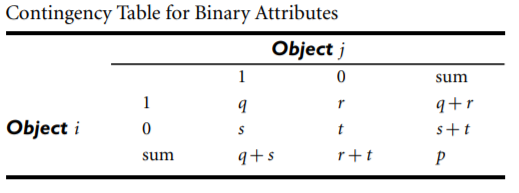
### Chapter-2

### Getting to know your Data

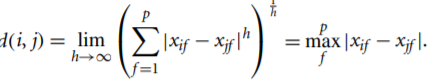
* 1. Data Objects:
     1. Samples, examples, instances, data points, objects.
  2. Attributes:
     1. Dimension, feature, variable
     2. Single attribute data---> univariate
     3. Multiple attributes data---> multivariate
     4. NOMINAL
        1. Categorical, example occupation, customer\_id etc(can be numeric but no numeric operations are sensible on them).
     5. BINARY
        1. They are nominal with 2 values or states
        2. Binary if 2 values are true/false.
        3. Symmetric (if they have same value[in real life sense] like gender)
        4. Asymmetric (if one has more value like HIV positive vs HIV negative)
     6. ORDINAL
        1. Have ranking but no values like small,medium,large
        2. Median can be found but not mean.
     7. NUMERIC
        1. Have numeric values like int, real no.
        2. Interval scaled
           1. Example year (cant be ratio-ed like double the year)
           2. Difference matters
        3. Ratio scaled
           1. Example year\_of\_experience
           2. Ration matters
     8. Discrete vs continuous values
        1. Discrete means--- finite OR countably infinite.
        2. Continuous means --- not Discrete.
  3. Basic statistical description of Data
     1. Needed to understand noise and outliers
     2. Measures of central tendency
        1. Intuitively-->given an attribute where does the most of its values fall.
        2. Mean, median, mode, midrange
        3. Mean-->trimmed mean(bcz highly sensitive to outliers)
        4. With one mode-->unimodal else multimodal,bimodal, trimodal etc.
        5. Mean - mode = 3 \* (mean - median) ---unimodal
        6. midrange=(max-min)/2
        7. Unimodal, perfect symmetric--->mean=median=mode
        8. Positively skewed(median is more than mode) else vice versa then negatively skewed
     3. Dispersion of data
        1. Intuitively-->how is the data spread
        2. Helps in finding outliers
        3. Range=max-min
        4. Quantile(points dividing data into essentially equal parts)
           1. 2 quantile point will be median
           2. 4 quantile = **quartile**(3 points dividing in 4 quarters)

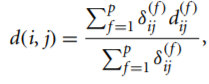
**I**nter **Q**uartile **R**ange = Q3-Q1

* + - * 1. 100 quantile = **percentile**
      1. Five-Number Summary, Boxplots, and Outliers :
         1. Five-Number summary = min,Q1,median,Q3,max.
         2. Boxplots ***[[[[[[[[[[[TO-DO]]]]]]]]]]]]***
      2. Variance and Standard Deviation
         1. sigma^2 = variance = 1/n \* sum(x\_i-mean)^2.
         2. Due to chebsevy’s inequality it is shown that SD shows good data spread so love it pls!
    1. Graphic Displays of Basic Statistical Descriptions of Data
       1. Quantile plot
          1. X\_i vs f\_i where f\_i = (i-0.5)/N.
          2. f\_i is no of data points less than x\_i.
          3. 0.25 will be Q1, 0.5 fi will be median, 0.75 will be Q3.
       2. Quantile Quantile plot (q q plot)
          1. Quantile plot of 2 observations of the same attribute.
          2. If m<n i.e. one observation has less points then→ we will find both x\_i and y\_i ‘s f\_i as (i-0.5)/M for both.
          3. Can know if both observations come from the same distribution.
       3. Histograms
          1. Can summarize the distribution of an attribute.
          2. It has buckets or bins of equal size (which has data points values).
          3. Vs the frequency of the points falling in the bucket.
       4. Scatter plots and data correlation
          1. Most effective to know a relationship, pattern or trend b/w any 2 numeric values.
          2. Shows Bivariate data and finds if X,Y are correlated like if anyone implies the other one or the clusters or outliers.
          3. Positive, negative, null correlation.
          4. Positive means pints are from lower left to upper right meaning as X increases, Y also increases.
  1. Data Visualisation
     1. Communicate data clearly and effectively.
     2. Helps in reporting, tracking, managing business operations.
     3. Shows data relations which were not otherwise visible
     4. Pixel-Oriented Visualization Techniques
        1. Can relate multiple attributes.
        2. Dense means high value.
        3. Space filling curve
           1. Hilbert, Gray code, Z curve, circular (dimension are side by side)
     5. Geometric Projection Visualization Techniques
        1. Drawback of pixel oriented is :
           1. Cant visualize dense area in multidimensional subspace.
        2. Scatter plot = 2D. If u want 3D can have different patterns like”+” or different colors for different data points of 3rd dimension.
        3. For more than 4D data SP is not efficient.
        4. Scatter plot matrix
           1. Can handle n dimensional data.
           2. Cannot handle when dimensionality increases.
        5. Parallel coordinates:
           1. One axis for each dimension.
           2. Cannot handle large datasets.
     6. Icon-Based Visualization Techniques
        1. Chernoff faces
           1. 18 dimensions-->assymentric(along y axis)--therefor now 36 dimensions.
        2. Stick figure
           1. 5 piece stick figure
     7. Hierarchical Visualization Techniques
        1. Example: Worlds within worlds-->n-Vision.
           1. Fix some dimensions’s values and take them as origin and plot 2D or 3D etc for rest dimensions and if more required do more nesting.
           2. Outer dimensions can be changed hence the inner too.
        2. Tree-maps
           1. Hierarchical data as a set of nested rectangles.
     8. Visualizing Complex Data and Relations
        1. Complex data like text/social networks.
        2. Tag clouds(eg. size of tags of websites according to popularity etc).
        3. COmplex relation between these datasets.
  2. Measuring data similarity and dissimilarity.
     1. Clustering, outlier analysis. Similarity = 1 if similar else 0 vice versa for dissimilarity of 2 objects.
     2. Data matrix vs dissimilarity matrix
        1. Main memory based clustering & nearest neighbour analysis uses either of the two data structures.
        2. Data matrix (object by attribute structure) (two mode)
           1. For n samples or feature vector→ n\*p vector where p is attributes.
           2. Data matrix can be transformed to dissimilarity matrix before applying algo like clustering.
        3. Dissimilarity matrix (One mode)
           1. has dissimilarities of objects therefor n\*n matrix.
           2. And it is symmetric. && d(i,i) = 0.
           3. sim(i,j) = 1-d(i,j)
     3. Proximity Measures for Nominal Attributes:
        1. d(i,j)=(p-m)/m where p = no of attributes, m = attributes matching for two objects.
     4. Proximity Measures for Binary Attributes:



