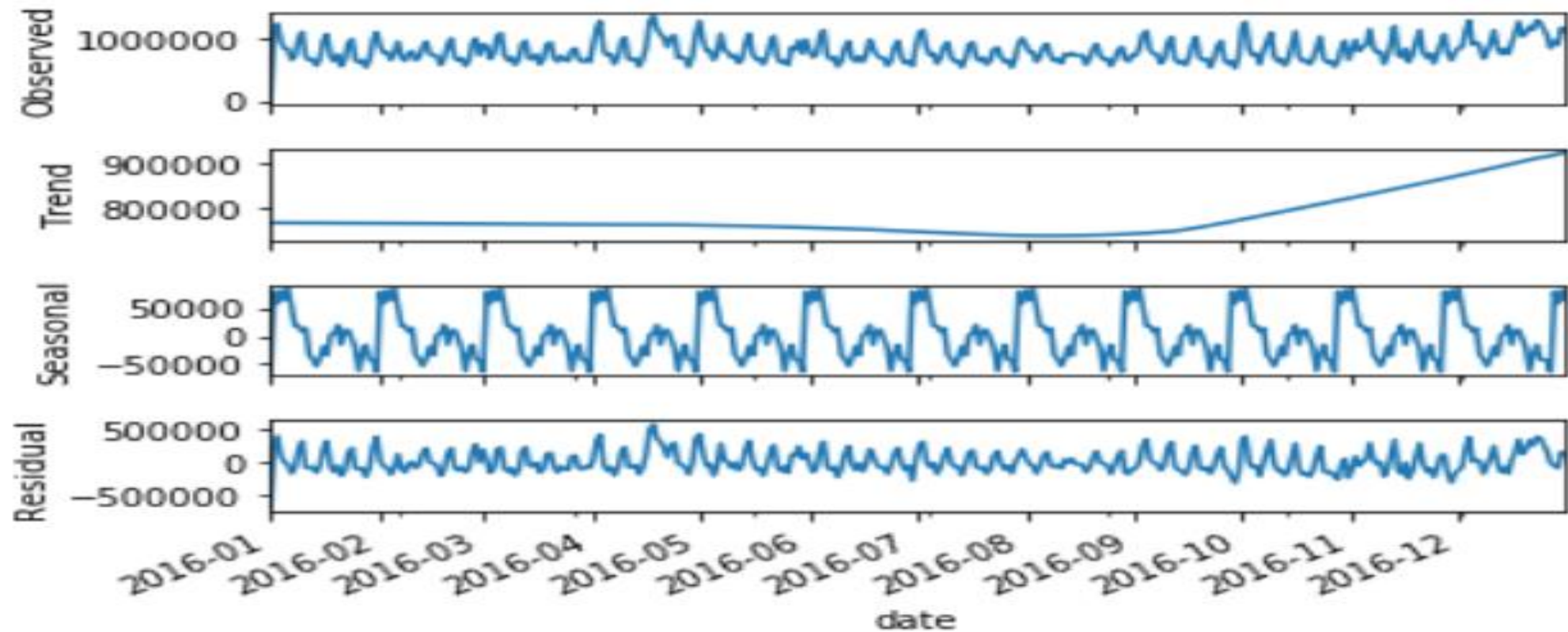
The higher the model's performance, the lower the WMAPE number. When evaluating forecasting models, this metric is useful for low volume data where each observation has a varied priority.

The weight value of observations with a higher priority is higher. The WMAPE number increases as the error in high-priority forecast values grows.

STL Approach

What is STL decomposition?

So, STL stands for **Seasonal and Trend decomposition using Loess**. This is a statistical method of decomposing a Time Series data into 3 components containing seasonality, trend and residual.



Now let's talk about trend.

Trend gives you a general direction of the overall data. From the above example, I can say that from 9:00am to 11:00am there is a downward trend and from 11:00am to 1:00pm there is an upward trend and after 1:00pm the trend is constant.

Tata Motors Limited Fully Paid Ord. Shrs
NSE: TATAMOTORS

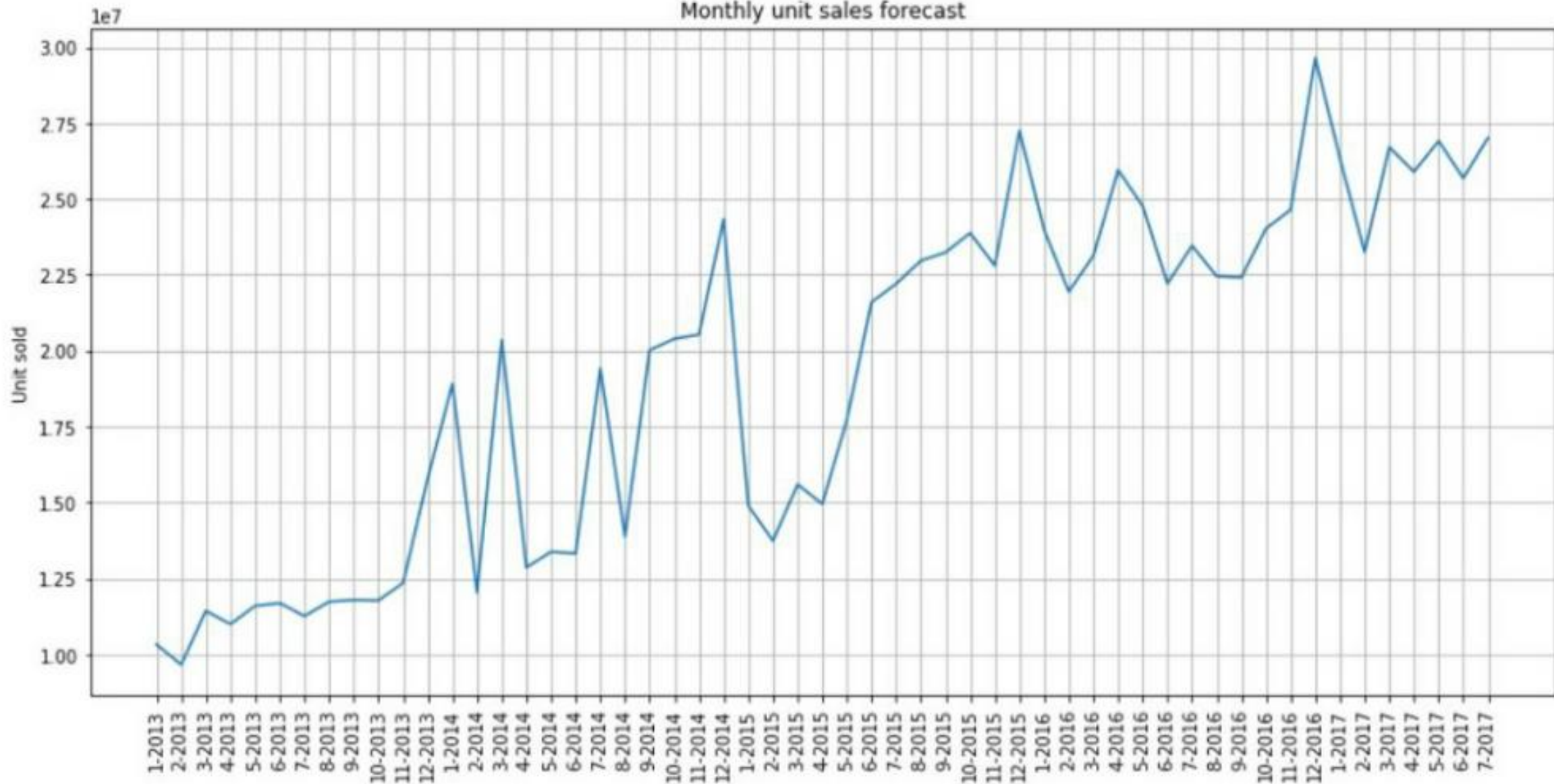
+ Follow

171.35 INR -3.15 (1.81%) ↓

7 Nov, 3:30 pm IST · Disclaimer

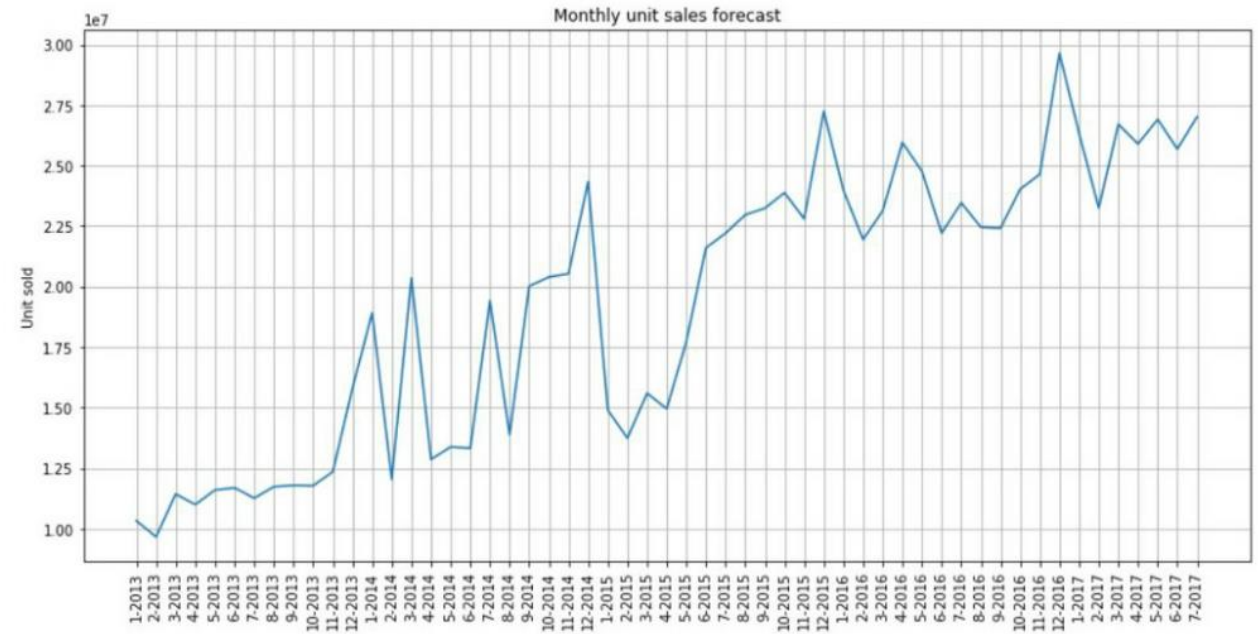


Monthly unit sales forecast



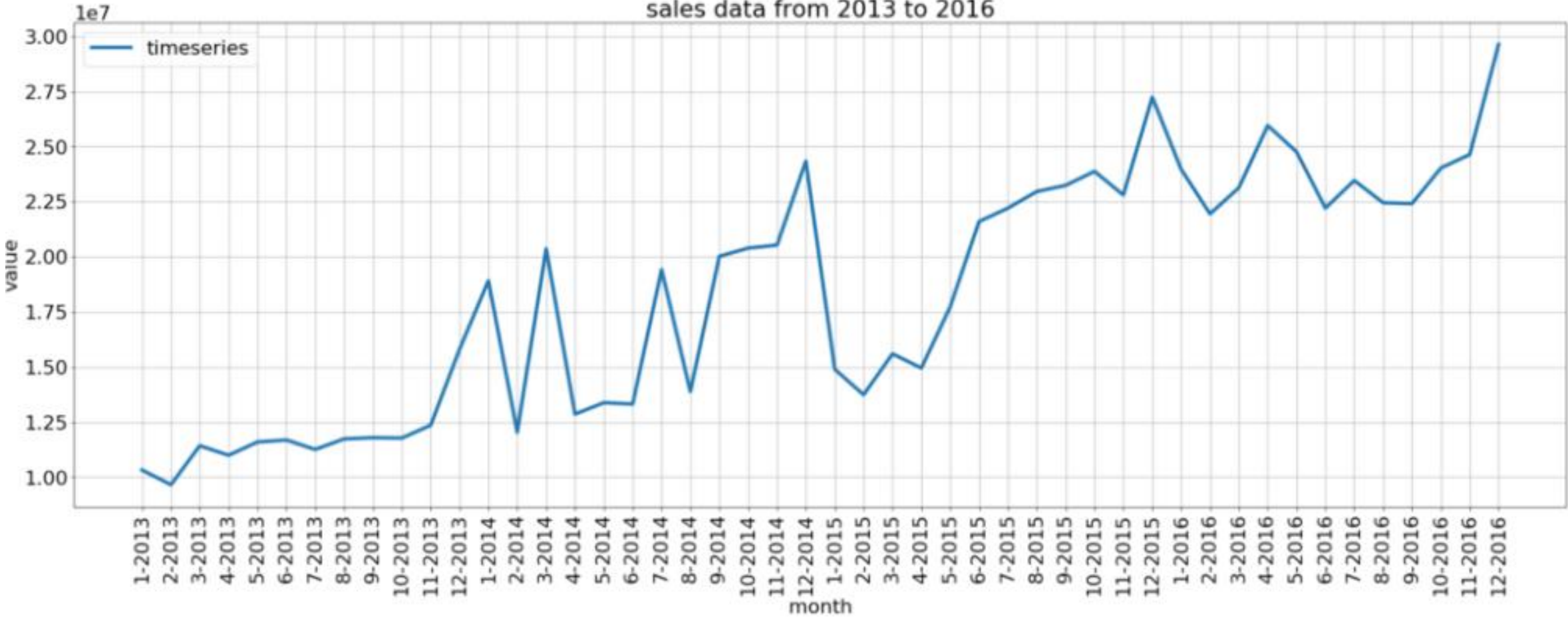
Whereas seasonality is a regular and predictable pattern that recur at a fixed interval of time.

For example, the below plot helps us understand the total units sold per month for a retailer. So, if we try to watch carefully, we can see an increment in the unit sales during the month of December every year. So, there is a regular pattern or seasonality in the unit sales associated with a period of 12 months.



