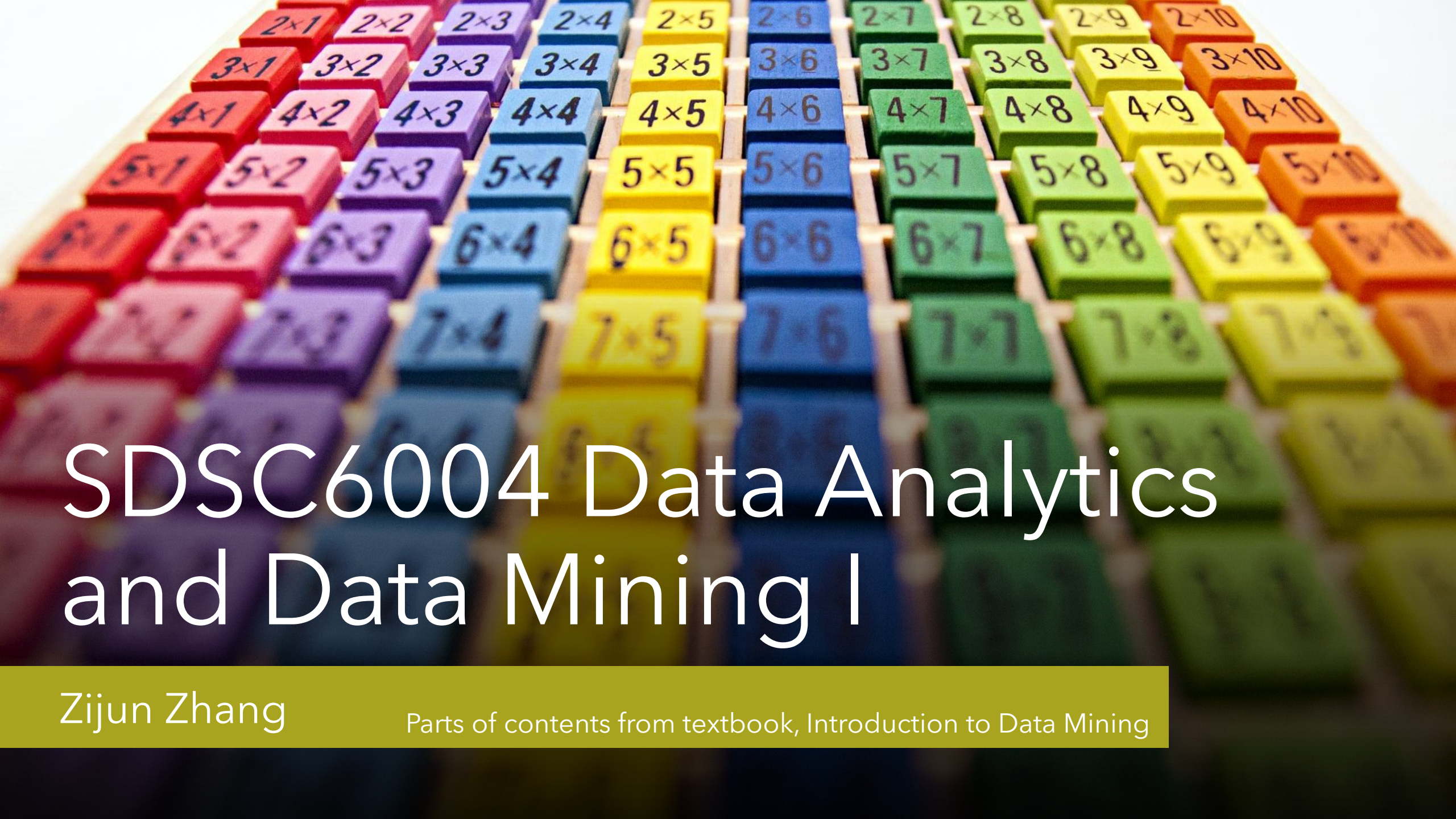
