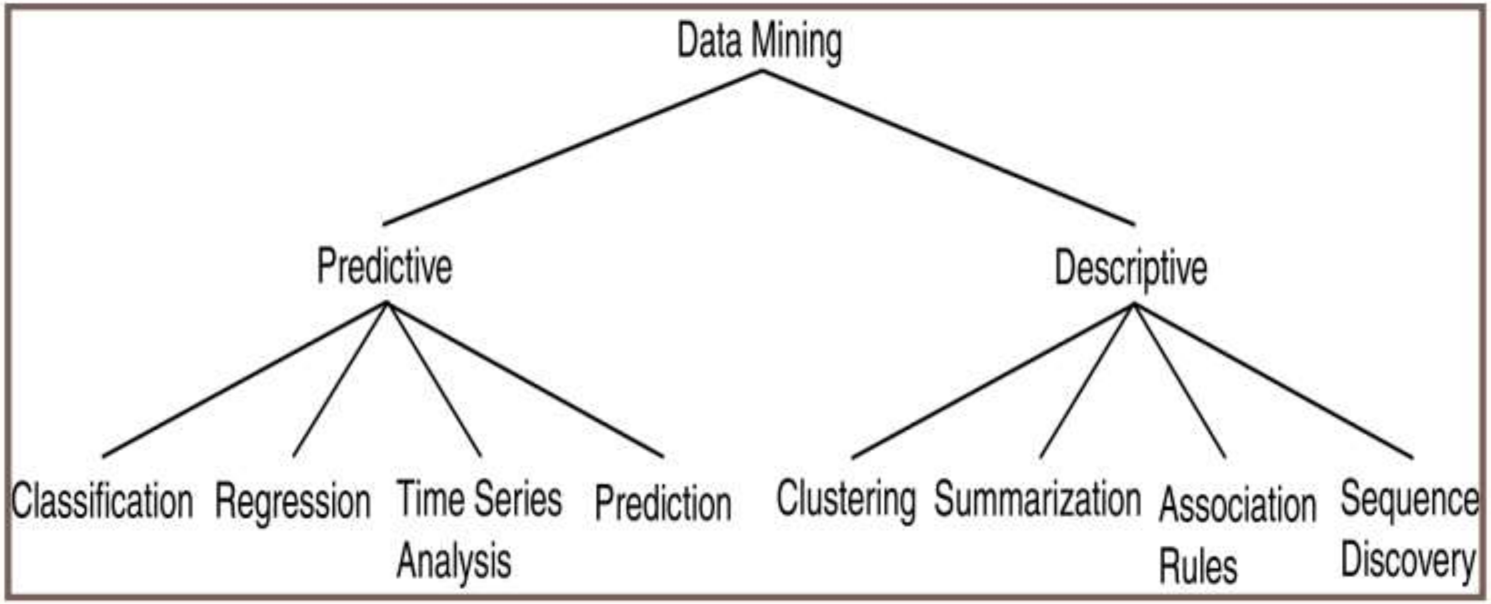
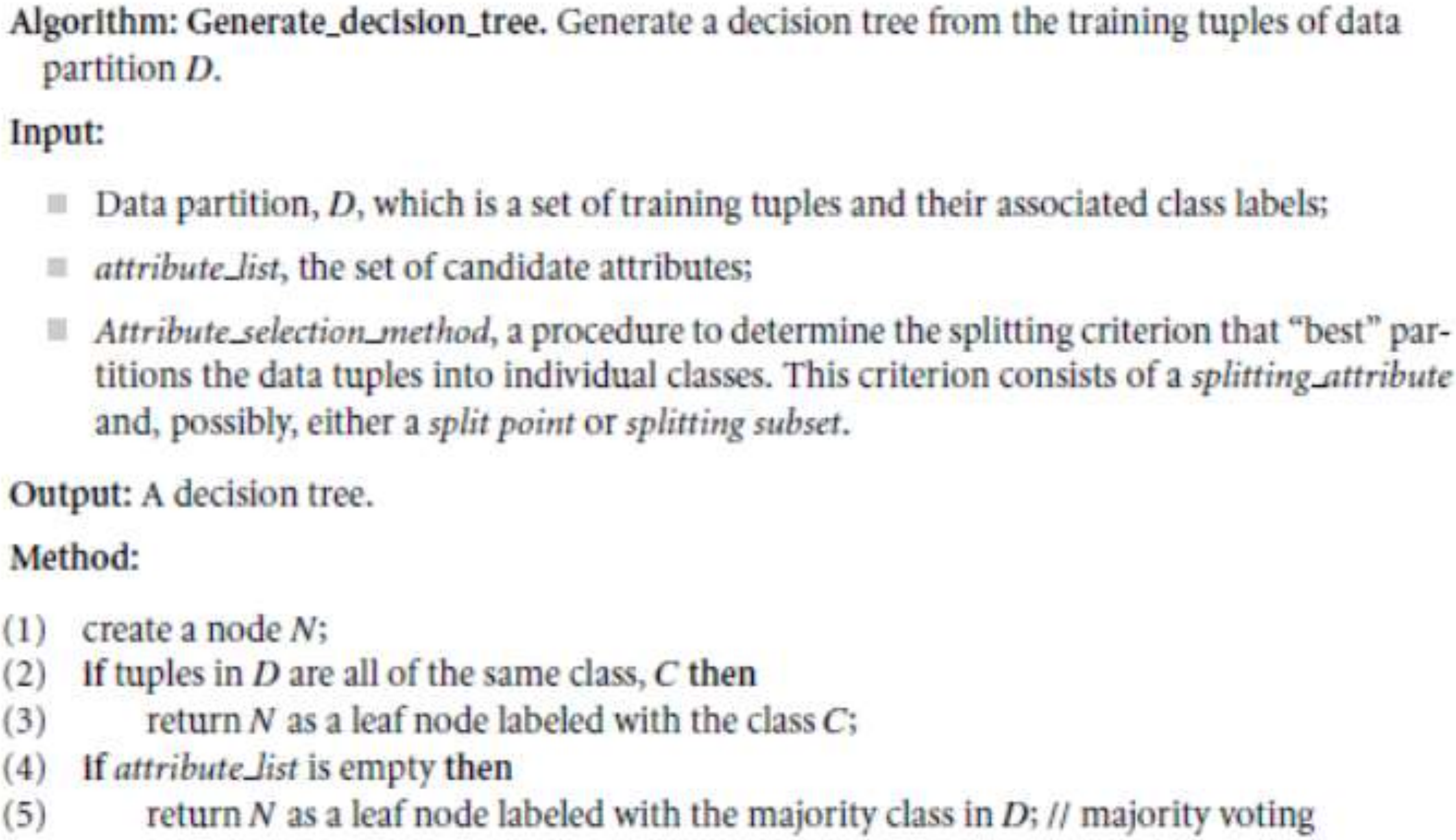
Data Mining Models and Tasks



Algorithm For Decision Tree Induction:



Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition *D*.

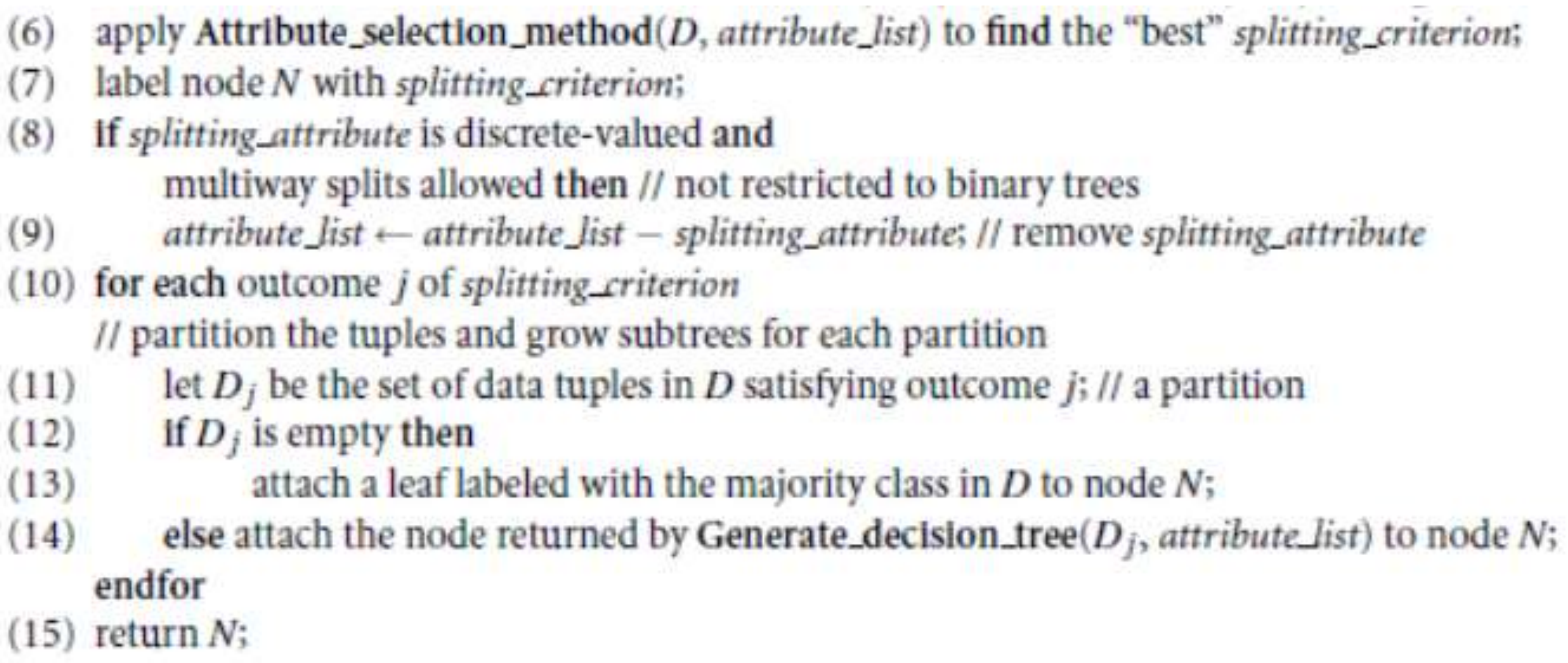
Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
create a node $N$;
if tuples in $D$ are all of the same class, $C$ then
    return $N$ as a leaf node labeled with the class $C$;
if attribute_list is empty then
    return $N$ as a leaf node labeled with the majority class in $D$; // majority voting
```



```
apply Attribute_selection_method($D$, attribute_list) to find the "best" splitting_criterion;
label node $N$ with splitting_criterion;
if splitting_attribute is discrete-valued and
        multiway splits allowed then // not restricted to binary trees
        attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
for each outcome $j$ of splitting_criterion
// partition the tuples and grow subtrees for each partition
        let $D_j$ be the set of data tuples in $D$ satisfying outcome $j$; // a partition
        if $D_j$ is empty then
                attach a leaf labeled with the majority class in $D$ to node $N$;
        else attach the node returned by Generate_decision_tree($D_j$, attribute_list) to node $N$;
endfor
return $N$;
```

The algorithm is called with three parameters:

- Data partition
- Attribute list
- Attribute selection method

- The parameter attribute list is a list of attributes describing the tuples.