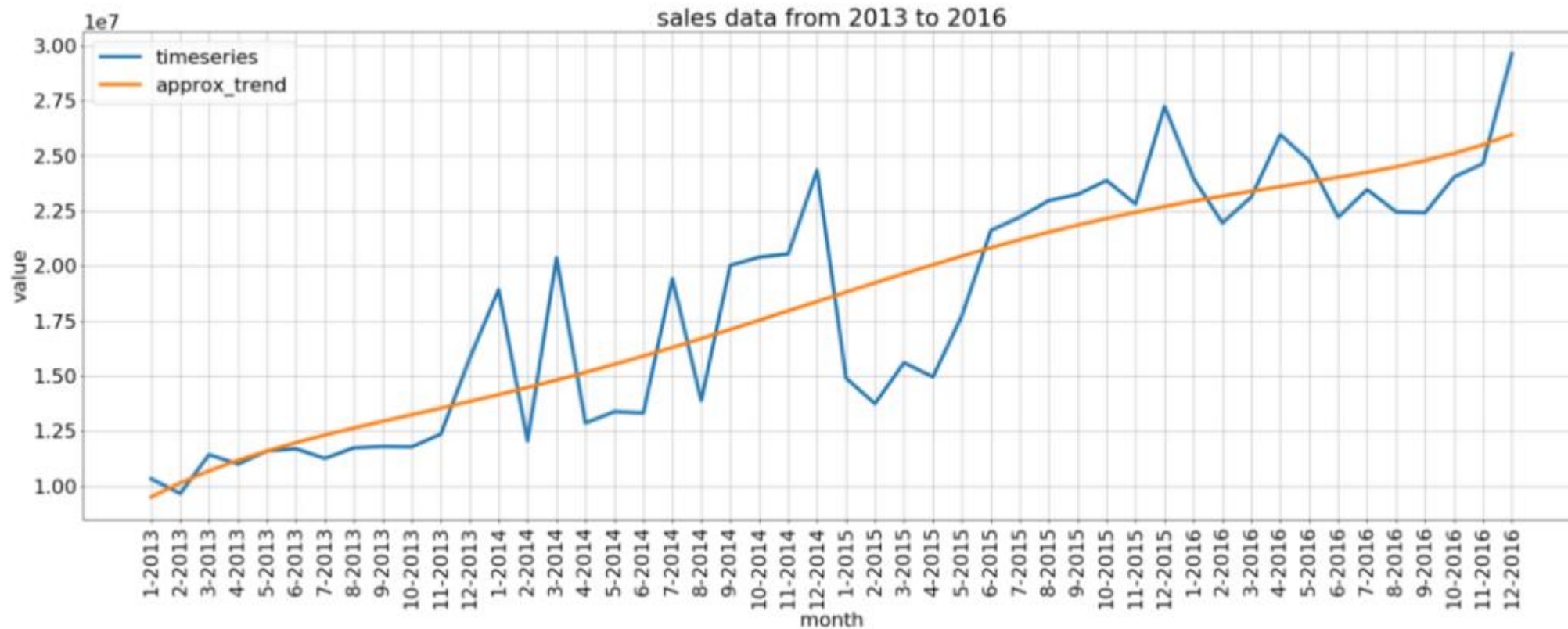
Now, let's talk about **Loess**. So, loess is a regression technique that uses local weighted regression to fit a smooth curve through points in a sequence, which in our case is the Time Series data.

How to do STL decomposition ?



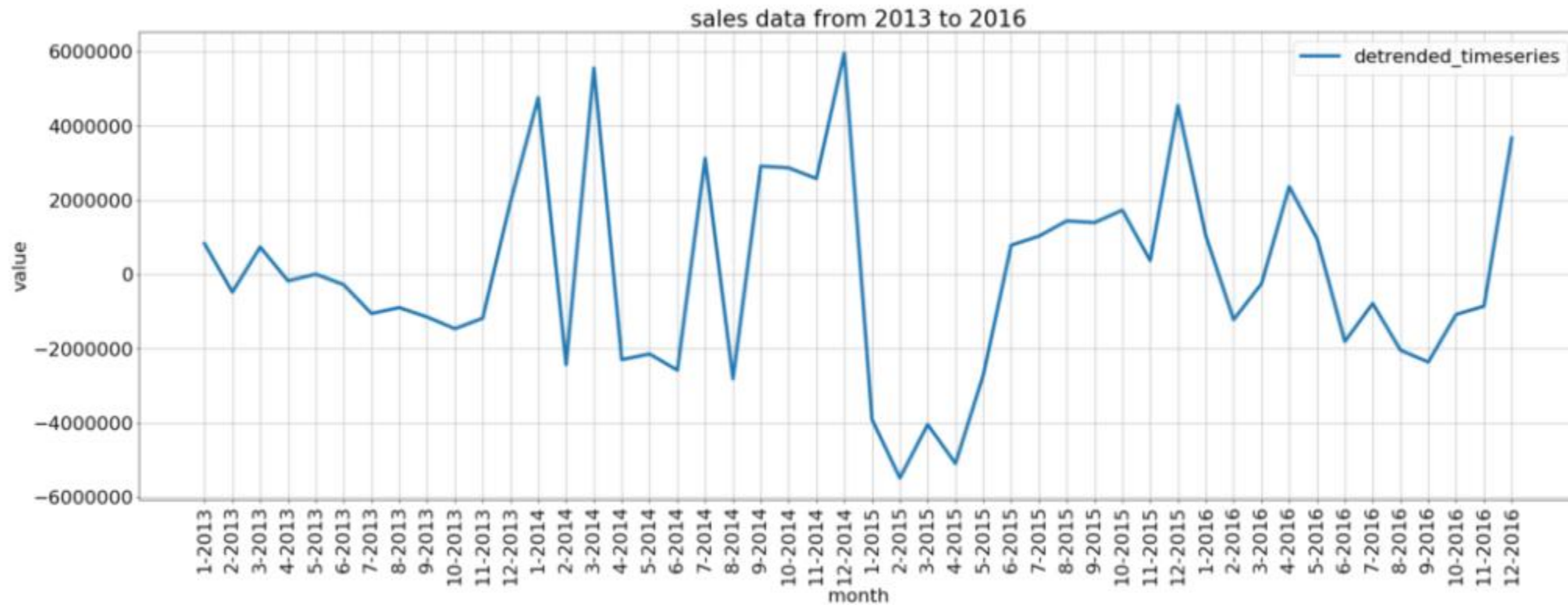
Step 1

Find the Approximate Trend line which fits the above time series data.



Step 2

Find the De-trended series by subtracting time series data with approximate trend line.



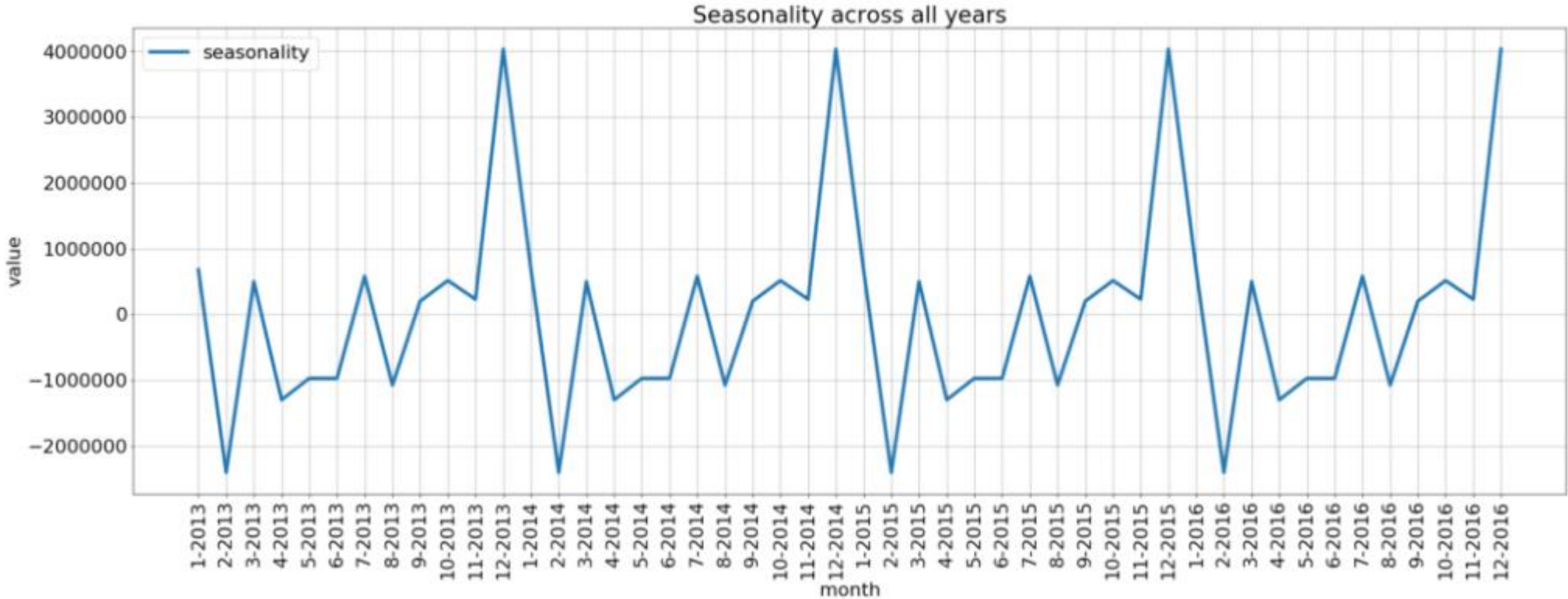
Step 3

Find the Seasonal Component. Here I am finding out seasonality on month level (for a period = 12). So I have to group the De-trended series by month to find the average value of units sold for each month across the whole data (for 48 months).



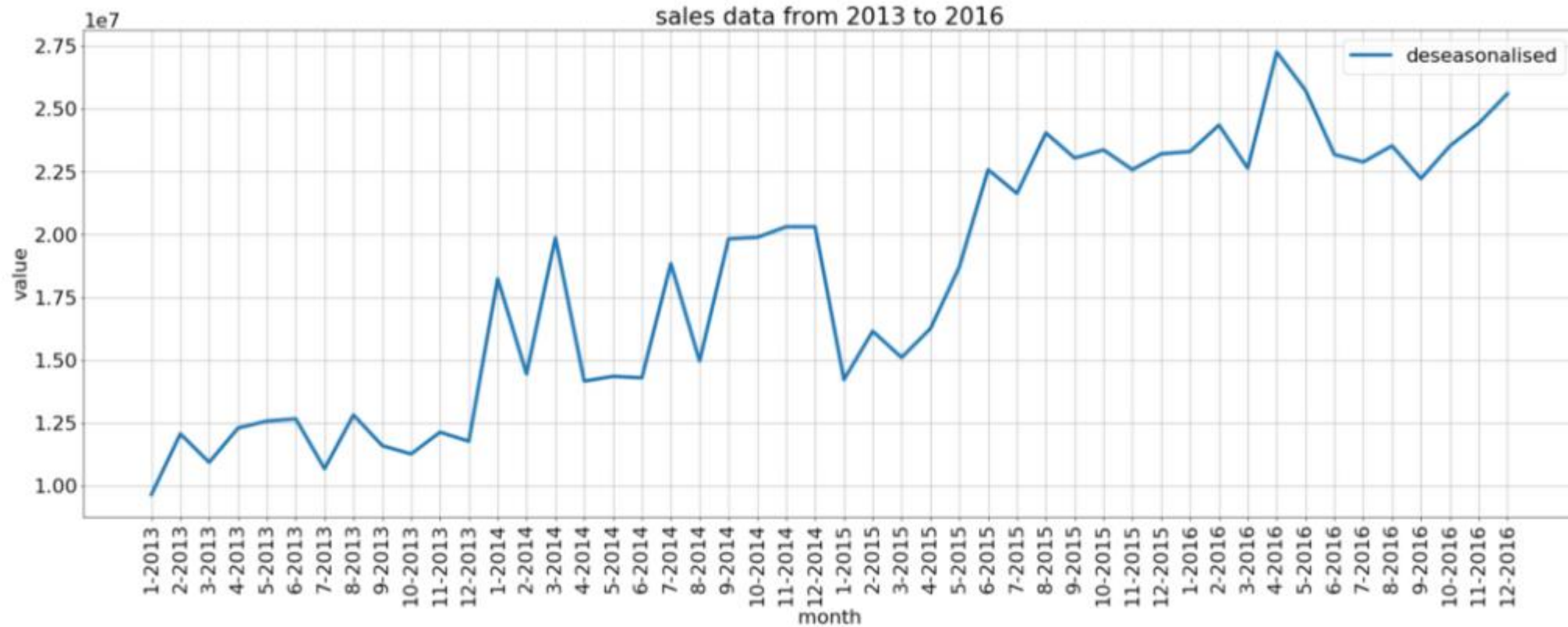
Step 4

Populate this component across the whole data (for 48 months).