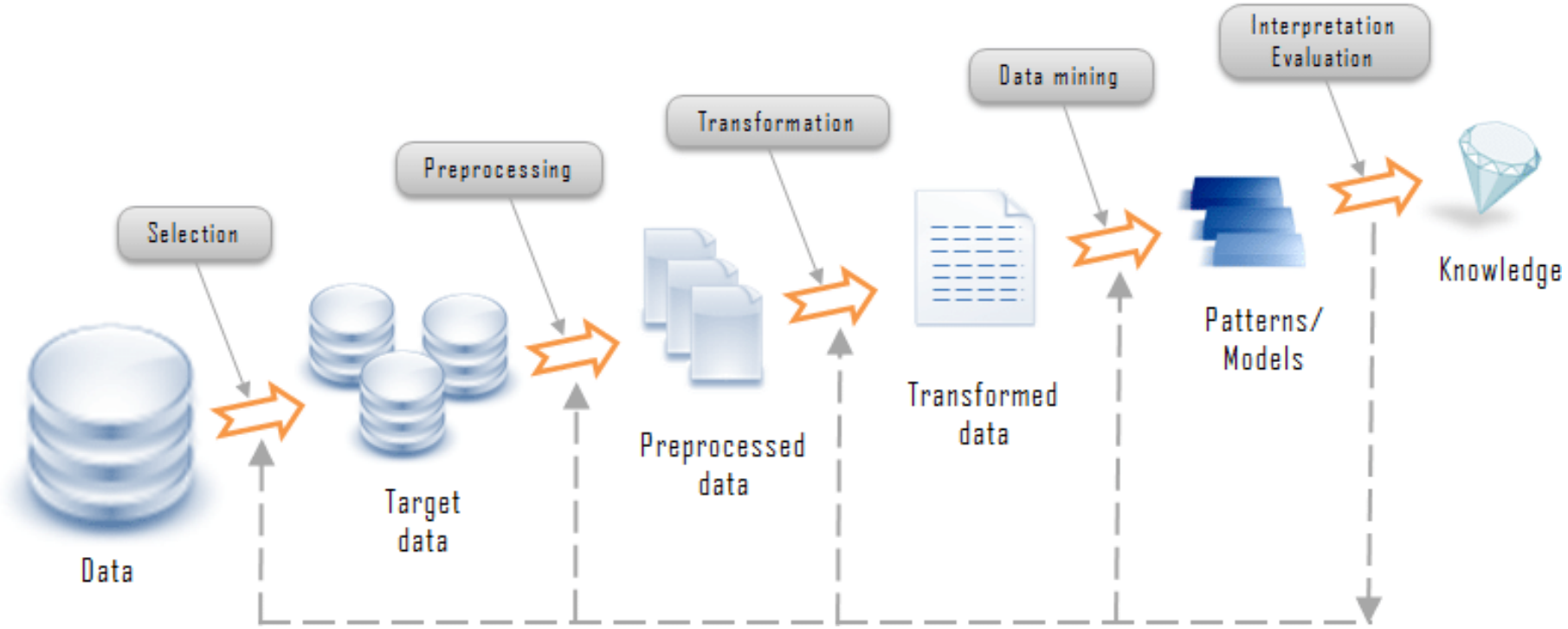