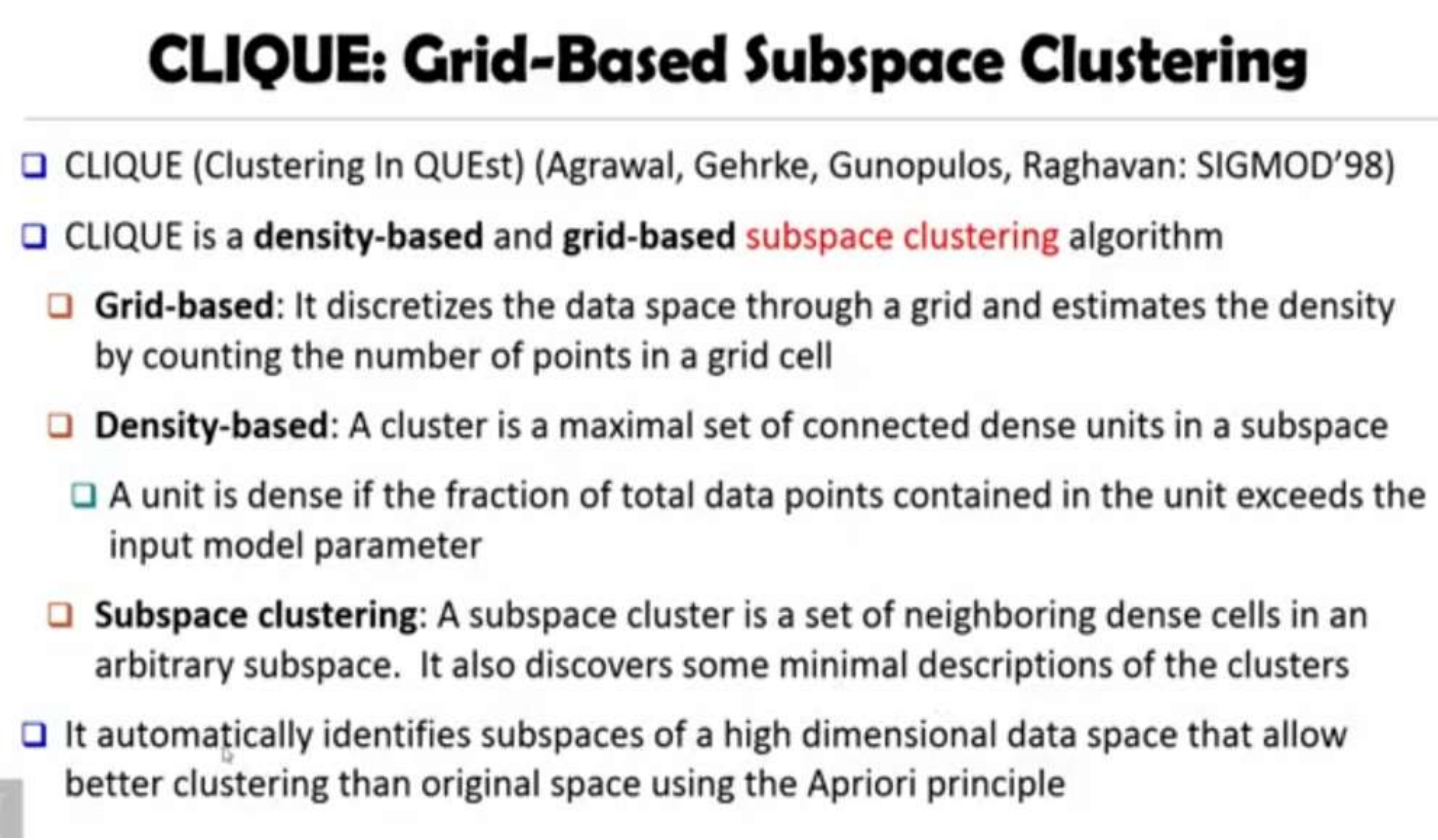
- Attribute selection method specifies a heuristic procedure for selecting the attribute that “best” discriminates the given tuples according to class.

- The tree starts as a single node, N, representing the training tuples in D.

- If the tuples in D are all of the same class, then node N becomes a leaf and is labeled with that class .

- All of the terminating conditions are explained at the end of the algorithm. Otherwise, the algorithm calls Attribute selection method to determine the splitting criterion.

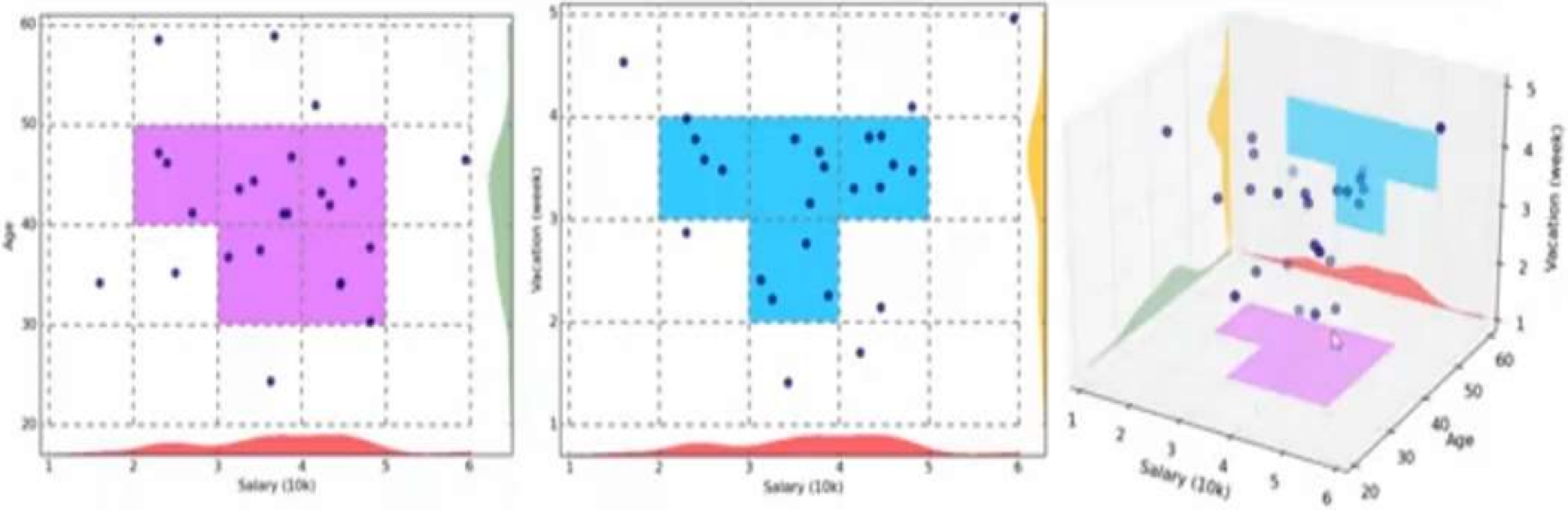
- The splitting criterion tells us which attribute to test at node N by determining the “best” way to separate or partition the tuples in D into individual classes.



CLIQUE: Grid-Based Subspace Clustering

- CLIQUE (Clustering In QUEst) (Agrawal, Gehrke, Gunopulos, Raghavan: SIGMOD'98)
- CLIQUE is a **density-based** and **grid-based** subspace clustering algorithm
  - **Grid-based**: It discretizes the data space through a grid and estimates the density by counting the number of points in a grid cell
  - **Density-based**: A cluster is a maximal set of connected dense units in a subspace
    - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - **Subspace clustering**: A subspace cluster is a set of neighboring dense cells in an arbitrary subspace. It also discovers some minimal descriptions of the clusters
- It automatically identifies subspaces of a high dimensional data space that allow better clustering than original space using the Apriori principle

Example of CLIQUE: Density and Grid-Based Subspace Clustering



- ❑ Start at 1-D space and discretize numerical intervals in each axis into grid
- ❑ Find dense regions (clusters) in each subspace and generate their minimal descriptions
- ❑ Use the dense regions to find promising candidates in 2-D space based on the Apriori principle
- ❑ Repeat the above in level-wise manner in higher dimensional subspaces

Major Steps of the CLIQUE Algorithm

- Identify subspaces that contain clusters
  - Partition the data space and find the number of points that lie inside each cell of the partition
  - Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests
- Generate minimal descriptions for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determine minimal cover for each cluster

DARK DATA

Dark data is a subset of big data but it constitutes the biggest portion of the total volume of big data collected by organizations in a year. Dark data is not usually analyzed or processed because of various reasons by companies but that does not lessen its importance in the context of business value.

That may sound difficult to bear, what with the data not being used for anything most of the time.

However, it can become critical to your business if it is put to proper use. In other words, you might not have to eliminate it from your knowledge base altogether if you know how to use it. It doesn’t have to stay “dark.”

Study trends to know your market and audience

Dark data you gather can be analyzed to determine what is trending based on your market and the audience that comes to your site. For instance, you might find that you have more young customers or the visitors to your site are interested in very specific products.

You might also find that people in certain geographic areas spend more than others. The trends you'll explore will make it easier for you to market your business.

Following categories of unstructured data usually are considered dark data:

- Customer Information
- Log Files
- Previous Employee Information
- Raw Survey Data
- Financial Statements
- Email Correspondences
- Account Information
- Notes or Presentations
- Old Versions of Relevant Documents

Reference:

https://www.kdnuggets.com/2015/11/importance-dark-data-big-data-world.html

https://www.ibm.com/blogs/cloud-computing/2016/05/19/dark-data-impact/

Sequence Pattern Mining

Sequence Pattern Mining, a subset of Data Mining, is the process of identifying frequently occurring ordered events or subsequences as patterns. It is highly useful for retail, telecommunications, and other businesses since it helps them detect sequential patterns for targeted marketing, customer retention, and many other tasks.

Thus, if you come across ordered data, and you extract patterns from the sequence, you are essentially doing Sequence Pattern Mining. The sequence need not have a notion of time, and therefore Sequence Pattern Mining is slightly different from Time-Series Mining.

When you are performing Sequence Pattern Mining, you are essentially:

- Finding frequently occurring patterns
- Comparing sequences
- Finding missing sequence items
- Building efficient indexes for sequence information

Sequence Pattern Mining helps companies to discover sequential patterns, and hence it finds several applications across many fields.