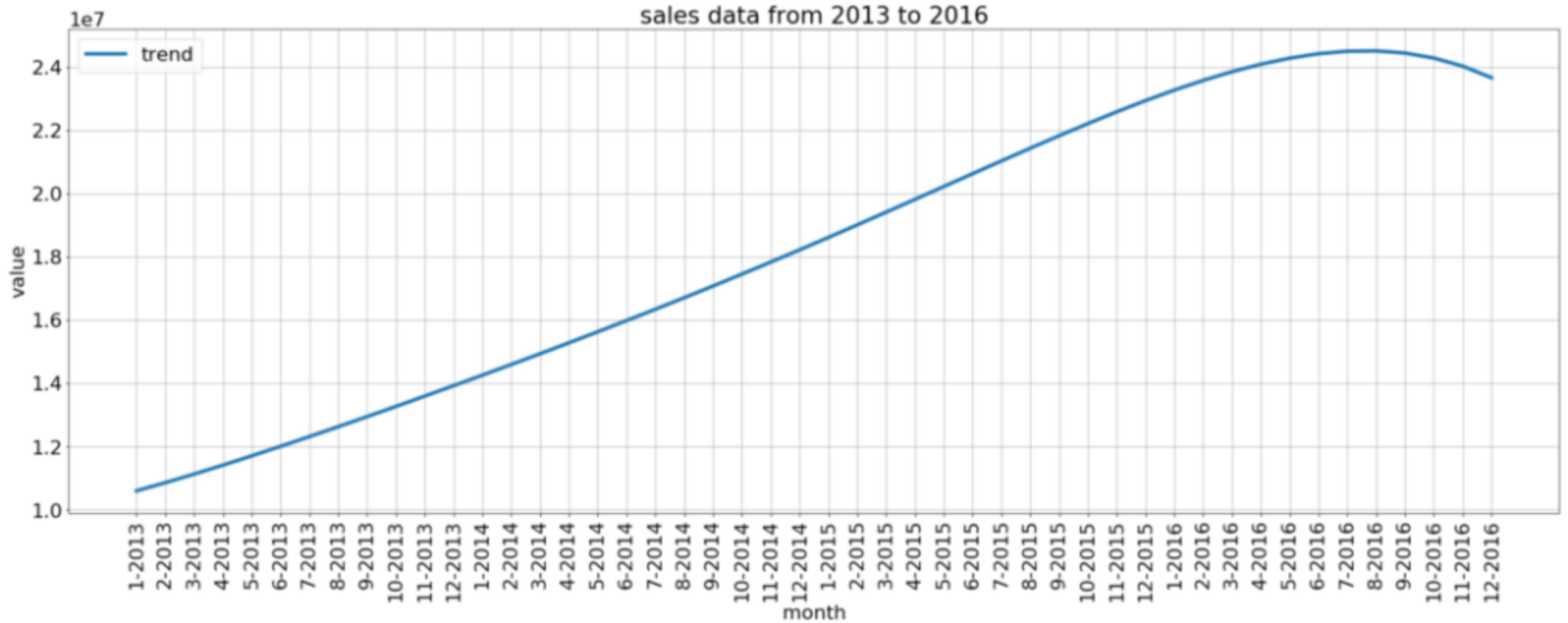
Step 5

Find the De-seasonalised time series by subtracting time series data with seasonal data.



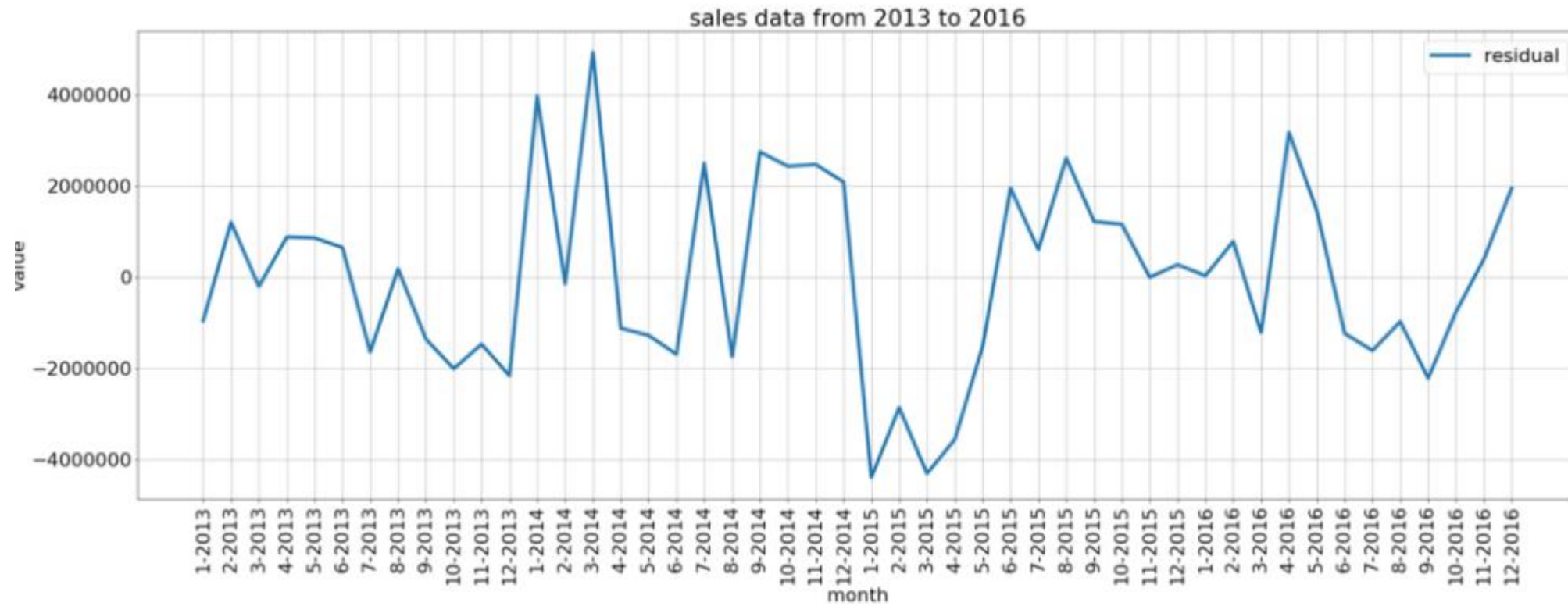
Step 6

Find the trend by fitting the De-seasonalised data to a polynomial model.



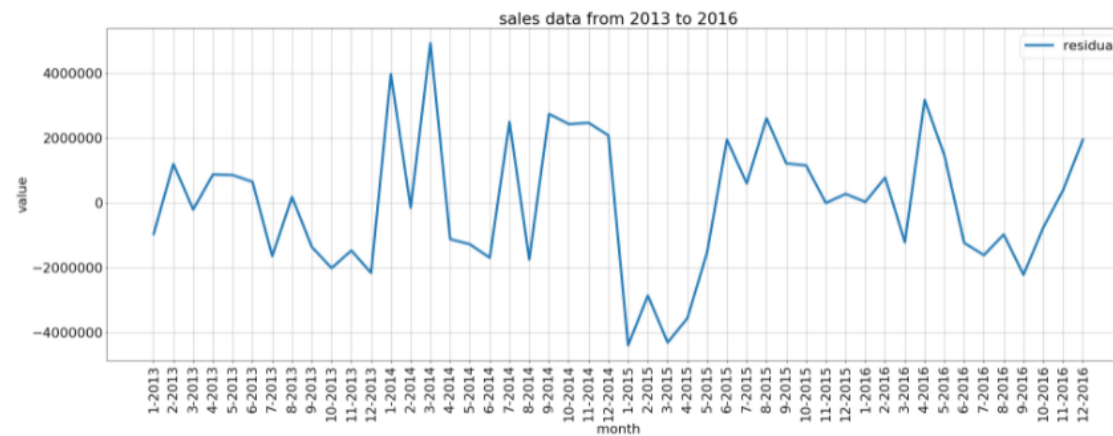
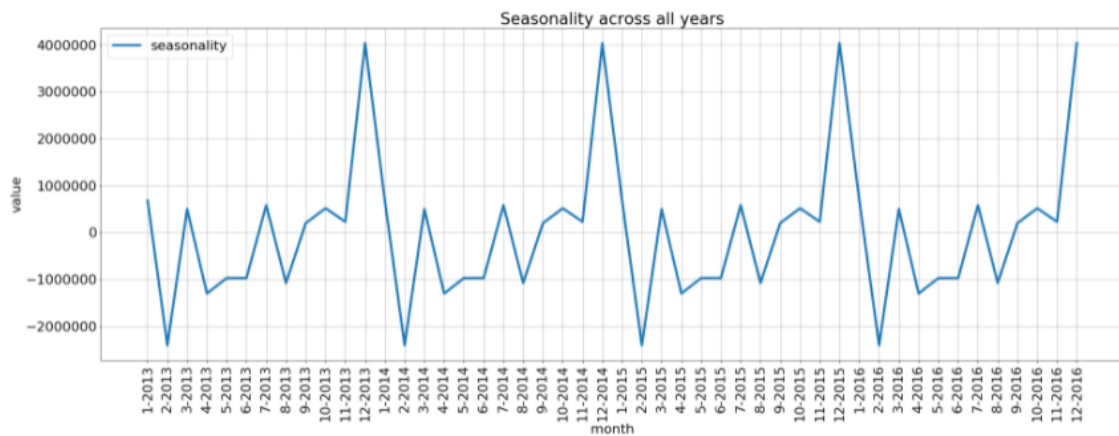
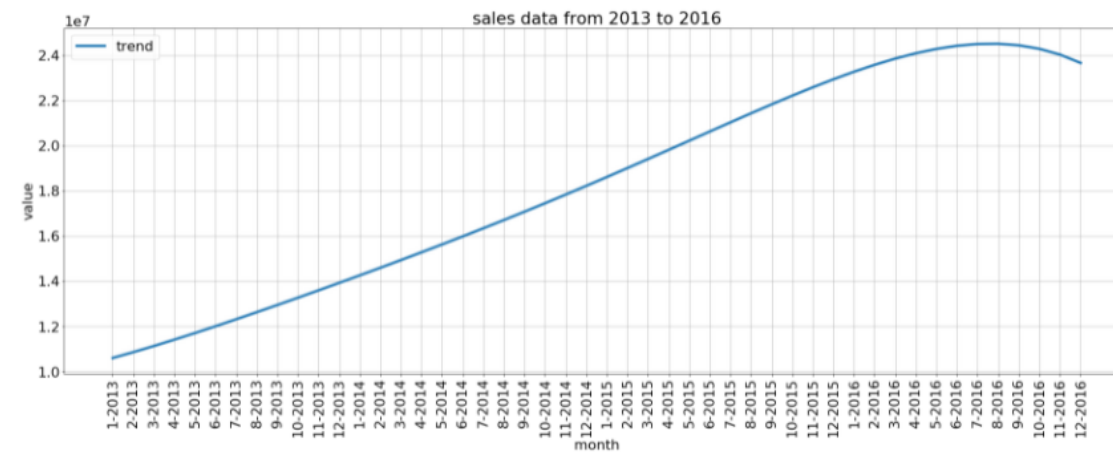
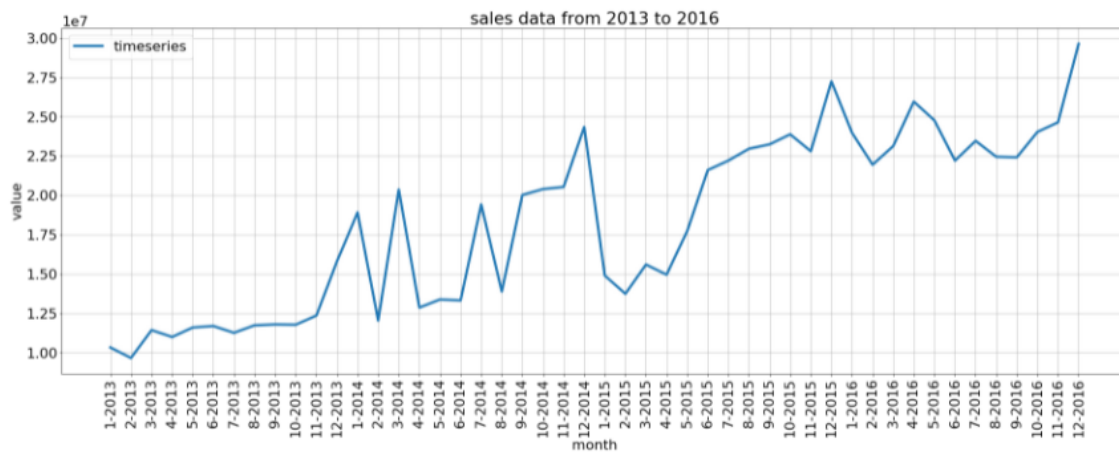
Step 7

Find the residual by subtracting time series data with seasonal and trend.



Decomposed components of Time series

So, here are the final figures of STL decomposition:



STL has several advantages over the classical decomposition method and X-12-ARIMA: Unlike X-12-ARIMA,

STL will handle any type of seasonality, not only monthly and quarterly data. The seasonal component is allowed to change over time, and the rate of change can be controlled by the user.

Time Series Analysis

Why Time Series Analysis?

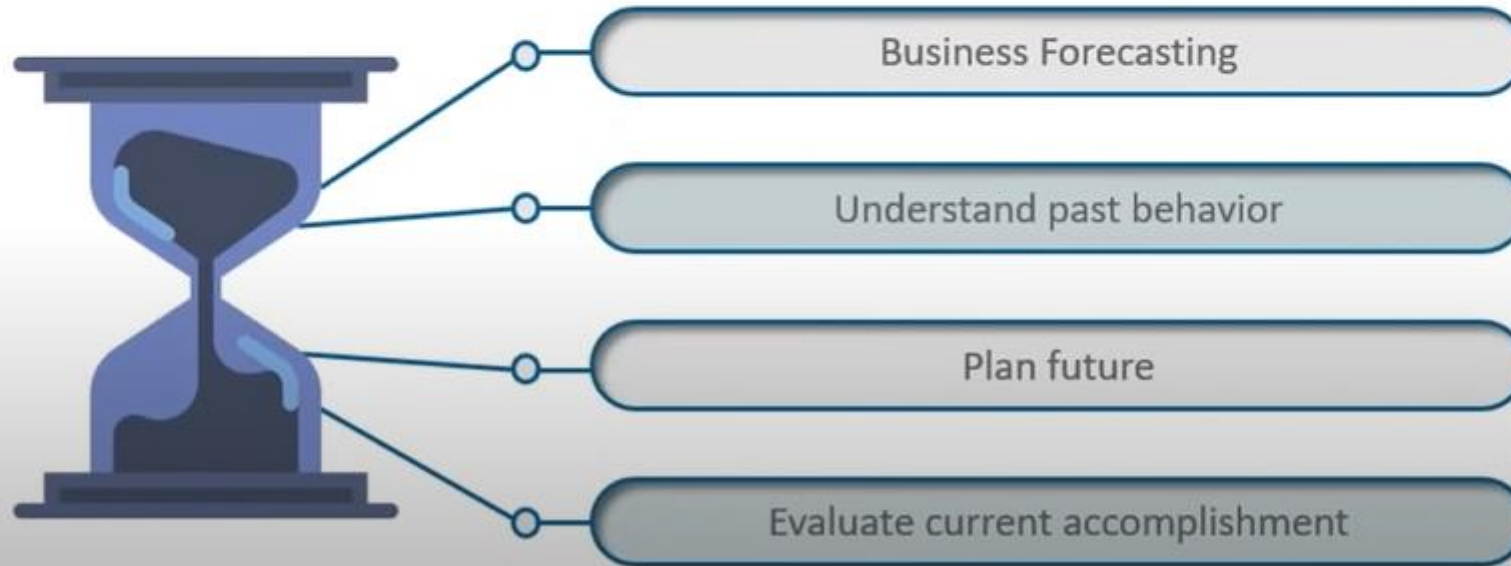
In this analysis, you just have one variable – **TIME**

You can analyse this **time series** data in order to extract meaningful statistics and other characteristics



What Is Time Series?