SDSC6004 Data Analytics and Data Mining I

Zijun Zhang

Parts of contents from textbook, Introduction to Data Mining

The Process of Data-driven Modeling



Data Pre-processing and Pre-Analytics

The raw data collected are unstructured or contain a lot of noises.

Data Pre-processing: Handling Missing Data, Correct Invalid Data, and Re-organizing Data

Data Pre-processing and Pre-Analytics

Data Pre-Analytics: Plot different charts and diagrams to understand the data composition, distribution, and properties

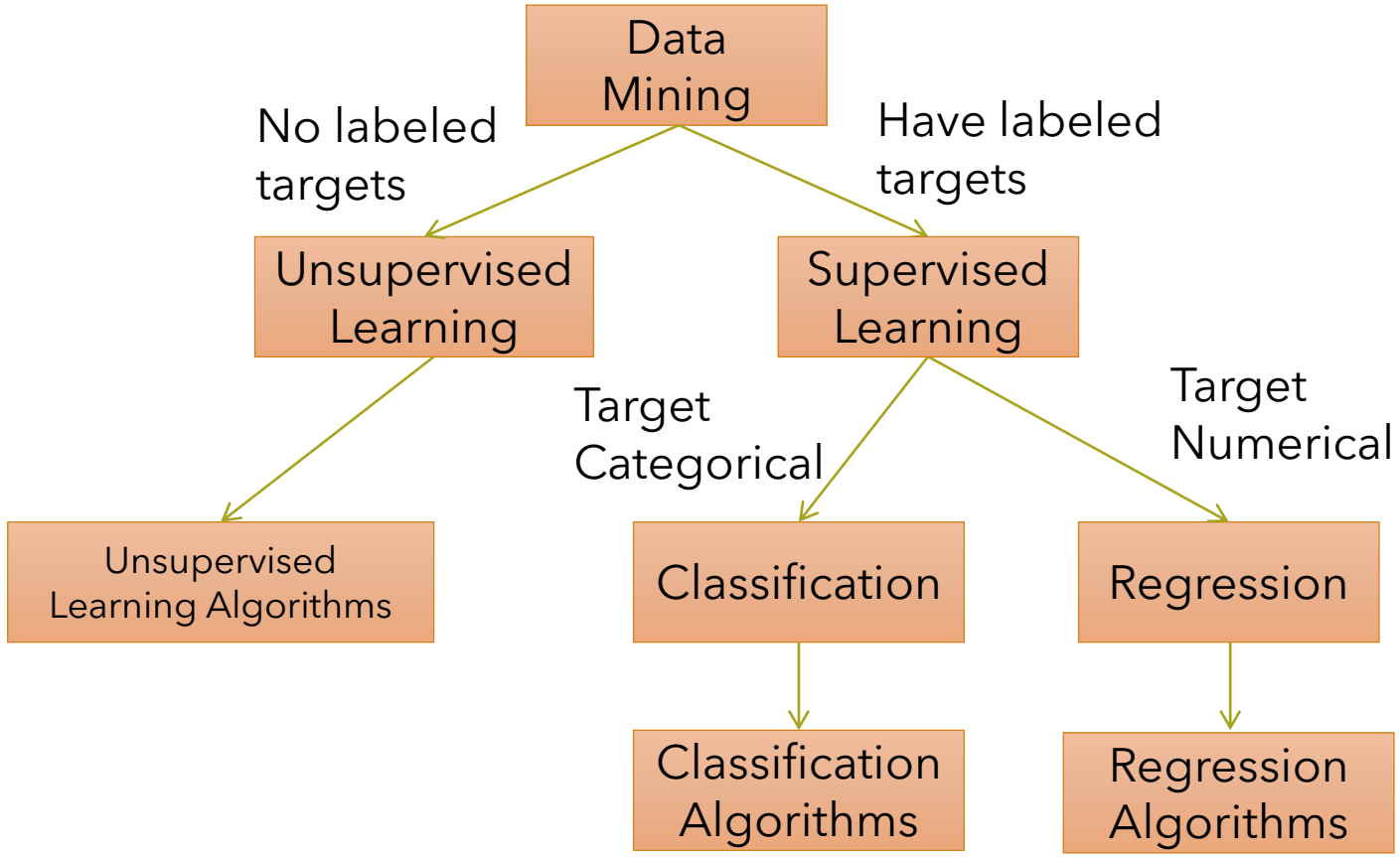
- Compute statistical metrics, mean, variance, etc.
- Spatial Temporal Relationships
- Run chart, Scatter Diagram, Pie chart, Bar chart, etc.
- Dashboard, a display of multiple charts