Types of Sequence Pattern Mining Problems

Sequence Pattern Mining can be broadly categorized into two types:

String Mining: This is the subset of Sequence Pattern Mining that deals with text data in a sequence. The data can contain only a limited number of characters. For example, a DNA sequence contains only the letters ‘A’, ’T’, ’C’, and ’G’, and therefore analysis of the same falls within String Mining. Similarly, finding patterns in ASCII character sequences falls under String Mining.

Itemset Mining: This is the broader subset of Sequence Pattern Mining that aims to find patterns in ordered datasets. Itemset Mining generally finds use in Marketing and Sales Applications (increasing co-purchases of items that are frequently brought together, cross-promoting products, managing inventory, setting price levels, and so on). A more detailed introduction to Itemset Mining, a subset of Sequence Pattern Mining, can be found here

Algorithms for Sequence Pattern Mining

GSP (Generalized Sequential Pattern Mining)
SPADE (Sequential Pattern Discovery using Equivalence Class)
Non-Apriori Based Algorithms
PrefixSpan (Prefix-projected Sequential Pattern Mining)

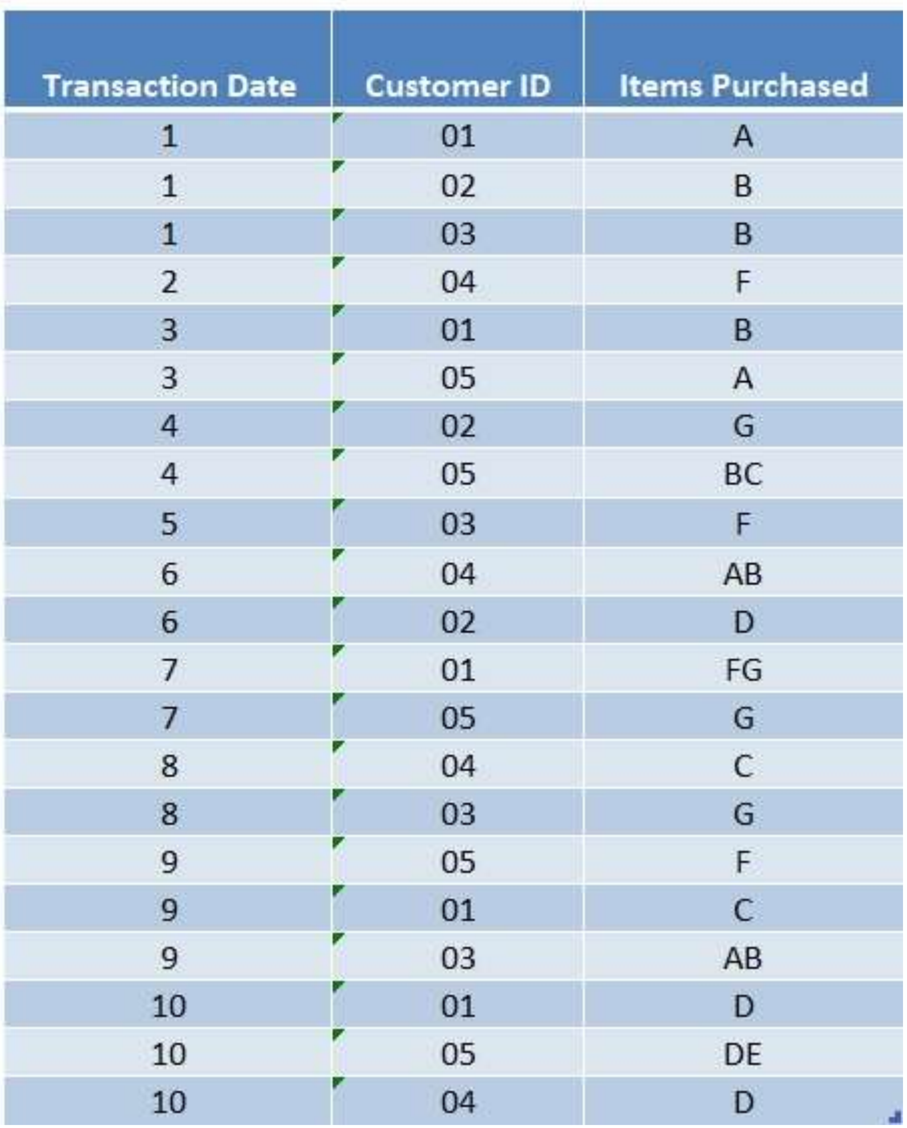
GSP (Generalized Sequential Pattern Mining)

This Sequence Pattern Mining algorithm takes a bottom-up approach to find frequent patterns. Initially, every element is considered as a candidate of length 1. Based on the minimum support, frequent sequences of length 1 are identified.

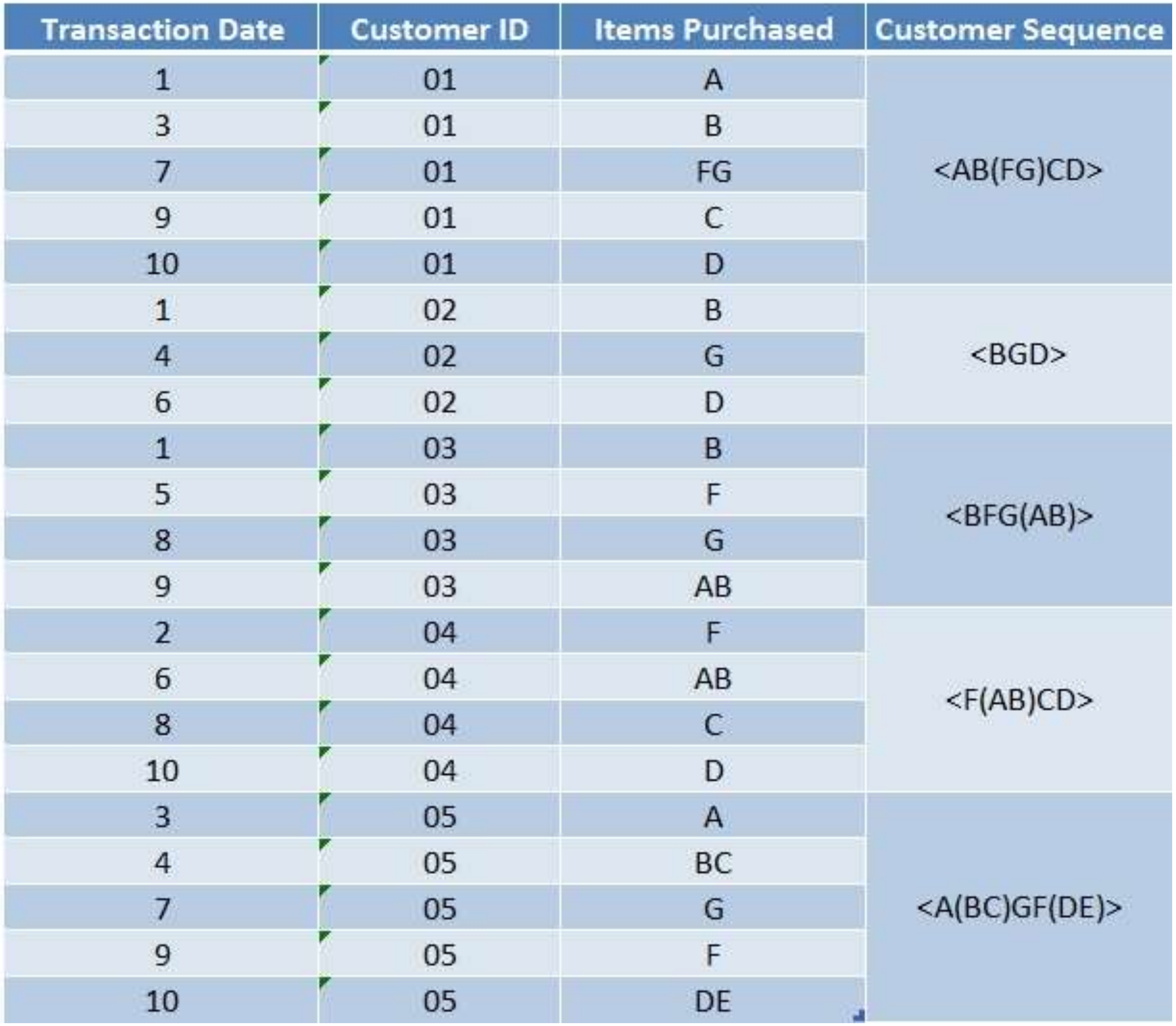
Now, using Apriori Pruning (discarding supersequences of infrequent sequences of length 1), supersequences of length 2 are constructed as candidates. This process repeats till no more candidates or no frequent sequence can be found. Thus, this process outputs all the frequent sequences from the dataset, starting from length 1.

While this algorithm reduces the search space by Apriori Pruning, it still scans the database multiple times and can generate a large number of candidates if the minimum support is less.