- A time series is a set of observation taken at specified **times** usually at equal intervals
- It is used to **predict** the future values based on the **previous** observed values



Stationarity

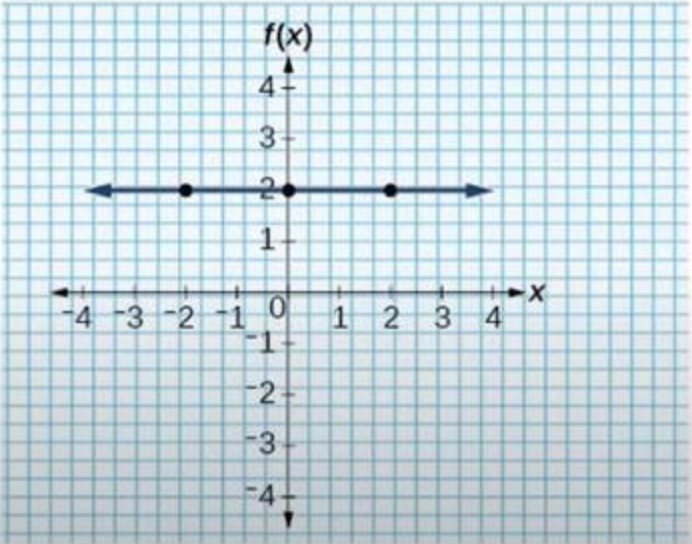
Stationarity is an important characteristic of time series. A time series is said to be stationary if its statistical properties do not change over time. In other words, it has **constant mean and variance**, and covariance is independent of time.

Components Of Time Series

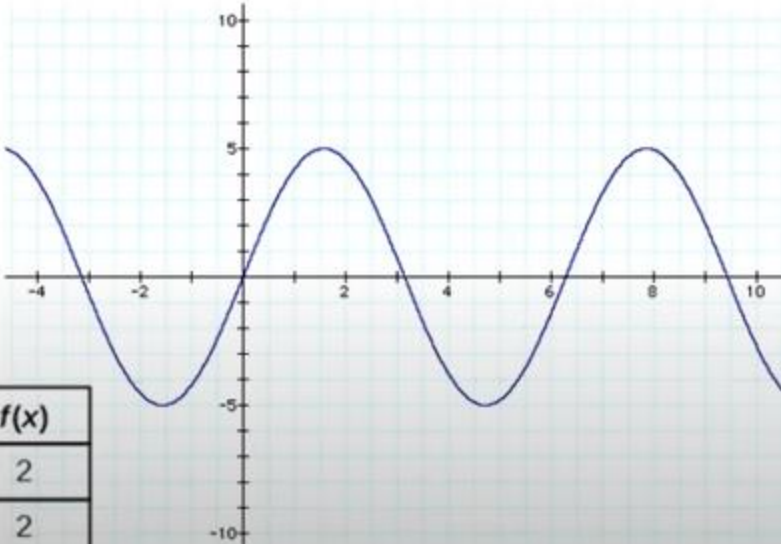


When Not To Use Time Series Analysis?

1 Values are constant



2 Values in the form of functions



x	$f(x)$
-2	2
0	2
2	2

Basics of Time-Series Forecasting

Timeseries forecasting in simple words means to forecast or to predict the future value(eg-stock price) over a period of time. There are different approaches to predict the value, consider an example there is a company XYZ records the website traffic in each hour and now wants to forecast the total traffic of the coming hour. If I ask you what will your approach to forecasting the upcoming hour traffic?

1) Seasonality

Seasonality is a simple term that means while predicting a time series data there are some months in a particular domain where the output value is at a peak as compared to other months. for example if you observe the data of tours and travels companies of past 3 years then you can see that in November and December the distribution will be very high due to holiday season and festival season. So while forecasting time series data we need to capture this seasonality.

2) Trend

The trend is also one of the important factors which describe that there is certainly increasing or decreasing trend time series, which actually means the value of organization or sales over a period of time and seasonality is increasing or decreasing.

3) Unexpected Events

Unexpected events mean some dynamic changes occur in an organization, or in the market which cannot be captured. for example a current pandemic we are suffering from, and if you observe the Sensex or nifty chart there is a huge decrease in stock price which is an unexpected event that occurs in the surrounding.

Methods and algorithms are using which we can capture seasonality and trend But the unexpected event occurs dynamically so capturing this becomes very difficult.

Measuring Time series Analysis

Measuring Time Series Forecasting Performance

- The fact that the future is wholly unknown and can only be predicted from what has already occurred is a significant distinction in forecasting. The ability of a time series forecasting model to predict the future is defined by its performance
- Time series prediction performance measurements provide a summary of the forecast model's skill and capability in making the forecasts.
- There are numerous performance metrics from which to pick. Knowing which metric to use and how to interpret the data might be difficult. .

Evaluation Metrics to Measure Performance

Now, let us have a look at the popular evaluation metrics used to measure the performance of a time-series forecasting model.

R-Squared

The stationary R-squared is used in time series forecasting as a measure that compares the stationary part of the model to a simple mean model. It is defined as,

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

Where,

SSres denotes the sum of squared residuals from expected values .

SStot denotes the sum of squared deviations from the dependent variable's sample mean.

It denotes the proportion of the dependent variable's variance that may be explained by the independent variable's variance. A high R^2 value shows that the model's variance is similar to that of the true values, whereas a low R^2 value suggests that the two values are not strongly related.

Mean Absolute Error (MAE) :

The MAE is defined as the average of the absolute difference between forecasted and true values. Where y_i is the expected value and x_i is the actual value (shown below formula). The letter n represents the total number of values in the test set.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

The MAE shows us how much inaccuracy we should expect from the forecast on average.

MAE = 0 means that the anticipated values are correct, and the error statistics are in the original units of the forecasted values.

The lower the MAE value, the better the model; a value of zero indicates that the forecast is error-free. In other words, the model with the lowest MAE is deemed superior when comparing many models.

However, because MAE does not reveal the proportional scale of the error, it can be difficult to distinguish between large and little errors.

It can be combined with other measures to see if the errors are higher (see Root Mean Square Error below). Furthermore, MAE might obscure issues related to low data volume; for more information, check the last two metrics in this article.

Mean Absolute Percentage Error (MAPE)

MAPE is the proportion of the average absolute difference between projected and true values divided by the true value. The anticipated value is F_t , and the true value is A_t . The number n refers to the total number of values in the test set.

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

It works better with data that is free of zeros and extreme values because of the in-denominator. The MAPE value also takes an extreme value if this value is exceedingly tiny or huge.

The model is better if the MAPE is low. Remember that MAPE works best with data that is devoid of zeros and extreme values. MAPE, like MAE, understates the impact of big but rare errors caused by extreme values.

Mean Square Error can be utilized to address this issue. This statistic may obscure issues related to low data volume; for more information, check the last two metrics in this article.

Root Mean Squared Error(RMSE)

This measure is defined as the square root of mean square error and is an extension of MSE. Where y' denotes the predicted value and y denotes the actual value. The number n refers to the total number of values in the test set. This statistic, like MSE, penalizes greater errors more.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$$

This statistic is likewise always positive, with lower values indicating higher performance. The RMSE number is in the same unit as the projected value, which is an advantage of this technique. In comparison to MSE, this makes it easier to comprehend.

The RMSE can also be compared to the MAE to see whether there are any substantial but uncommon inaccuracies in the forecast. The wider the gap between RMSE and MAE, the more erratic the error size. This statistic can mask issues with low data volume.

Normalized Root Mean Squared Error (NRMSE)

The normalized RMSE is used to calculate NRMSE, which is an extension of RMSE. The mean or the range of actual values are the two most used methods for standardizing RMSE (difference of minimum and maximum values). The maximum true value is y_{max} , while the smallest true value is y_{min} .

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}} \text{ or } \text{NRMSE} = \frac{\text{RMSE}}{\bar{y}}$$

NRMSE is frequently used to compare datasets or forecasting models with varying sizes (units and gross revenue, for example).

The smaller the value, the better the model's performance. When working with little amounts of data, this metric can be misleading.

However, Weighted Absolute Percentage Error and Weighted Mean Absolute Percentage Error can help.

Weighted Mean Absolute Percentage Error (WMAPE)

WMAPE (sometimes called wMAPE) is an abbreviation for Weighted Mean Absolute Percentage Error. It is a measure of a forecasting method's prediction accuracy. It is a MAPE version in which errors are weighted by real values (e.g. in the case of sales forecasting, errors are weighted by sales volume).

$$\text{WMAPE} = \frac{\sum_{t=1}^n |A_t - F_t|}{\sum_{t=1}^n |A_t|}$$

where A is the current data vector and F is the forecast This metric has an advantage over MAPE in that it avoids the 'infinite error' problem.

The higher the model's performance, the lower the WMAPE number. When evaluating forecasting models, this metric is useful for low volume data where each observation has a varied priority.

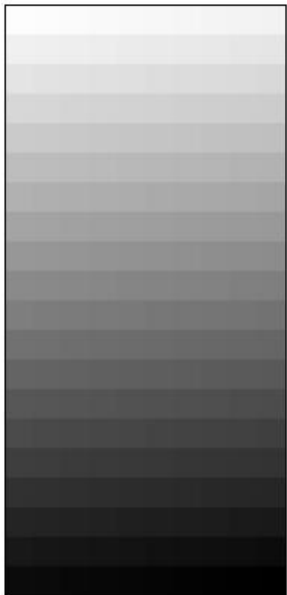
The weight value of observations with a higher priority is higher. The WMAPE number increases as the error in high-priority forecast values grows.

Data Visualization

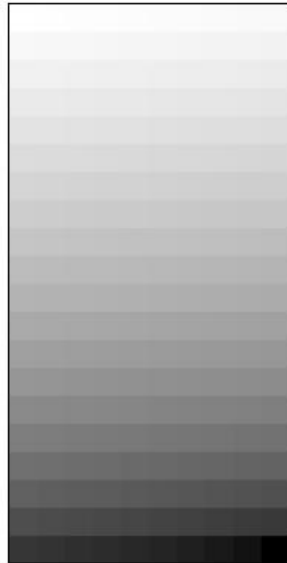
- Why data visualization?
 - Gain insight into an information space by mapping data onto graphical primitives
 - Provide qualitative overview of large data sets
 - Search for patterns, trends, structure, irregularities, relationships among data
 - Help find interesting regions and suitable parameters for further quantitative analysis
 - Provide a visual proof of computer representations derived
- **Categorization of visualization methods:**
 - **Pixel-oriented visualization techniques**
 - **Geometric projection visualization techniques**
 - **Icon-based visualization techniques**
 - **Hierarchical visualization techniques**
 - **Visualizing complex data and relations**

Pixel-Oriented Visualization Techniques

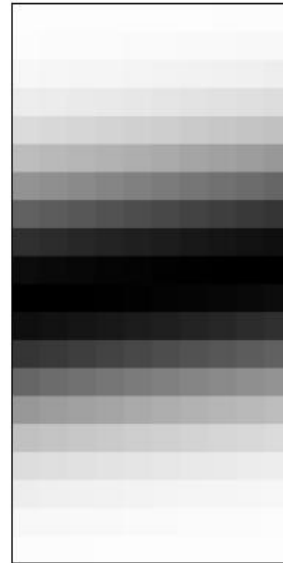
- For a data set of m dimensions, create m windows on the screen, one for each dimension
- The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows
- The colors of the pixels reflect the corresponding values