Data Mining Problems

Un-supervised learning v.s. Supervised learning

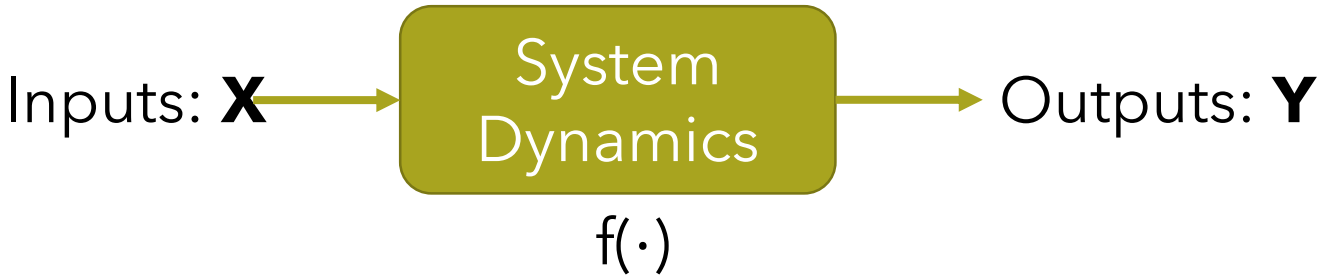
In the problem, whether a labelled target is concerned

Organization of Data Mining Methods



Data-driven Modeling Formulation

A System



Problem: **X**, **Y** are known as data, while f(·) needs to be obtained

Given a loss function L, a data-driven system modeling task typically aims at identifying a non-parametric model $g_A(\cdot)$ via algorithm A by minimizing the $L(g_A(\mathbf{X}), \mathbf{Y})$.

Unsupervised learning in Modeling

Where can we apply unsupervised learning in system modeling problems?