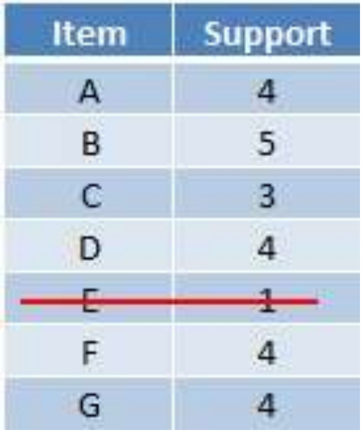
GSP example



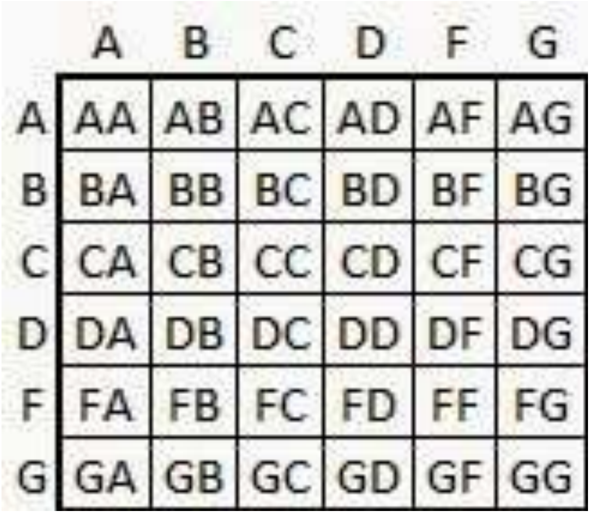
Transaction Date	Customer ID	Items Purchased
1	01	A
1	02	B
1	03	B
2	04	F
3	01	B
3	05	A
4	02	G
4	05	BC
5	03	F
6	04	AB
6	02	D
7	01	FG
7	05	G
8	04	C
8	03	G
9	05	F
9	01	C
9	03	AB
10	01	D
10	05	DE
10	04	D



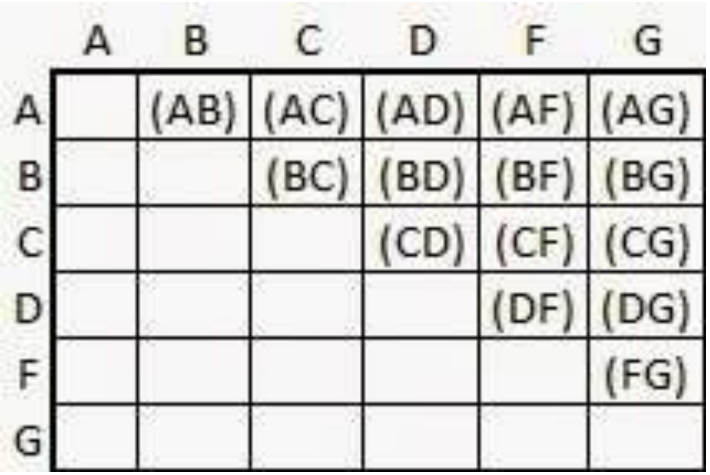
Transaction Date	Customer ID	Items Purchased	Customer Sequence
1	01	A	<AB(FG)CD>
3	01	B	
7	01	FG	
9	01	C	
10	01	D	
1	02	B	<BGD>
4	02	G	
6	02	D	
1	03	B	<BFG(AB)>
5	03	F	
8	03	G	
9	03	AB	
2	04	F	<F(AB)CD>
6	04	AB	
8	04	C	
10	04	D	
3	05	A	<A(BC)GF(DE)>
4	05	BC	
7	05	G	
9	05	F	
10	05	DE	



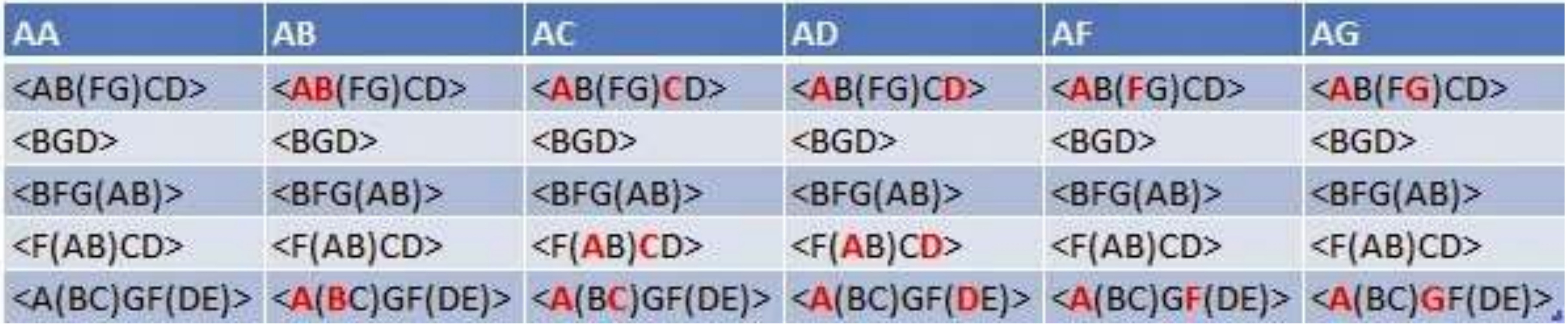
Item	Support
A	4
B	5
C	3
D	4
~~E~~	~~1~~
F	4
G	4



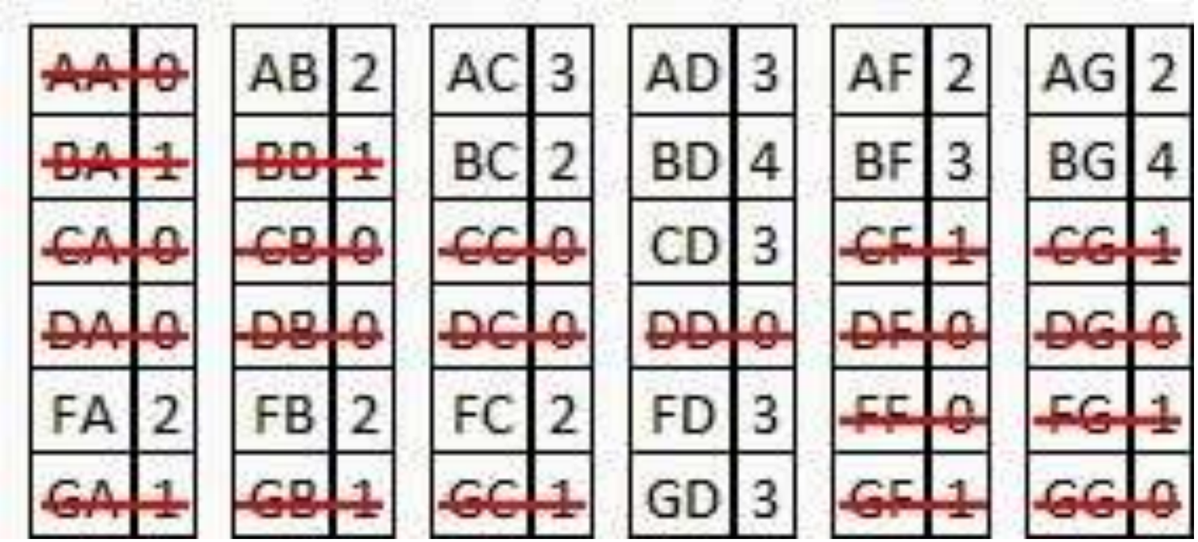
	A	B	C	D	F	G
A	AA	AB	AC	AD	AF	AG
B	BA	BB	BC	BD	BF	BG
C	CA	CB	CC	CD	CF	CG
D	DA	DB	DC	DD	DF	DG
F	FA	FB	FC	FD	FF	FG
G	GA	GB	GC	GD	GF	GG



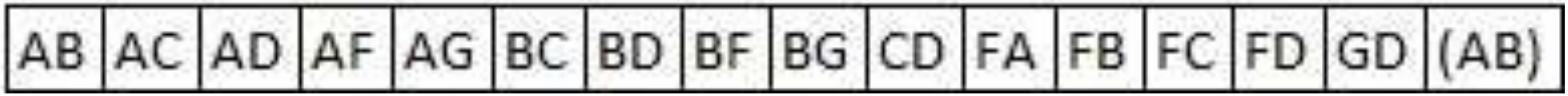
	A	B	C	D	F	G
A		(AB)	(AC)	(AD)	(AF)	(AG)
B			(BC)	(BD)	(BF)	(BG)
C				(CD)	(CF)	(CG)
D					(DF)	(DG)
F						(FG)
G						



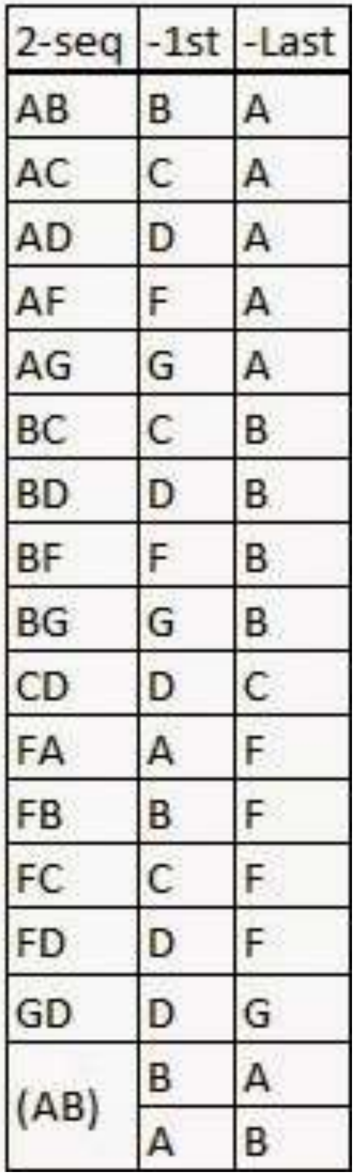
AA	AB	AC	AD	AF	AG
<AB(FG)CD>	<AB(FG)CD>	<AB(FG)CD>	<AB(FG)CD>	<AB(FG)CD>	<AB(FG)CD>
<BGD>	<BGD>	<BGD>	<BGD>	<BGD>	<BGD>
<BFG(AB)>	<BFG(AB)>	<BFG(AB)>	<BFG(AB)>	<BFG(AB)>	<BFG(AB)>
<F(AB)CD>	<F(AB)CD>	<F(AB)CD>	<F(AB)CD>	<F(AB)CD>	<F(AB)CD>
<A(BC)GF(DE)>	<A(BC)GF(DE)>	<A(BC)GF(DE)>	<A(BC)GF(DE)>	<A(BC)GF(DE)>	<A(BC)GF(DE)>



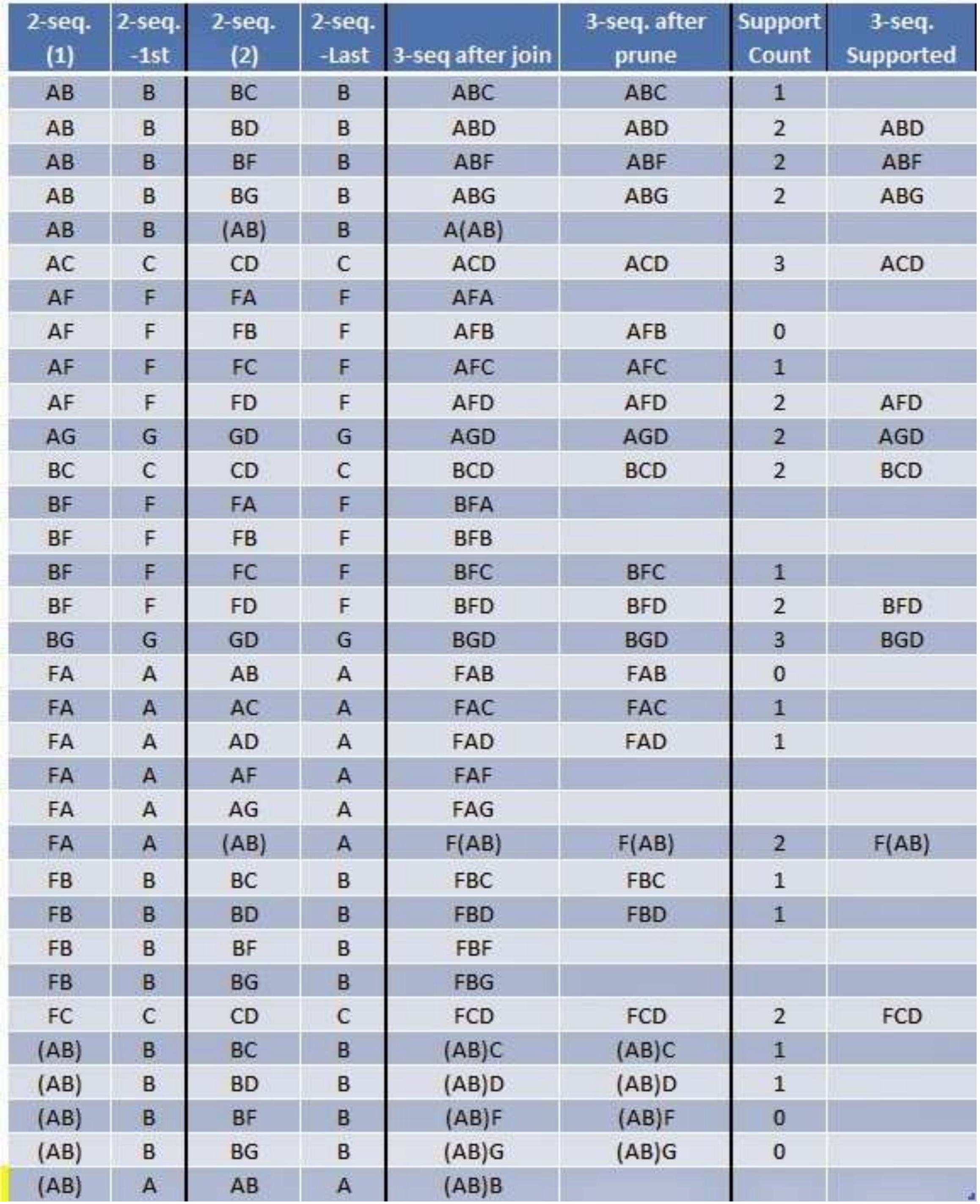
~~AA~~	~~0~~	AB	2	AC	3	AD	3	AF	2	AG	2
~~BA~~	~~1~~	~~BB~~	~~1~~	BC	2	BD	4	BF	3	BG	4
~~CA~~	~~0~~	~~CB~~	~~0~~	~~CC~~	~~0~~	CD	3	~~CF~~	~~1~~	~~CG~~	~~1~~
~~DA~~	~~0~~	~~DB~~	~~0~~	~~DC~~	~~0~~	~~DD~~	~~0~~	~~DF~~	~~0~~	~~DG~~	~~0~~
FA	2	FB	2	FC	2	FD	3	~~FF~~	~~0~~	~~FG~~	~~1~~