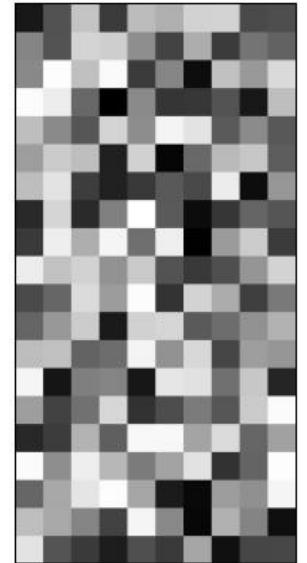
(a) Income



(b) Credit Limit



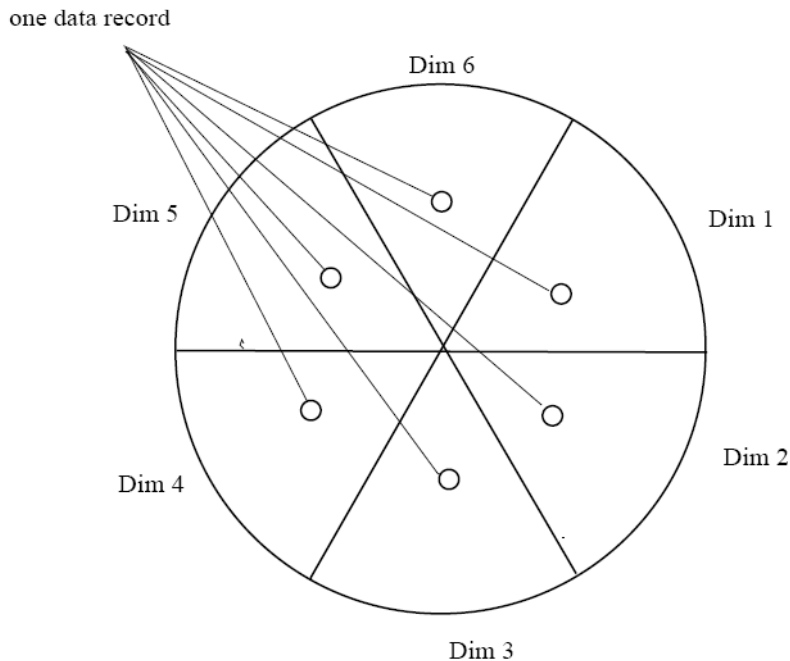
(c) transaction volume



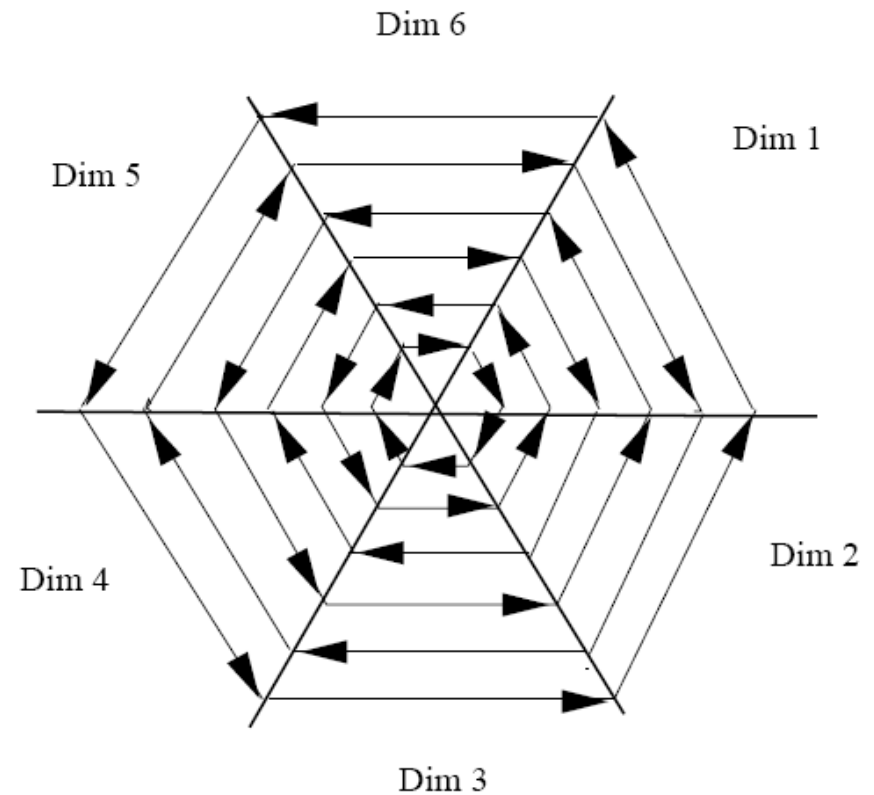
(d) age

Laying Out Pixels in Circle Segments

- To save space and show the connections among multiple dimensions, space filling is often done in a circle segment



(a) Representing a data record in circle segment

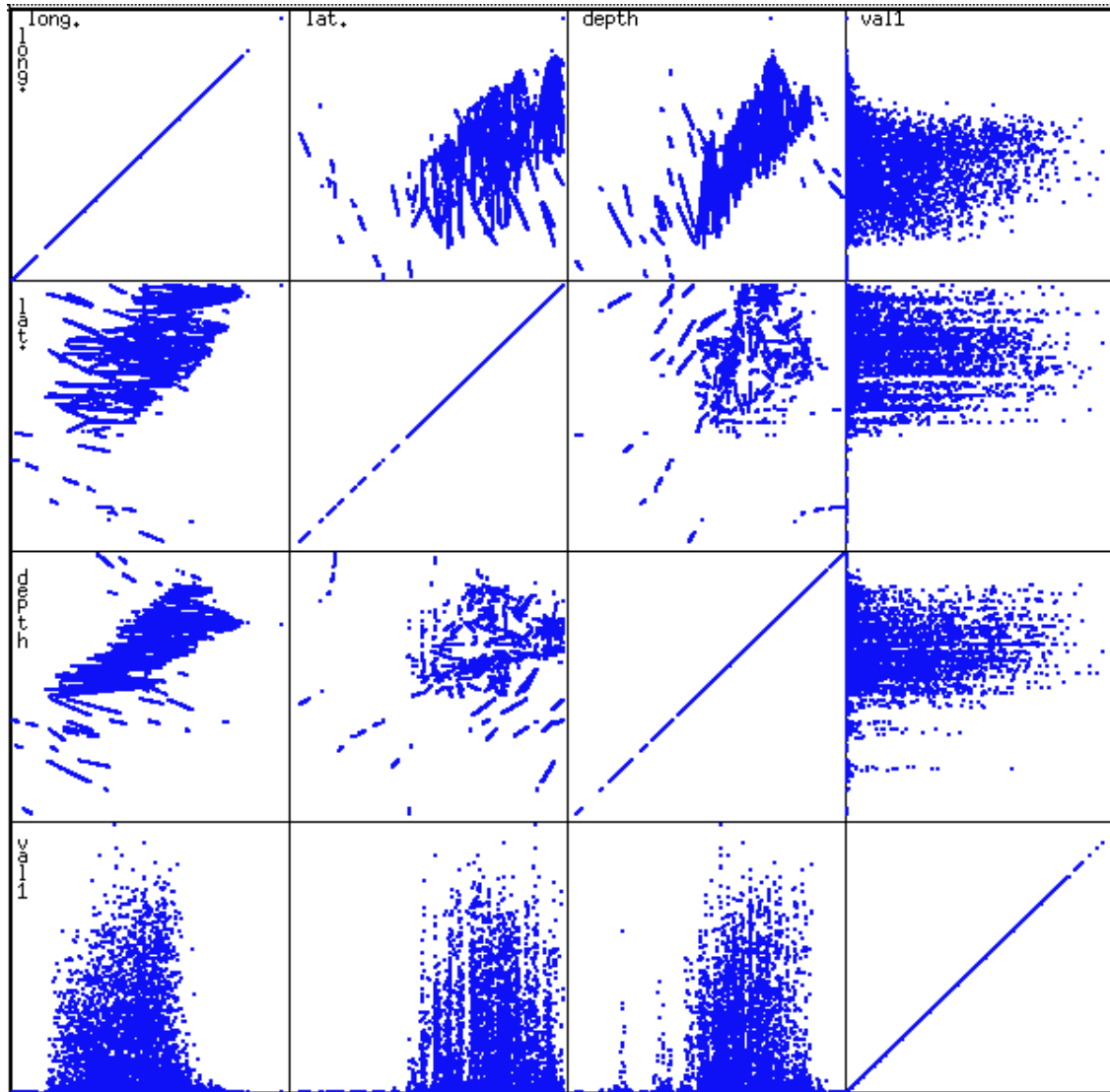


(b) Laying out pixels in circle segment

Geometric Projection Visualization Techniques

- Visualization of geometric transformations and projections of the data
- Methods
 - Direct visualization
 - Scatterplot and scatterplot matrices
 - Landscapes
 - Projection pursuit technique: Help users find meaningful projections of multidimensional data
 - Projection views
 - Hyperslice
 - Parallel coordinates

Scatterplot Matrices

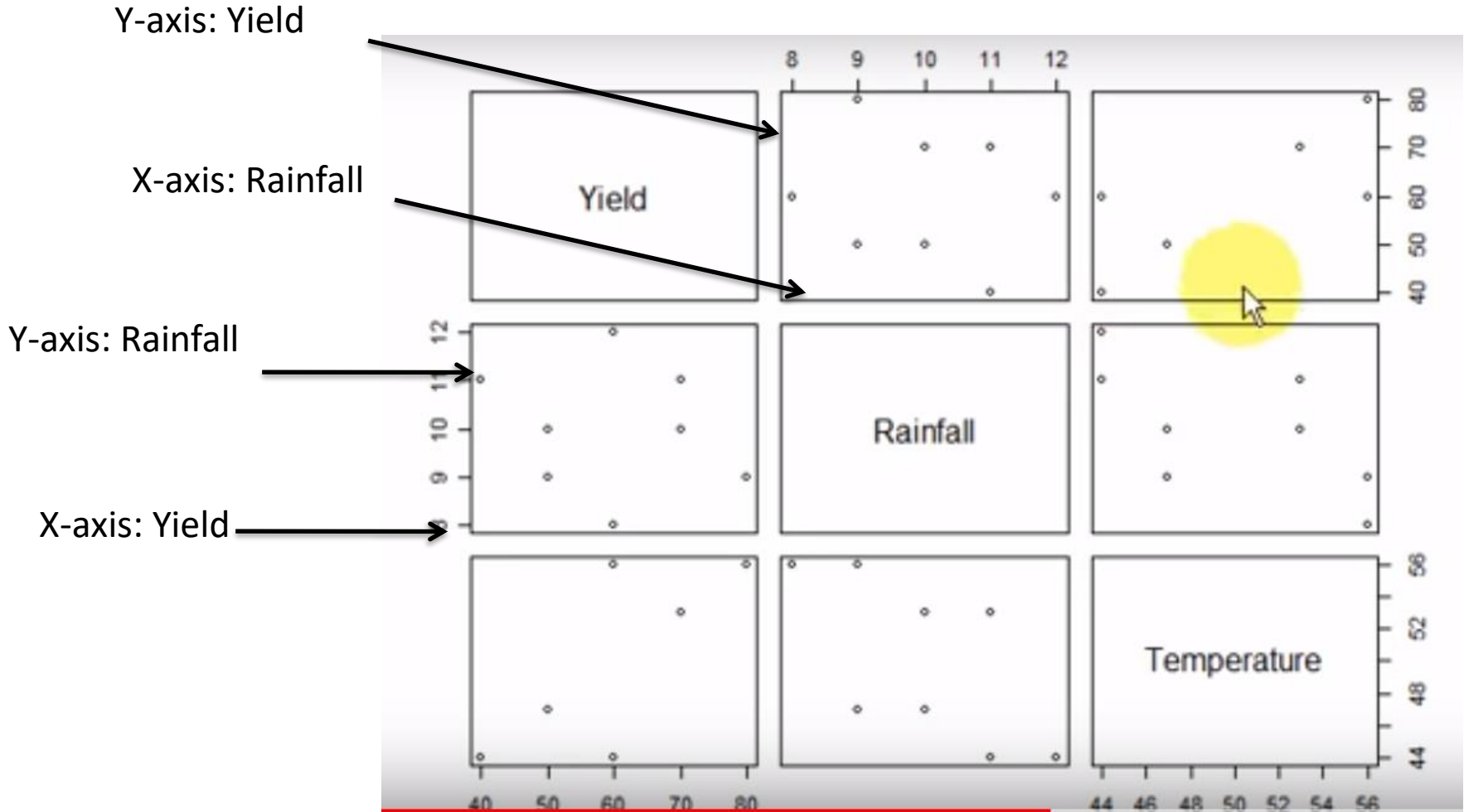


Used by permission of M. Ward, Worcester Polytechnic Institute

Matrix of scatterplots (x-y-diagrams) of the k-dim. data [total of $(k^2/2-k)$ scatterplots]

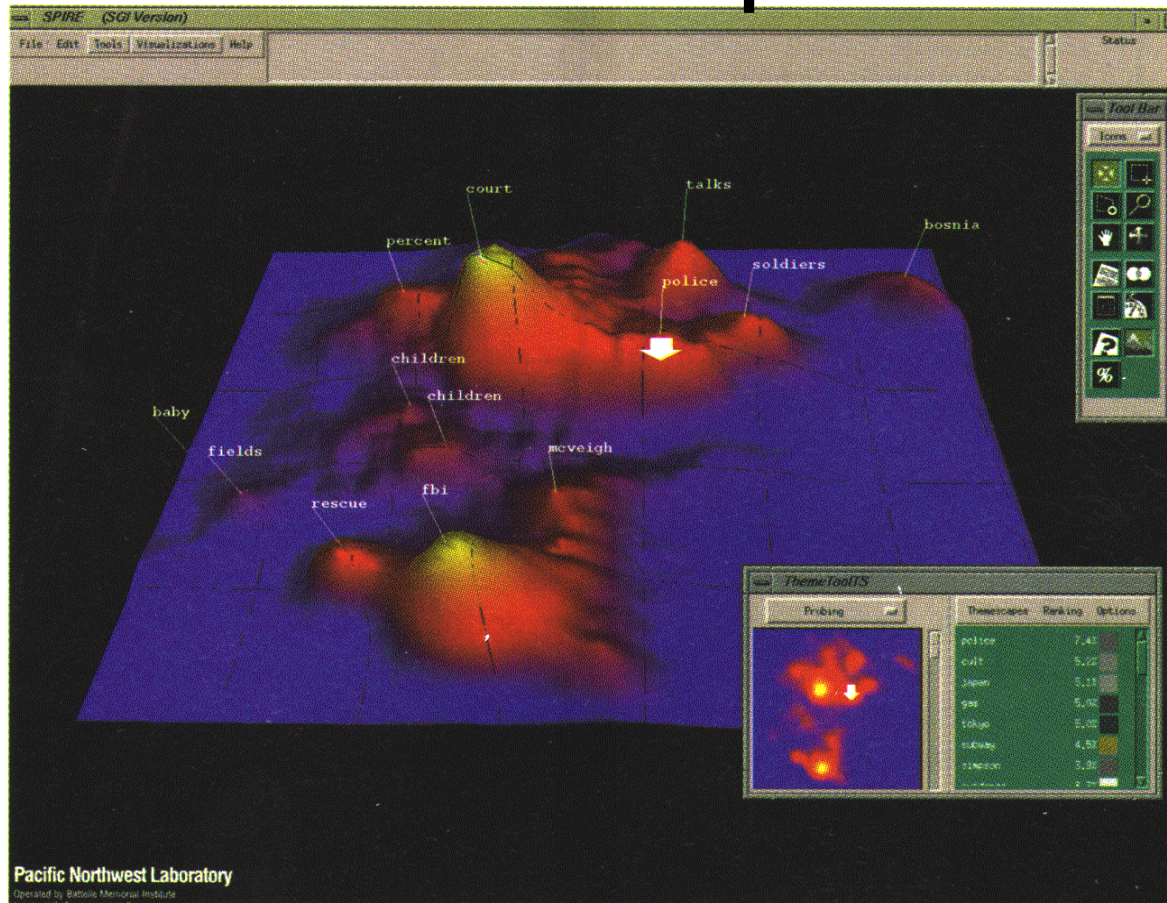
Interpreting Scatter plot matrix:

Leave the diagonal squares. Now, for example, in the below scatter plot. Yield will be on Y-axis for it left and right sides and it will be on X-axis for top and bottom.