* + - 1. Symmetric binary dissimilarity(symmetric because of both attributes have same value)
         1. d(i,j) = (r+s)/(q+r+s+t)
      2. Asymmetric binary similarity
         1. sim(i,j) = q/(q+r+s) here we are neglecting t n bcz it has not much significant like both HIV negative.
         2. Above is called as **jaccard coefficient\*\*.**
    1. Dissimilarity of Numeric Data: Minkowski Distance
       1. Euclidean, manhattan, minkowski distances.
       2. May use normalising before finding dissimilarity.
       3. Euclidean dist b/w x\_i and x\_j having p attributes.
       4. Manhattan dist :
          1. 
       5. Measures that satisfy foll. Properties are = **metric.**
          1. Non-negativity:d(i,j) >=0
          2. Identity of indiscernibles: d(i,i)=0
          3. Symmetry: d(i,j)=d(j,i)
          4. Triangle inequality: d(i,j) <=d(i,k)+d(k,j).
          5. Both euclidean & manhattan are **metric.**
       6. Minkowski:
          1. , also called as L\_h norm.
          2. When h=1→ manhattan dist.
          3. When h=2→ euclidean dist.
          4. When h=max or infinity→ supreme dist. Or Chebyshev dist.--> L\_max Uniform norm.
          5. To find supreme dist, we find the attribute p which gives max difference between its values for the two objects.



* + - 1. Weighted dist is also a concept dude.. :)
    1. Proximity Measures for Ordinal Attributes
       1. Assign ranks to each state
       2. Normalize the ranks.
          1. Z\_f = (r\_if-1) / (M\_f-1) where M\_f = total no of states of an attribute. And r\_if is the ranking of i object and f attribute.
       3. Find out dist using any measures used for numeric attributes to find out dissimilarity.
    2. Dissimilarity for Attributes of Mixed Types:
       1. Means finding dissimilarity between objects having different kinds of attributes.
       2. Bring dissimilarity of all attributes to common scal [0.0,1.0]
       3.  where del\_ij = 0 if any one object’s attribute is missing value OR if x\_if = x\_jf = 0 and it is asymmetric binary attribute.\
       4. d\_ij is computed:
          1. Numeric

where h is scanned over all objects for attribute f.

* + - * 1. Nominal, Binary

d\_ij(f) = 0 if x\_if =x\_jf else d\_ij(f) = 1.

* + - * 1. Ordinal

Compute ranks r\_if and z\_if using formula then treat z\_if as numeric.

* + 1. Cosine Similarity:
       1. Document = thousands of attributes, each recording a particular word’s(or phrase’s) freq.
       2. Object = term-freq vector.
          1. Are very long and sparse.
       3. Applications:
          1. Information retrieval, text doc clustering, biological taxonomy.
       4. Can compare documents OR ranking to a particular query vector.
       5. where ||x|| is euclidean norm of vector or simply its length. **x.y** is dot product.
       6. It is **nonmetric** measure.
       7. If attributes = binary→ Tanimoto similarity.
          1.  which is ratio of attributes possessed(value is 1) by x and y to possesed by x or y.