~~GA~~	~~1~~	~~GB~~	~~1~~	~~GC~~	~~1~~	GD	3	~~GF~~	~~1~~	~~GG~~	~~0~~



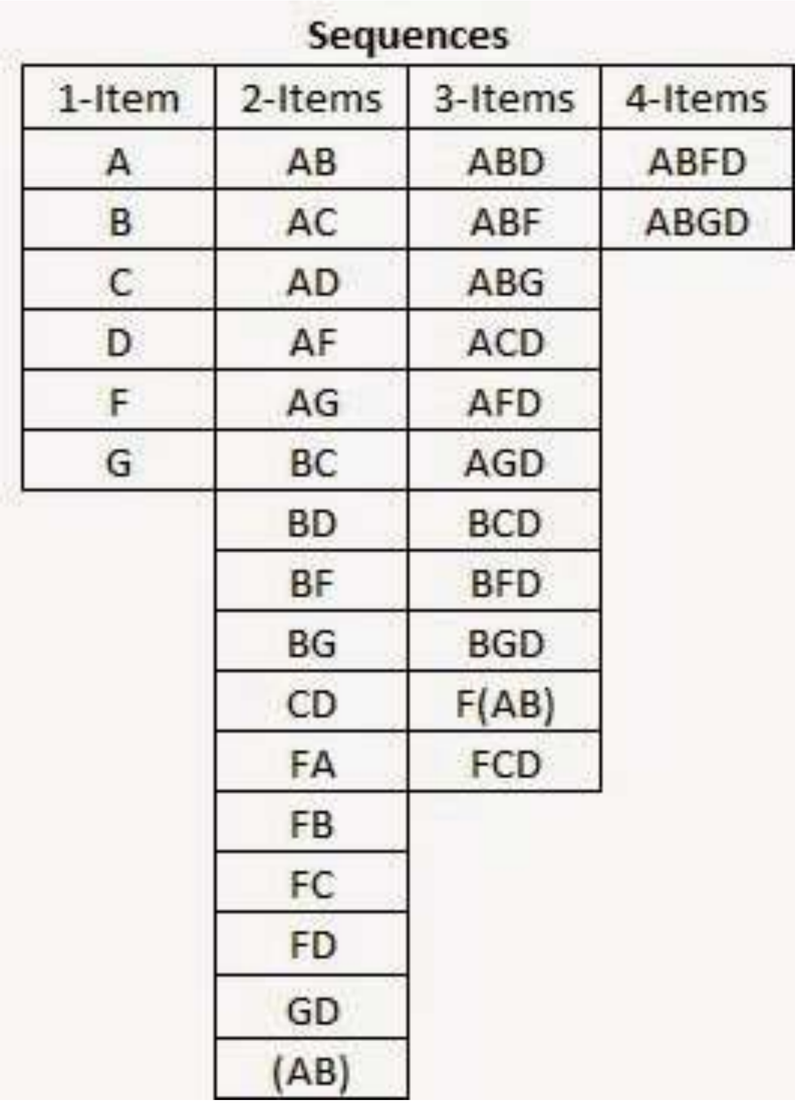
AB	AC	AD	AF	AG	BC	BD	BF	BG	CD	FA	FB	FC	FD	GD	(AB)



2-seq	-1st	-Last
AB	B	A
AC	C	A
AD	D	A
AF	F	A
AG	G	A
BC	C	B
BD	D	B
BF	F	B
BG	G	B
CD	D	C
FA	A	F
FB	B	F
FC	C	F
FD	D	F
GD	D	G
(AB)	B	A
	A	B



2-seq. (1)	2-seq. -1st	2-seq. (2)	2-seq. -Last	3-seq after join	3-seq. after prune	Support Count	3-seq. Supported
AB	B	BC	B	ABC	ABC	1	
AB	B	BD	B	ABD	ABD	2	ABD
AB	B	BF	B	ABF	ABF	2	ABF
AB	B	BG	B	ABG	ABG	2	ABG
AB	B	(AB)	B	A(AB)			
AC	C	CD	C	ACD	ACD	3	ACD
AF	F	FA	F	AFA			
AF	F	FB	F	AFB	AFB	0	
AF	F	FC	F	AFC	AFC	1	
AF	F	FD	F	AFD	AFD	2	AFD
AG	G	GD	G	AGD	AGD	2	AGD
BC	C	CD	C	BCD	BCD	2	BCD
BF	F	FA	F	BFA			
BF	F	FB	F	BFB			
BF	F	FC	F	BFC	BFC	1	
BF	F	FD	F	BFD	BFD	2	BFD
BG	G	GD	G	BGD	BGD	3	BGD
FA	A	AB	A	FAB	FAB	0	
FA	A	AC	A	FAC	FAC	1	
FA	A	AD	A	FAD	FAD	1	
FA	A	AF	A	FAF			
FA	A	AG	A	FAG			
FA	A	(AB)	A	F(AB)	F(AB)	2	F(AB)
FB	B	BC	B	FBC	FBC	1	
FB	B	BD	B	FBD	FBD	1	
FB	B	BF	B	FBF			
FB	B	BG	B	FBG			
FC	C	CD	C	FCD	FCD	2	FCD
(AB)	B	BC	B	(AB)C	(AB)C	1	
(AB)	B	BD	B	(AB)D	(AB)D	1	
(AB)	B	BF	B	(AB)F	(AB)F	0	
(AB)	B	BG	B	(AB)G	(AB)G	0	
(AB)	A	AB	A	(AB)B			



Sequences

1-Item	2-Items	3-Items	4-Items
A	AB	ABD	ABFD
B	AC	ABF	ABGD
C	AD	ABG	
D	AF	ACD	
F	AG	AFD	
G	BC	AGD	
	BD	BCD	
	BF	BFD	
	BG	BGD	
	CD	F(AB)	
	FA	FCD	
	FB		
	FC		
	FD		
	GD		
	(AB)		

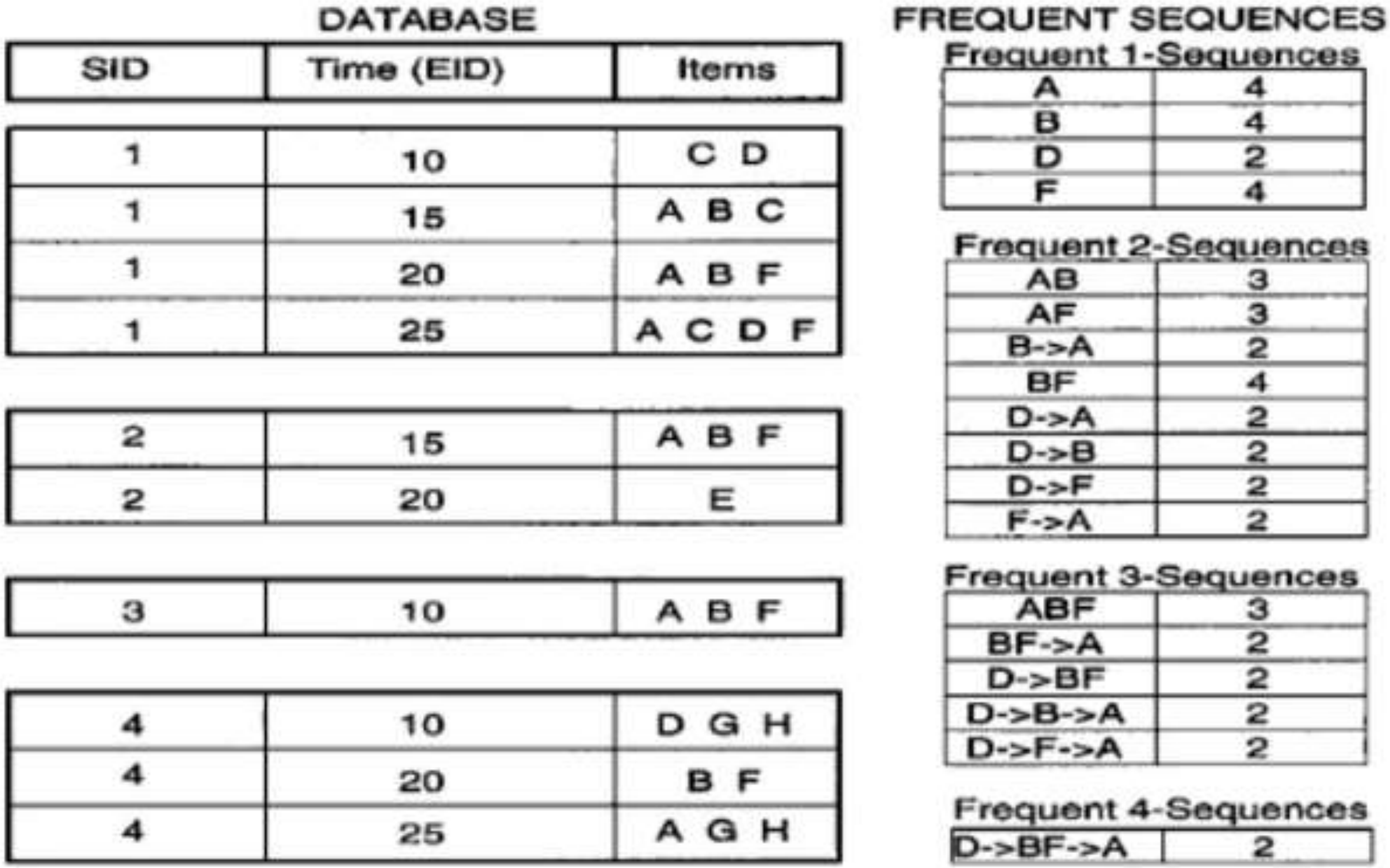
Reference: http://simpledatamining.blogspot.com/2015/03/generalized-sequential-pattern-gsp.html

SPADE (Sequential Pattern Discovery Using Equivalence Class)

This Sequence Pattern Mining algorithm identifies each element in each sequence in a dataset with a Sequence ID (SID) and the Element ID (EID). Candidates of length 1 are constructed, and the SIDs and EIDs of all elements where they occur are noted.