Landscapes

Used by permission of B. Wright, Visible Decisions Inc.

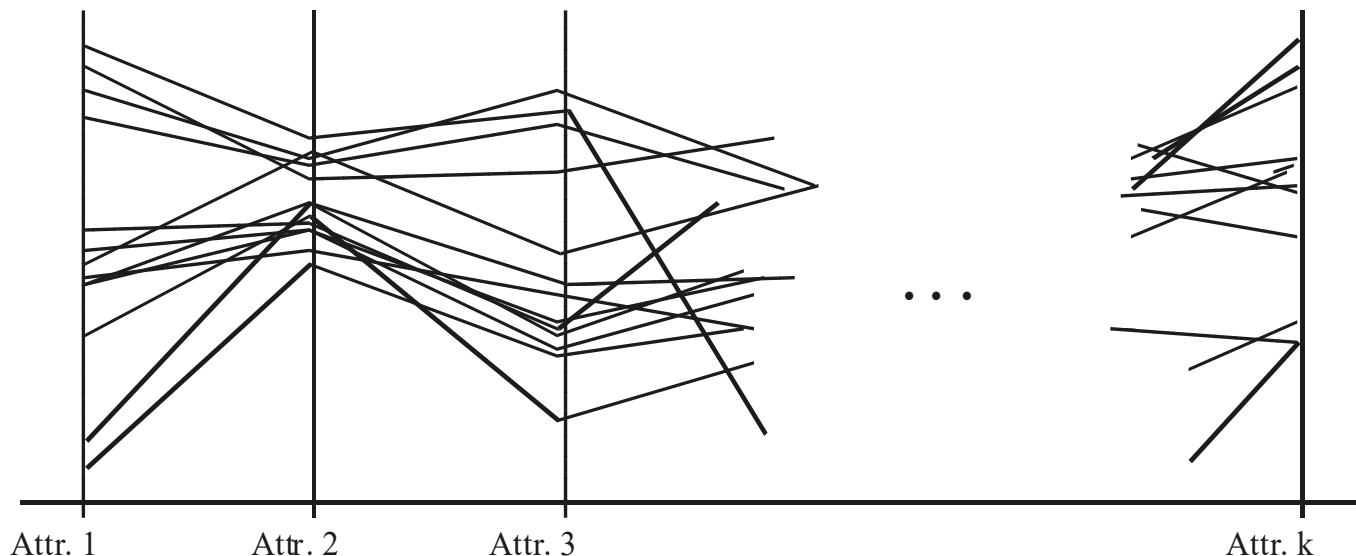


news articles
visualized as
a landscape

- Visualization of the data as perspective landscape
- The data needs to be transformed into a (possibly artificial) 2D spatial representation which preserves the characteristics of the data

Parallel Coordinates

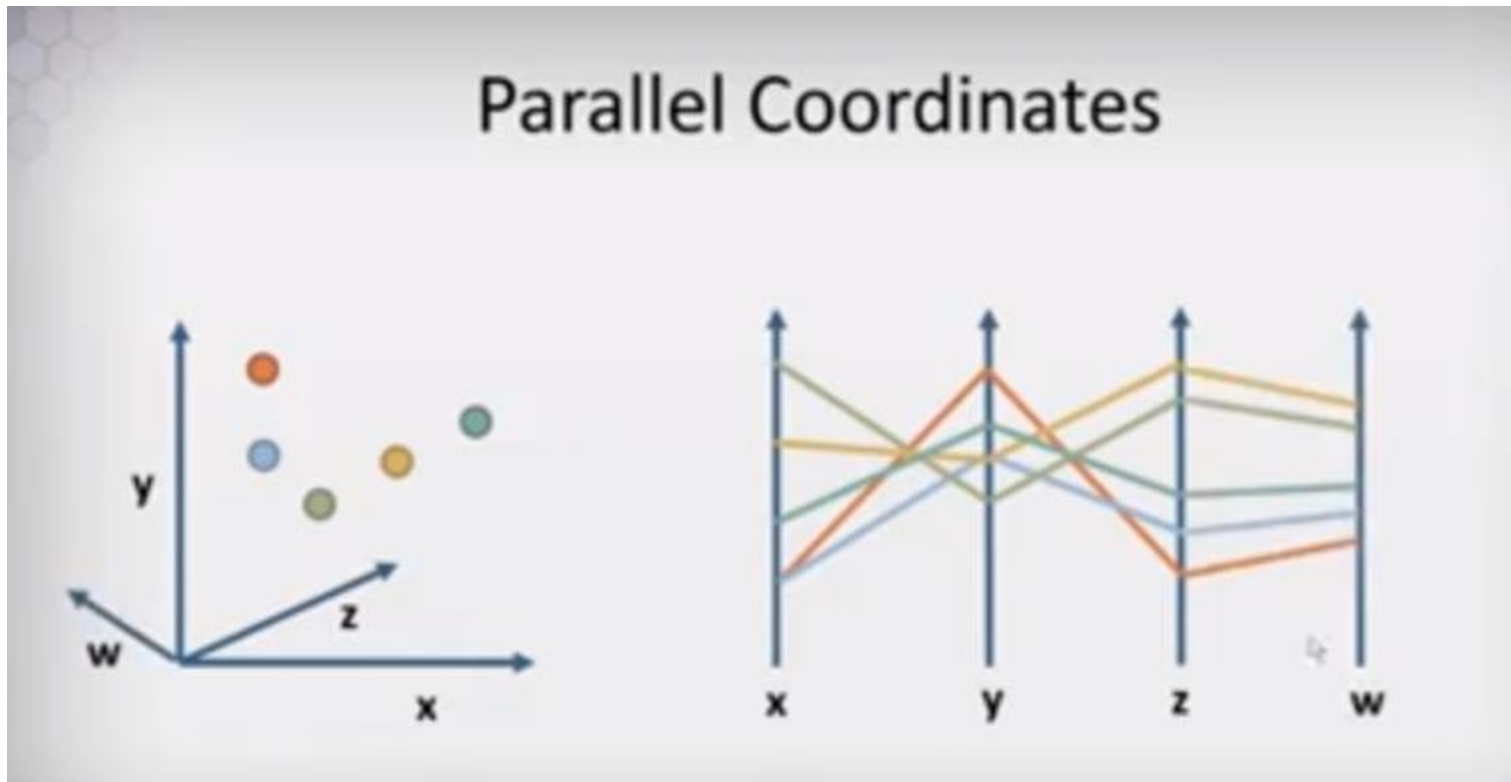
- n equidistant axes which are parallel to one of the screen axes and correspond to the attributes
- The axes are scaled to the [minimum, maximum]: range of the corresponding attribute
- Every data item corresponds to a polygonal line which intersects each of the axes at the point which corresponds to the value for the attribute



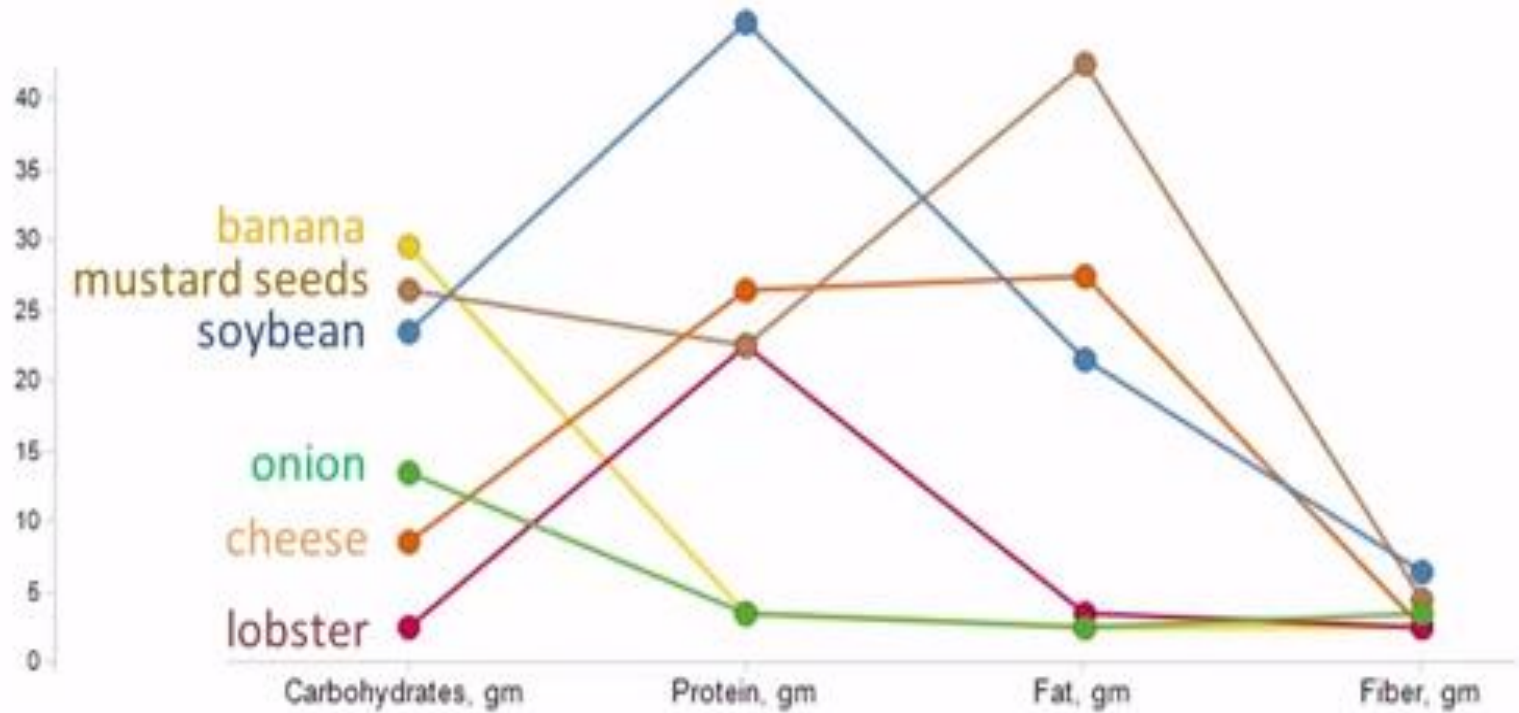
Note

Parallel Coordinates are useful to represent higher dimension of data. X, Y, X, are dimensions.

They are represented as vertical lines and the points are noted on those line and mapped.



Example



Icon-Based Visualization Techniques

- Visualization of the data values as features of icons
- Typical visualization methods
 - Chernoff Faces
 - Stick Figures
- General techniques
 - Shape coding: Use shape to represent certain information encoding
 - Color icons: Use color icons to encode more information
 - Tile bars: Use small icons to represent the relevant feature vectors in document retrieval

ICON BASED VISUALIZATION TECHNIQUES

- Uses small icons to represent multidimensional data values.