Candidates of higher length are constructed with the property that the Element IDs of all the elements in the candidate should be in increasing order. This algorithm also facilitates joins (for example, if a set of SIDs and EIDs are identified for candidates ab and ba, then SIDs and EIDs can be obtained for candidate aba through joins.

SPADE (Sequential Pattern Discovery Using Equivalence Class)



DATABASE

SID	Time (EID)	Items
1	10	C D
1	15	A B C
1	20	A B F
1	25	A C D F
2	15	A B F
2	20	E
3	10	A B F
4	10	D G H
4	20	B F
4	25	A G H

FREQUENT SEQUENCES

Frequent 1-Sequences

A	4
B	4
D	2
F	4

Frequent 2-Sequences

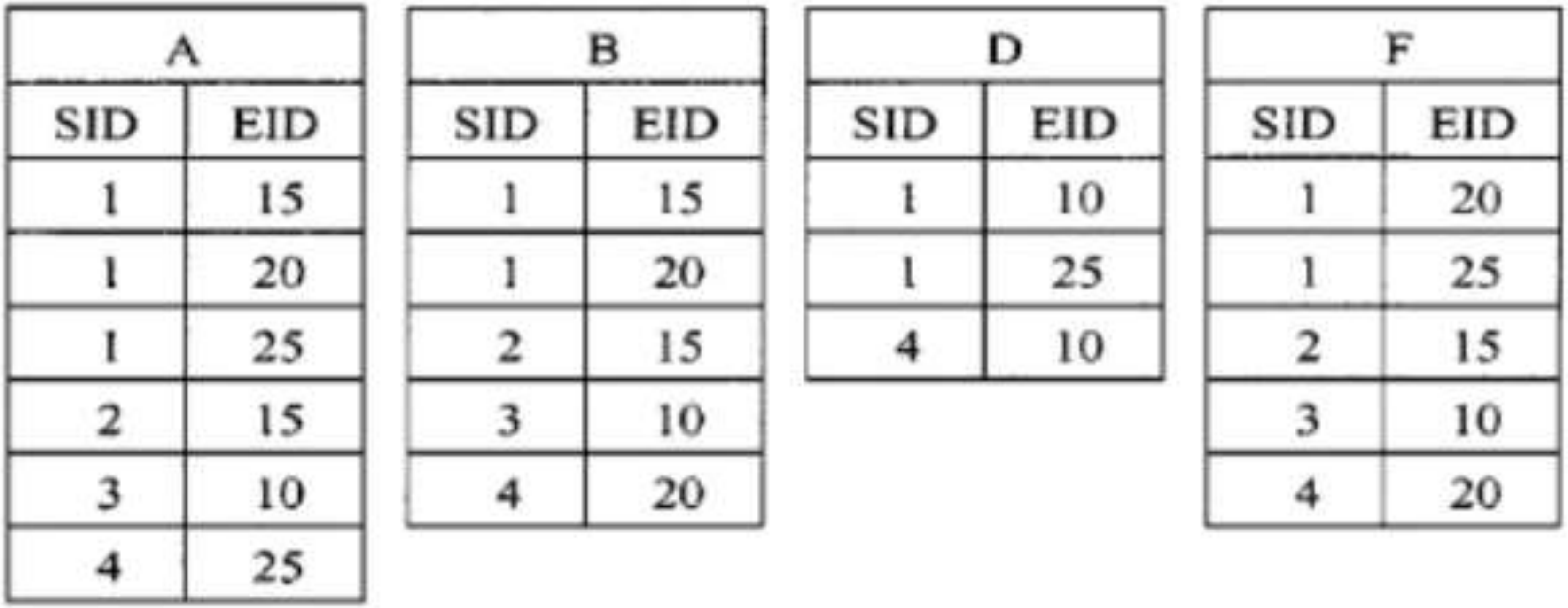
AB	3
AF	3
B->A	2
BF	4
D->A	2
D->B	2
D->F	2
F->A	2

Frequent 3-Sequences

ABF	3
BF->A	2
D->BF	2
D->B->A	2
D->F->A	2

Frequent 4-Sequences

D->BF->A	2



A	
SID	EID
1	15
1	20
1	25
2	15
3	10
4	25

B	
SID	EID
1	15
1	20
2	15
3	10
4	20

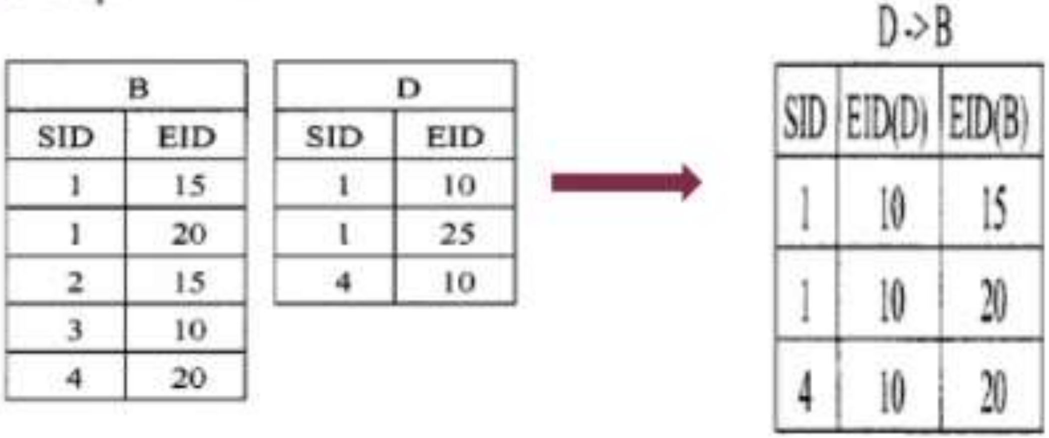
D	
SID	EID
1	10
1	25
4	10

F	
SID	EID
1	20
1	25
2	15
3	10
4	20

Id-lists of the most frequent items (1-sequences)

D->BF->A

- Step 1: D->B



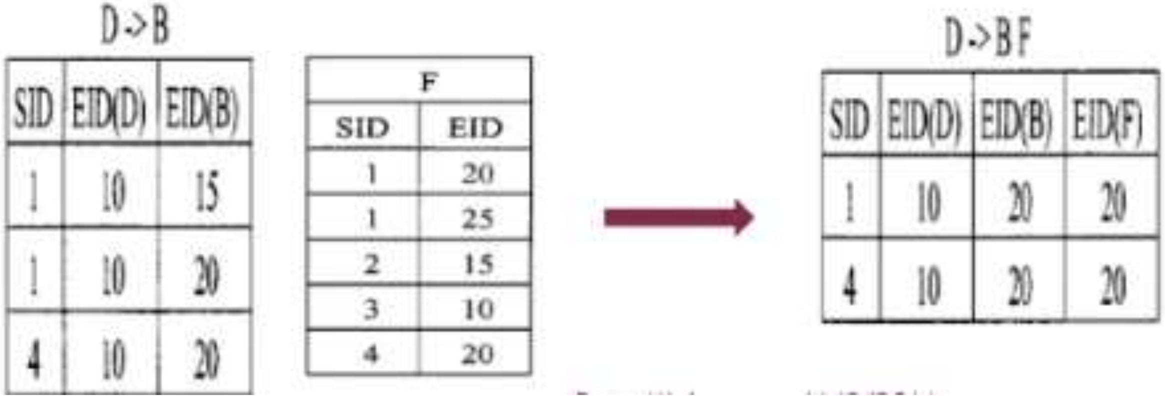
B	
SID	EID
1	15
1	20
2	15
3	10
4	20

D	
SID	EID
1	10
1	25
4	10

D->B

SID	EID(D)	EID(B)
1	10	15
1	10	20
4	10	20

- Step 2: D->BF



D->B

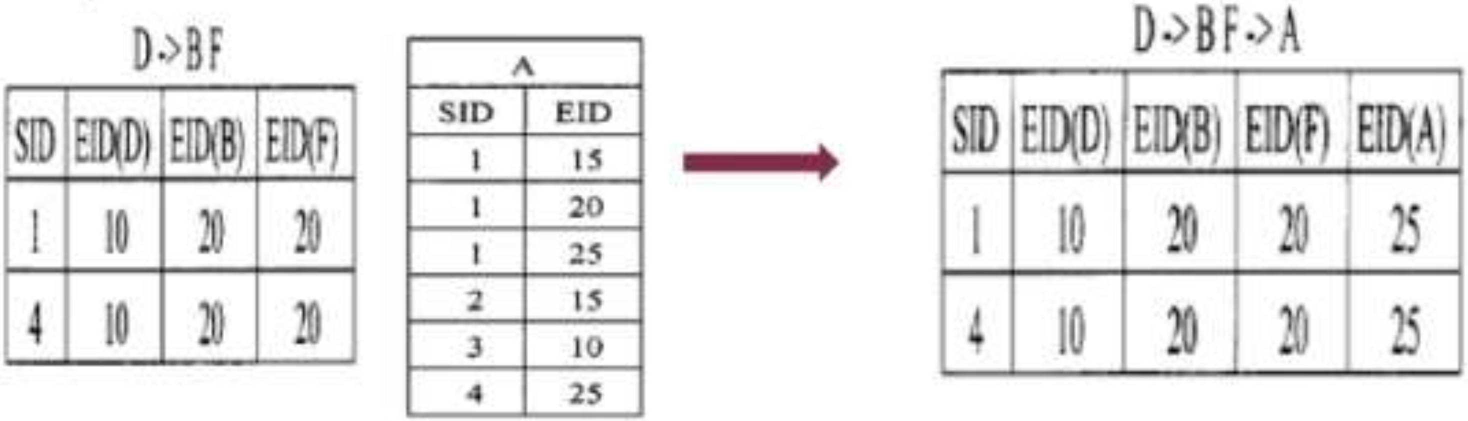
SID	EID(D)	EID(B)
1	10	15
1	10	20
4	10	20

F	
SID	EID
1	20
1	25
2	15
3	10
4	20

D->BF

SID	EID(D)	EID(B)	EID(F)
1	10	20	20
4	10	20	20

- D->BF->A
  - Step 3 : D->BF->A



D->BF

SID	EID(D)	EID(B)	EID(F)
1	10	20	20
4	10	20	20

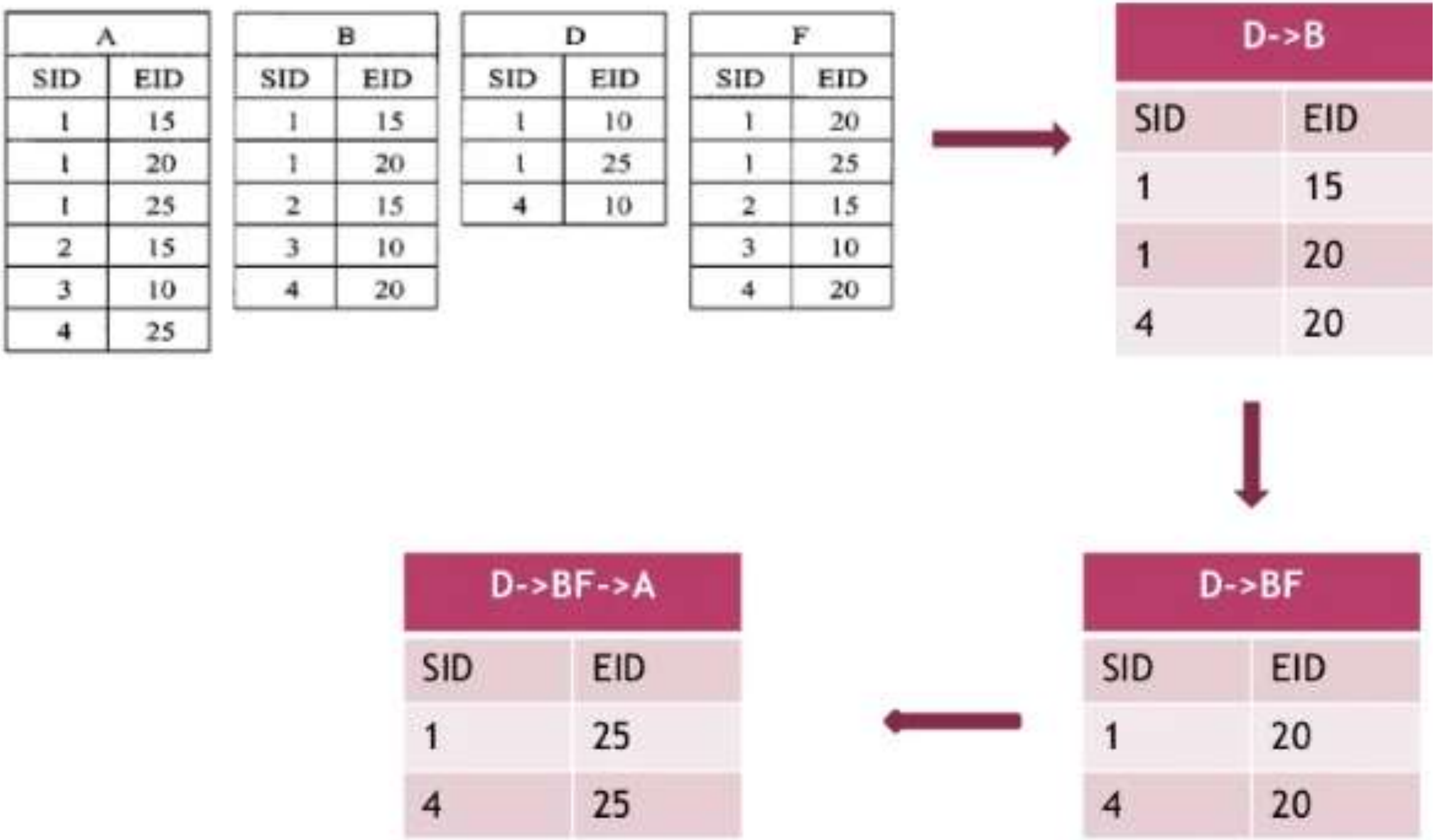
A

SID	EID
1	15
1	20
1	25
2	15
3	10
4	25

D->BF->A

SID	EID(D)	EID(B)	EID(F)	EID(A)
1	10	20	20	25
4	10	20	20	25

- D->BF->A (space-efficient id-list joins)



A	
SID	EID
1	15
1	20
1	25
2	15
3	10
4	25

B	
SID	EID
1	15
1	20
2	15
3	10
4	20

D	
SID	EID
1	10
1	25
4	10

F	
SID	EID
1	20
1	25
2	15
3	10
4	20

D->B	
SID	EID
1	15
1	20
4	20

D->BF	
SID	EID
1	20
4	20

D->BF->A	
SID	EID
1	25
4	25

Reference: https://philippe-fournier-viger.com/spmf/SPADE.pdf

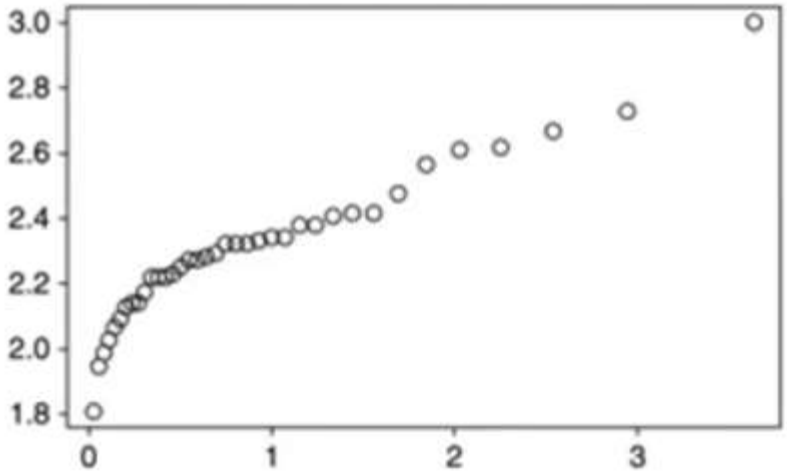
DATA SUMMARIZATION TECHNIQUES

1. Histogram
2. Quantile Plot
3. Quantile- Quantile Plot
4. Scatter Plot
5. Box Plot

1. Quantile Plot

Univariate distribution

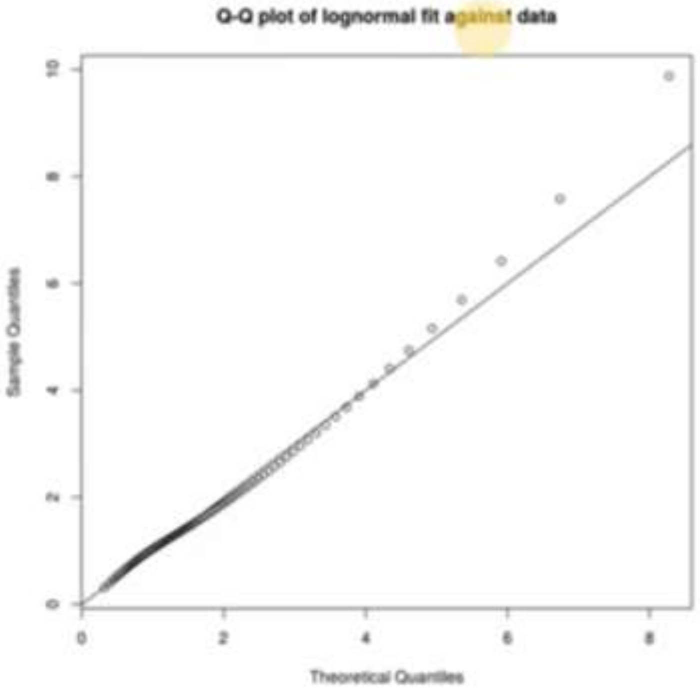
It is used when data base has variety of information.



2. Quantile-Quantile Plot

Q-Q(quantile-quantile) plots play a very vital role to graphically analyze and compare two probability distributions by plotting their quantiles against each other.

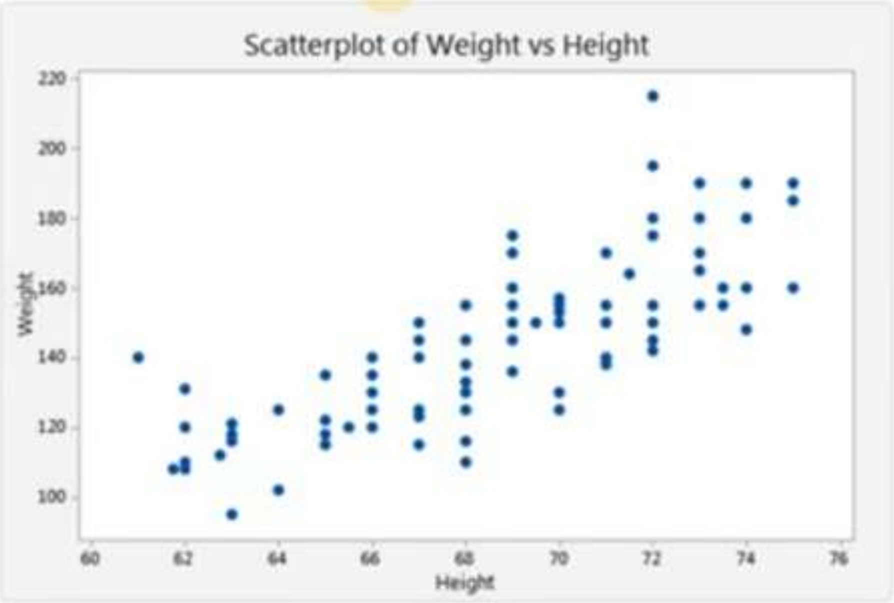
If the two distributions which we are comparing are exactly equal then the points on the Q-Q plot will perfectly lie on a straight line y = x.