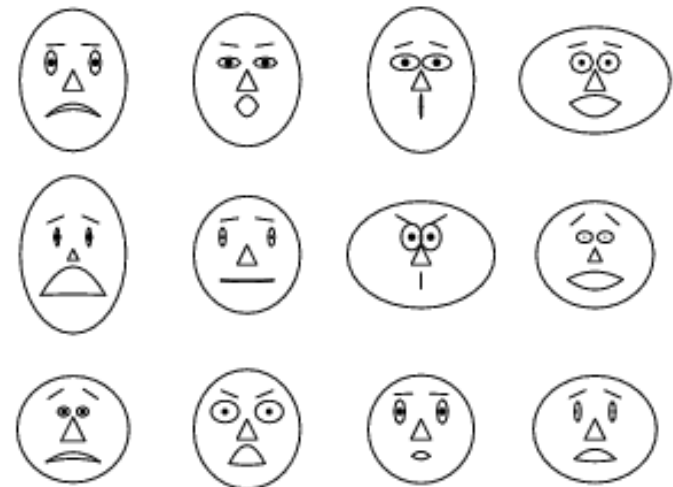
TWO POPULAR ICON-BASED TECHNIQUES

1. Chernoff faces
2. Stick figures

Chernoff Faces

- A way to display variables on a two-dimensional surface, e.g., let x be eyebrow slant, y be eye size, z be nose length, etc.
- The figure shows faces produced using 10 characteristics--head eccentricity, eye size, eye spacing, eye eccentricity, pupil size, eyebrow slant, nose size, mouth shape, mouth size, and mouth opening): Each assigned one of 10 possible values, generated using [Mathematica](#) (S. Dickson)

- REFERENCE: Gonick, L. and Smith, W. [The Cartoon Guide to Statistics](#). New York: Harper Perennial, p. 212, 1993
- Weisstein, Eric W. "Chernoff Face." From *MathWorld*--A Wolfram Web Resource. mathworld.wolfram.com/ChernoffFace.html



1. Chernoff face: Each dimension is represented as part of a face.

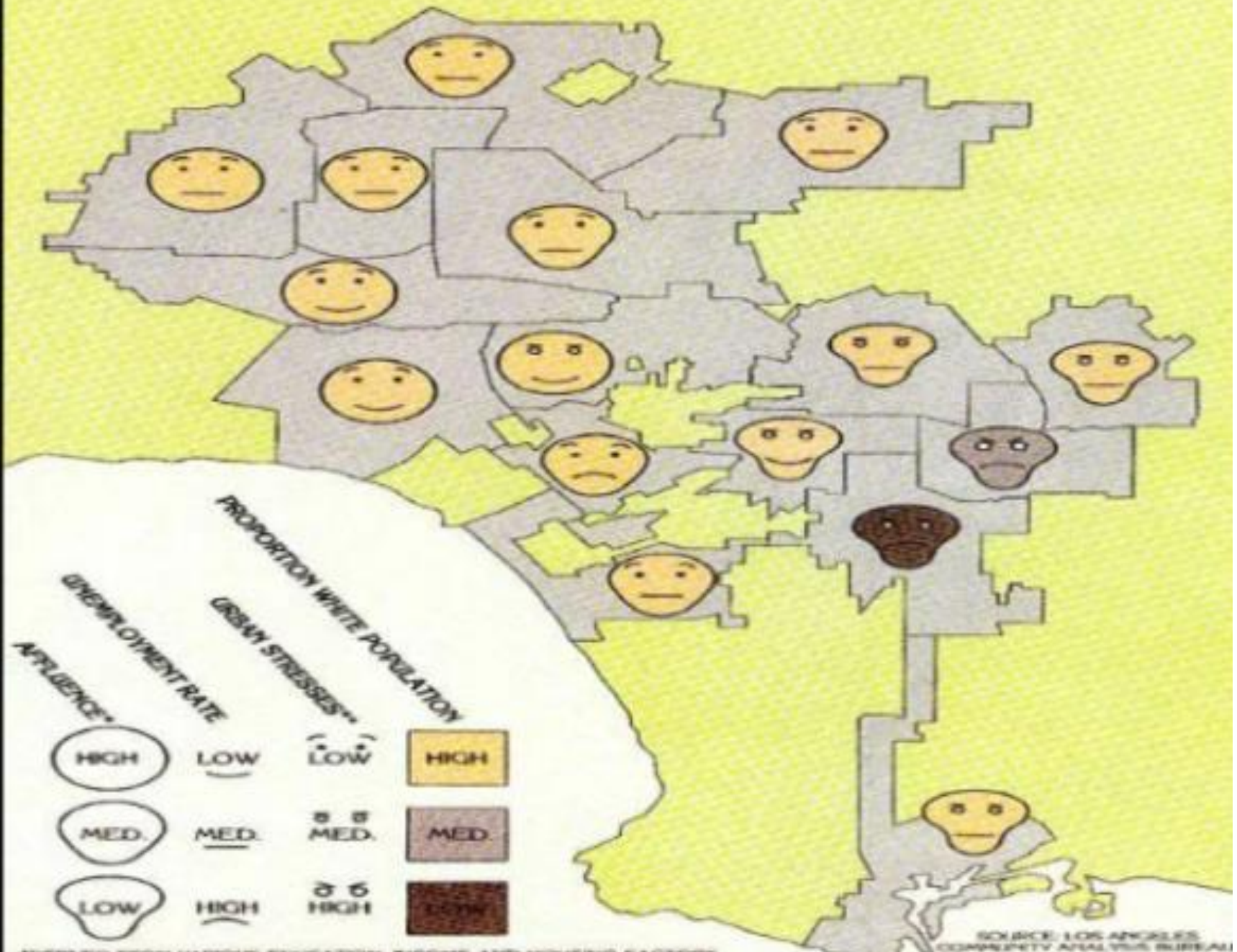


- A way to display variables on a two-dimensional surface. For instance, let x be eyebrow slant, y be eye size, z be nose length, etc. The above figures show faces produced using 10 characteristics-- head eccentricity, eye size, eye spacing, eye eccentricity, pupil size, eyebrow slant, nose size, mouth shape, mouth size, and mouth opening.

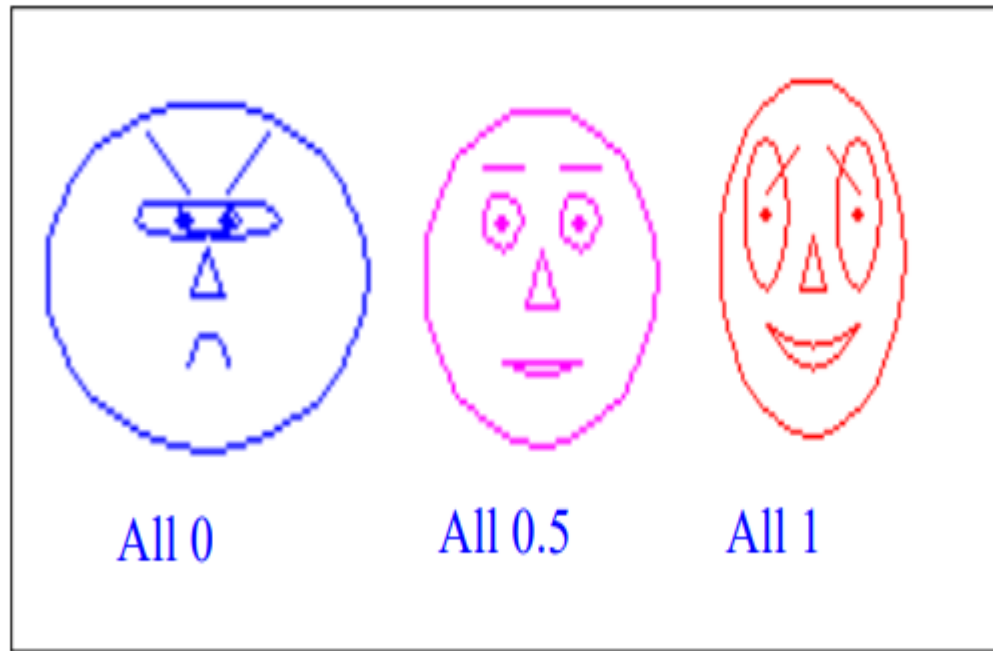
CHERNOFF FACES

- Introduced in 1973 by statistician Herman Chernoff.
- They display multidimensional data of up to 18 variables (or dimensions) as a cartoon human face.
- Chernoff faces help reveal trends in the data.

Life in Los Angeles



SOURCE: LOS ANGELES COMMUNITY ANALYSIS BUREAU

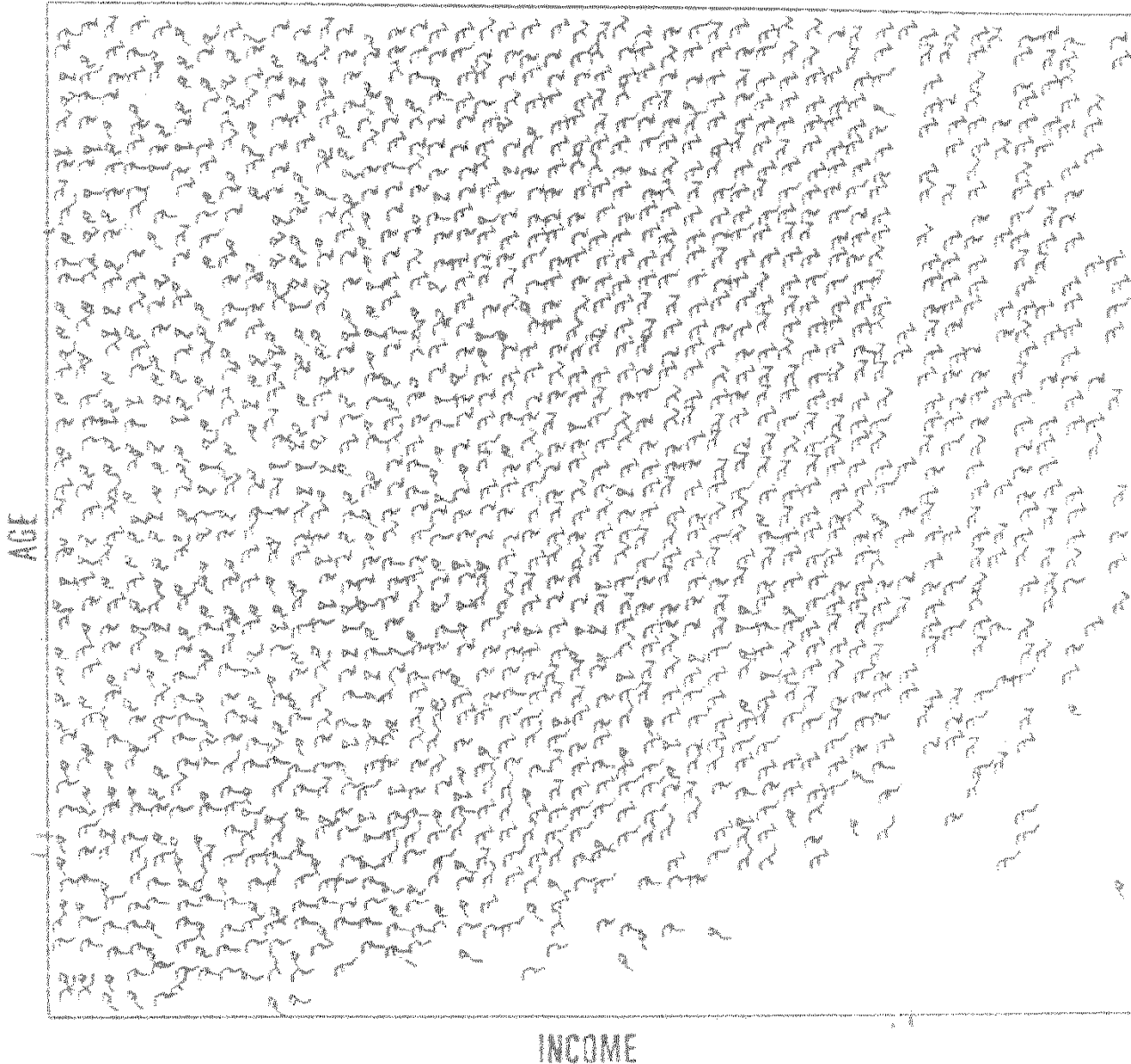


Chernoff faces with 10 facial characteristic parameters:

1. head eccentricity, 2. eye eccentricity, 3. pupil size, 4. eyebrow slant, 5. nose size, 6. mouth shape, 7. eye spacing, 8. eye size, 9. mouth length, and 10. degree of mouth opening

Stick Figure

used by permission of G. Grinstein, University of Massachusetts at Lowell



A census data figure showing age, income, gender, education, etc.

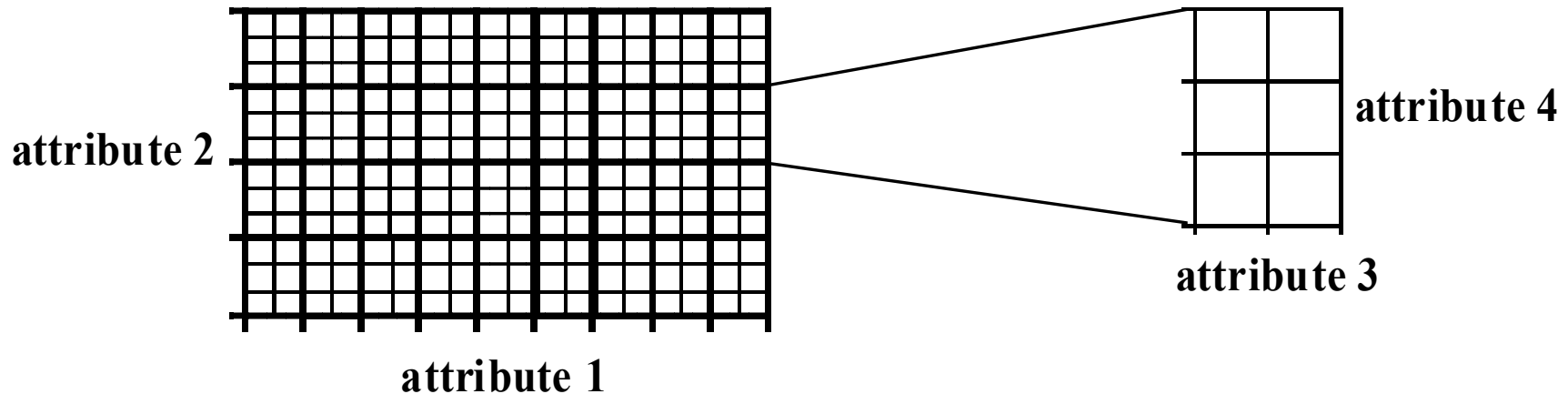
A 5-piece stick figure (1 body and 4 limbs w. different angle/length)

Two attributes mapped to axes, remaining attributes mapped to angle or length of limbs". Look at texture pattern

Hierarchical Visualization Techniques

- Visualization of the data using a hierarchical partitioning into subspaces
- Methods
 - Dimensional Stacking
 - Worlds-within-Worlds
 - Tree-Map
 - Cone Trees
 - InfoCube