3. Scatter Plot

A scatter plot is a type of data visualization that shows the relationship between different variables. This data is shown by placing various data points between an x- and y-axis.

Essentially, each of these data points looks "scattered" around the graph, giving this type of data visualization its name.

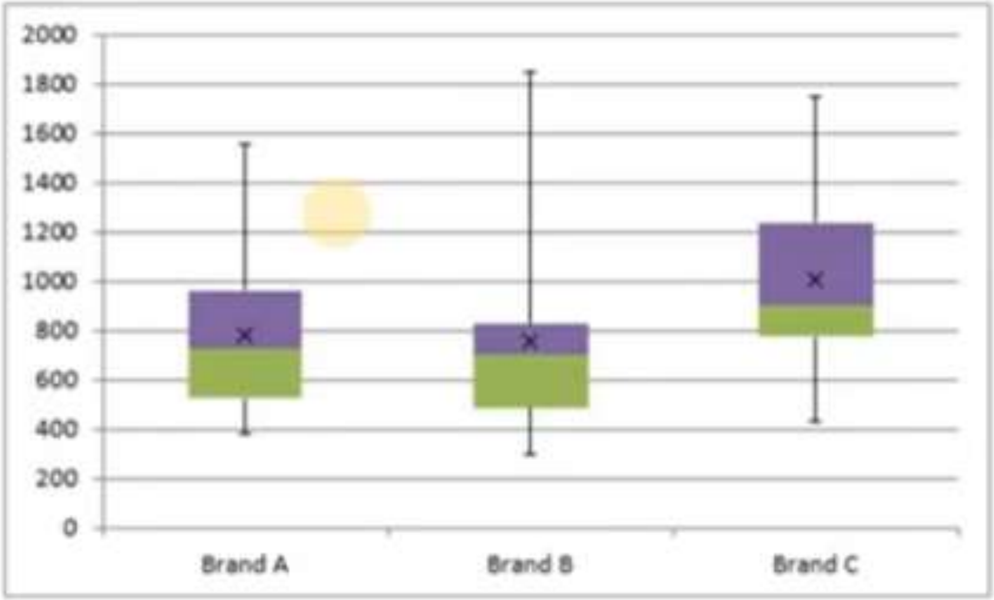


4. Box plot

A boxplot is a graph that gives you a good indication of how the values in the data are spread out.

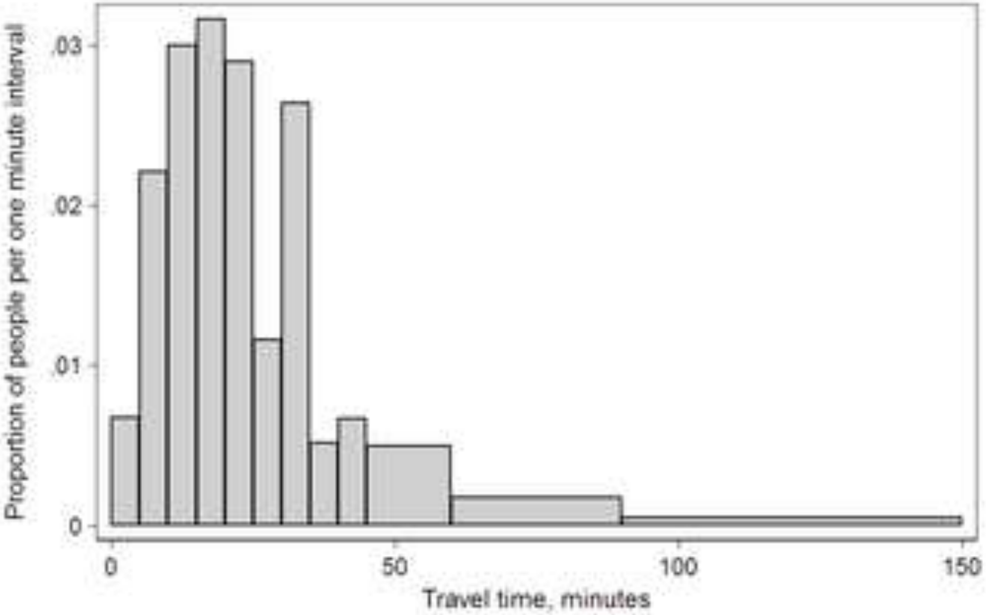
A box plot is a way of summarizing a set of data measured on an interval scale.

It is often used in explanatory data analysis. This type of graph is used to show the shape of the distribution, its central value, and its variability.



5. Histogram

A histogram is **a bar graph-like representation of data that buckets a range of classes into columns along the horizontal x-axis**. The vertical y-axis represents the number count or percentage of occurrences in the data for each column. Columns can be used to visualize patterns of data distributions.



Time Series

A time series is…

❑A set of data depending on the time

❑A series of values over a period of time

❑Collection of magnitudes belonging to different time periods of some variable or composite of variables such as production of steel, per capita income, gross national income, price of tobacco, index of industrial production.

•In time series, time act as an independent variable to estimate dependent variables.

Mathematical presentation of Time Series

■ A time series is a set of observation taken at specified times, usually at 'equal intervals'.

Mathematically a time series is defined by the values Y_1, Y2...of a variable Y at times t1, t2....

Thus,

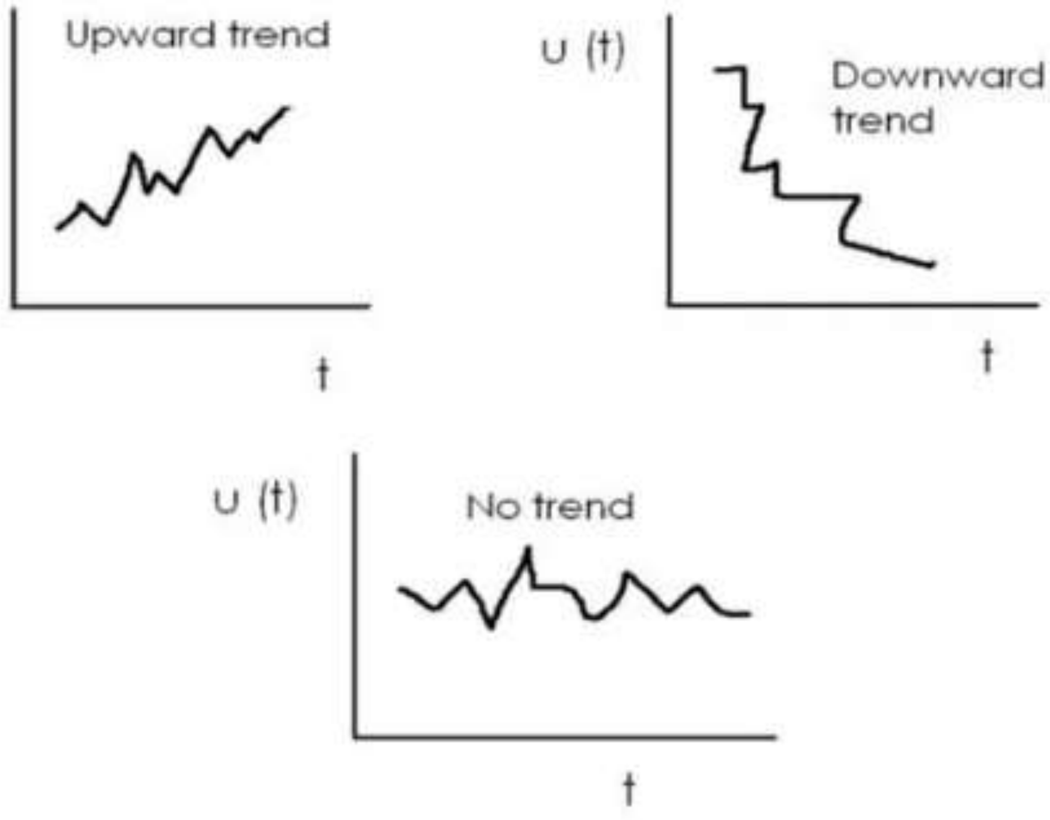
Y= F(t)

Types of Components

1. Secular Trend or Trend
2. Seasonal Variations/Fluctuations
3. Cyclical Variations/Fluctuations
4. Irregular Variations/Movements

SECULAR TREND OR TREND

- The general tendency of the data to grow or decline over a long period of time.
- The forces which are constant over a long period (or even if they vary they do so very gradually) produce the trend. For e.g., population change, technological progress, improvement in business organization, better medical facility etc.



Examples:
Downward trend-declining death rate
Upward trend-population growth
Mathematically trend may be Linear or non-linear

PURPOSE OF MEASURING TREND

Knowledge of past behavior
Estimation
Study of other components

SEASONAL VARIATIONS/FLUCTUATIONS

The component responsible for the regular rise or fall (fluctuations) in the time series during a period not more than 1 year.

■Fluctuations occur in regular sequence (periodical)

The period being a year, a month, a week, a day, or even a fraction of the day, an hour etc.

■Term "SEASONAL" is meant to include any kind of variation which is of periodic nature and whose repeating cycles are of relatively short duration.

The factors that cause seasonal variations are: (a) Climate & weather condition, (b) Customs traditions & habits

PURPOSE OF MEASURING SEASONAL VARIATIONS

Analysis of past behavior of the series

Forecasting the short time fluctuations

Elimination of the seasonal variations for measuring cyclic variations

EXAMPLES OF SEASONAL VARIATIONS

- Crops are sown and harvested at certain times every year and the demand for the labour growing up during sowing and harvesting seasons.
- Demands for woolen clothes goes up in winter
- Price increases during festivals
- Withdraws from banks are heavy during first week of the month.

CYCLIC VARIATIONS

- Cycle refers to recurrent variations in time series

Cyclical variations usually last longer than a year

- Cyclic fluctuations/variations are long term movements that represent consistently recurring rises and declines in activity.

Purpose

Measures of past cyclical behavior

Forecasting

Useful in formulating policies in business

IRREGULAR VARIATIONS

■ Also called erratic, random, or "accidental" variations

■ Do not repeat in a definite pattern

Strikes, fire, wars, famines, floods, earthquakes

■ unpredictable

CHARACTERISTICS

Irregular & unpredictable

No definite pattern

Short period of time

No Statistical technique