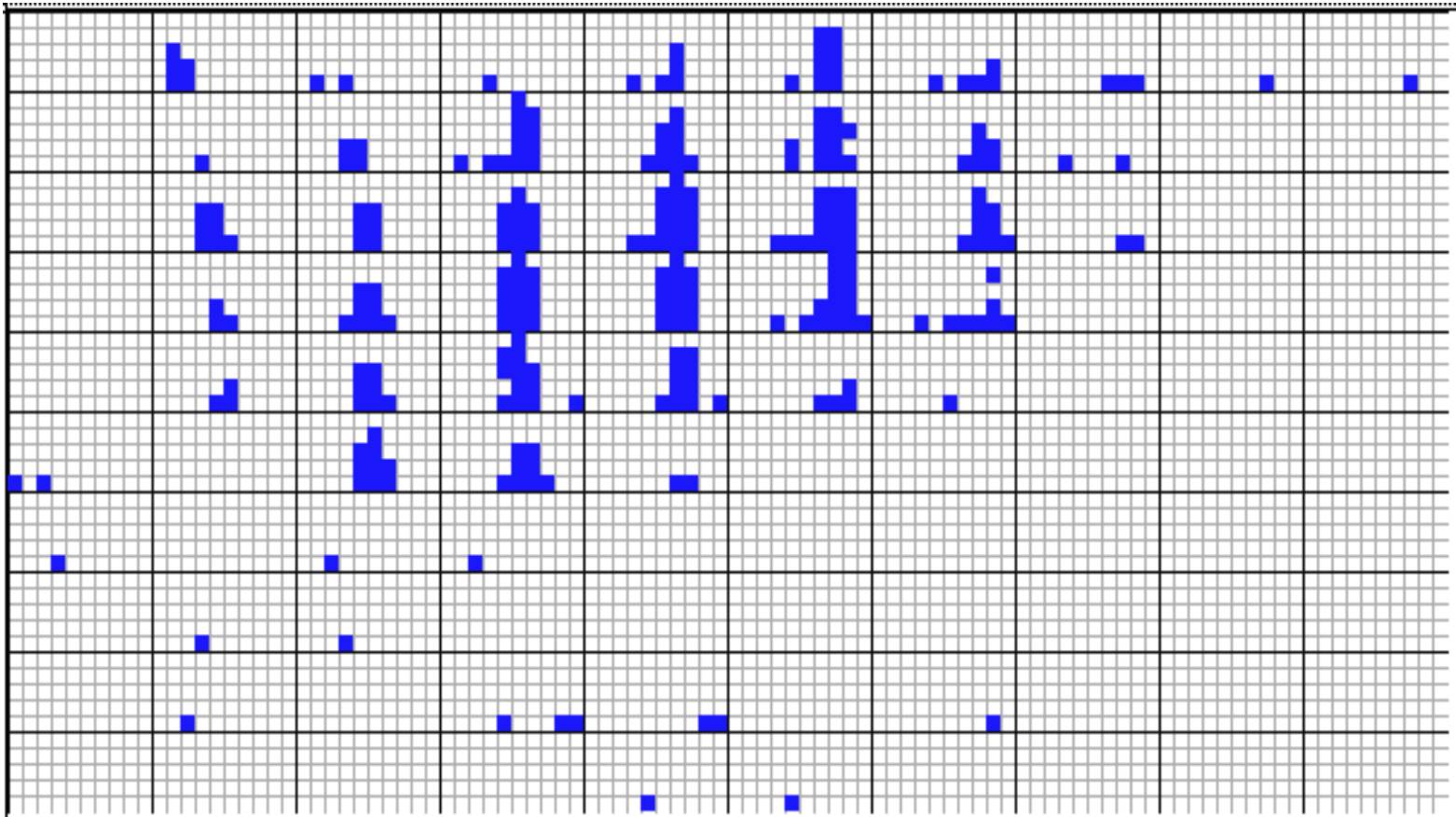
Dimensional Stacking



- Partitioning of the n-dimensional attribute space in 2-D subspaces, which are 'stacked' into each other
- Partitioning of the attribute value ranges into classes. The important attributes should be used on the outer levels.
- Adequate for data with ordinal attributes of low cardinality
- But, difficult to display more than nine dimensions
- Important to map dimensions appropriately

Dimensional Stacking

Used by permission of M. Ward, Worcester Polytechnic Institute

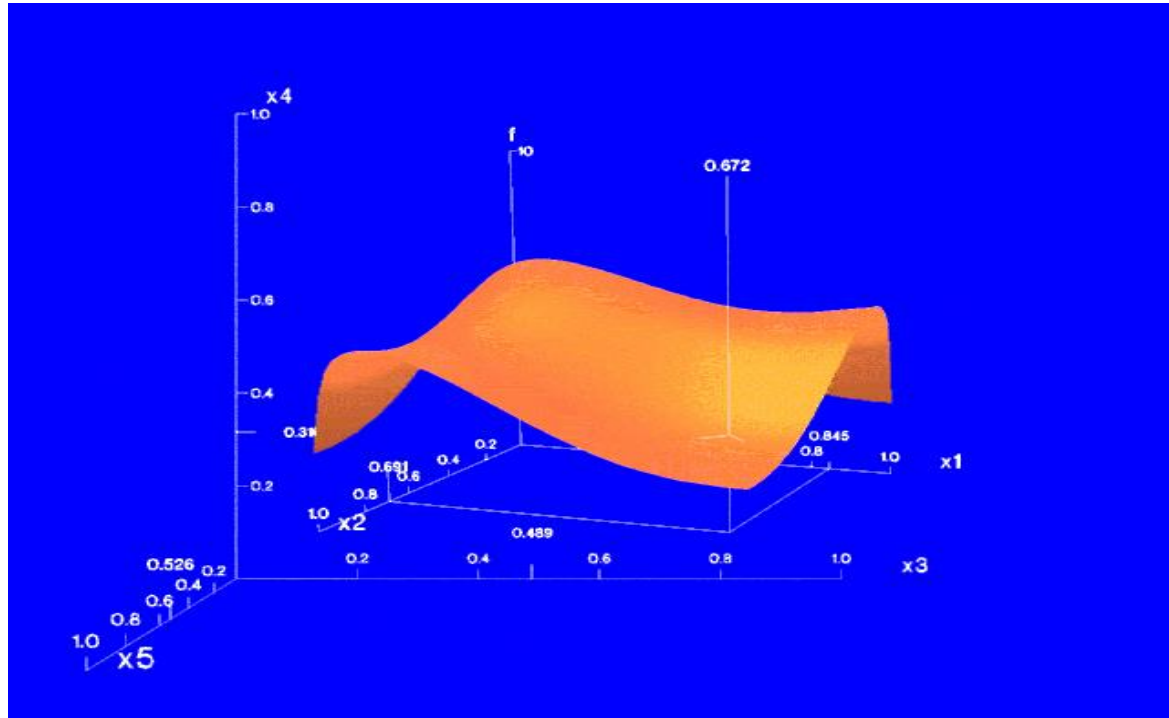


Visualization of oil mining data with longitude and latitude mapped to the outer x-, y-axes and ore grade and depth mapped to the inner x-, y-axes

Worlds-within-Worlds

- Assign the function and two most important parameters to innermost world
- Fix all other parameters at constant values - draw other (1 or 2 or 3 dimensional worlds choosing these as the axes)
- Software that uses this paradigm

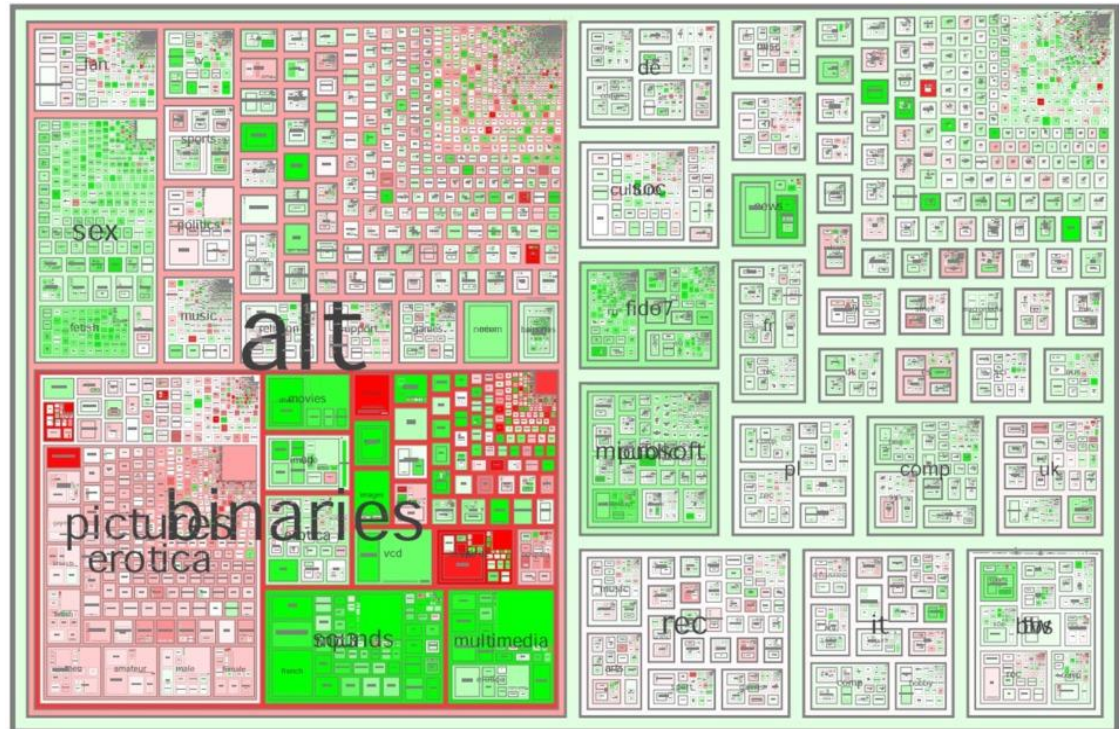
- N-vision: Dynamic interaction through data glove and stereo displays, including rotation, scaling (inner) and translation (inner/outer)
- Auto Visual: Static interaction by means of queries



Tree-Map

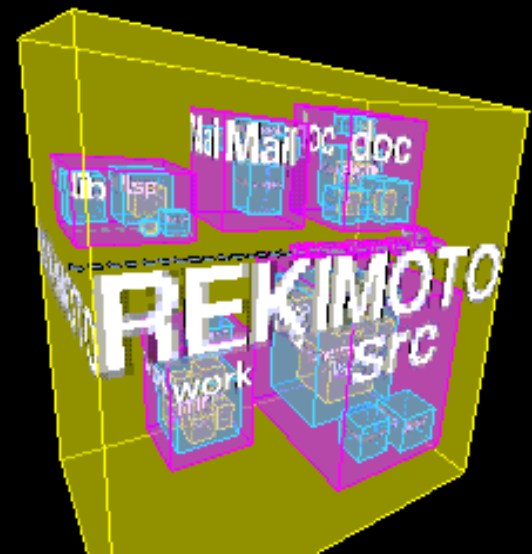
- Screen-filling method which uses a hierarchical partitioning of the screen into regions depending on the attribute values
- The x- and y-dimension of the screen are partitioned alternately according to the attribute values (classes)

MSR Netscan Image



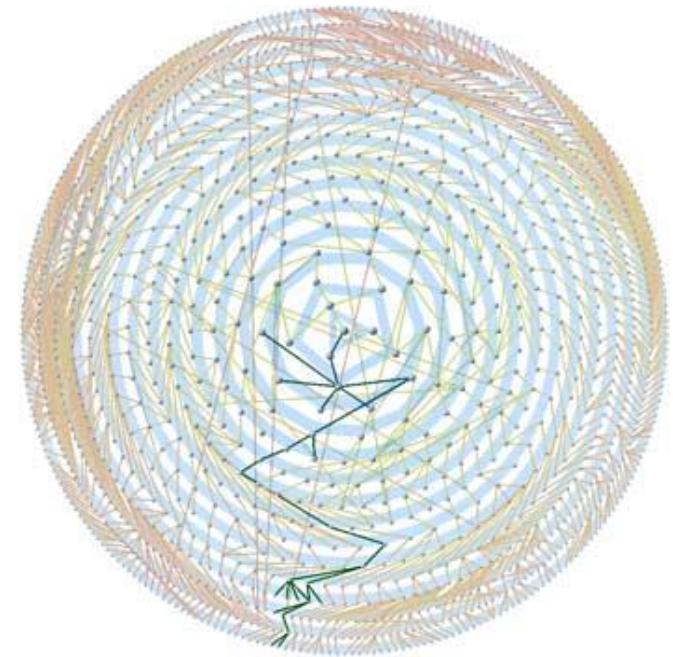
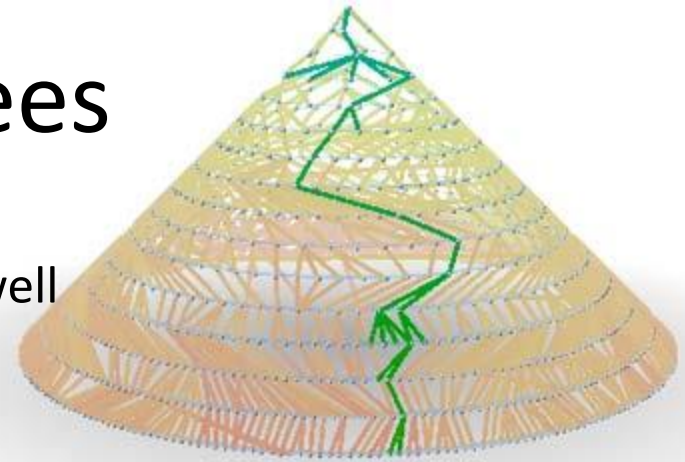
InfoCube

- A 3-D visualization technique where hierarchical information is displayed as nested semi-transparent cubes
- The outermost cubes correspond to the top level data, while the subnodes or the lower level data are represented as smaller cubes inside the outermost cubes, and so on



Three-D Cone Trees

- *3D cone tree* visualization technique works well for up to a thousand nodes or so
- First build a *2D circle tree* that arranges its nodes in concentric circles centered on the root node
- Cannot avoid overlaps when projected to 2D
- G. Robertson, J. Mackinlay, S. Card. “Cone Trees: Animated 3D Visualizations of Hierarchical Information”, *ACM SIGCHI'91*
- Graph from Nadeau Software Consulting website: Visualize a social network data set that models the way an infection spreads from one person to the next



Visualizing Complex Data and Relations

- Visualizing non-numerical data: text and social networks
- Tag cloud: visualizing user-generated tags
 - The importance of tag is represented by font size/color
 - Besides text data, there are also methods to visualize relationships, such as visualizing social networks



Newsmap: Google News Stories in 2005