1. Dimension reduction
2. Data Compression

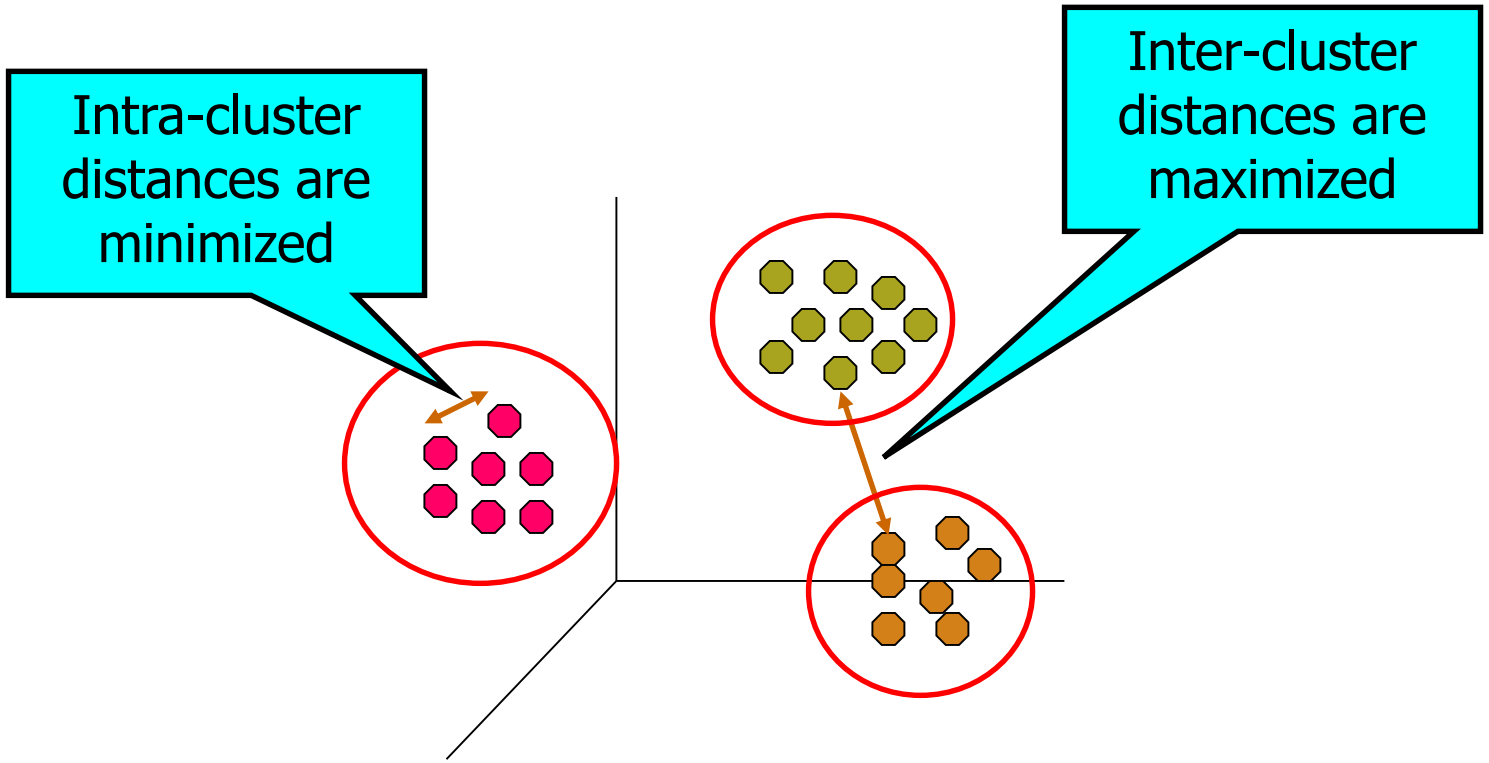
Unsupervised learning: Cluster Analysis

Selected popular algorithms

- K-means
- Hierarchical clustering
- Density-based clustering

What is Cluster Analysis?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

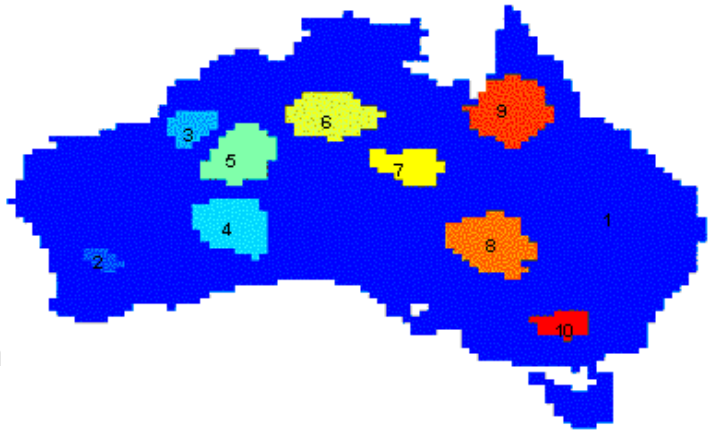


Applications of Cluster Analysis

- **Understanding**
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

- **Summarization**
  - Reduce the size of large data sets

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP



Clustering precipitation in Australia

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

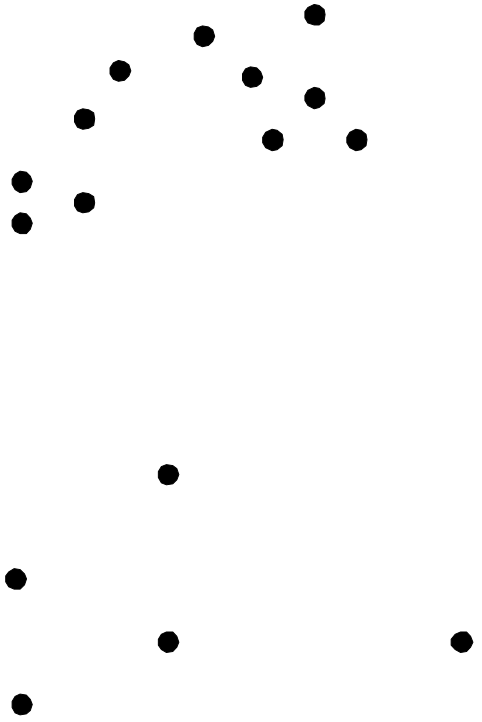
Two Clusters

Four Clusters

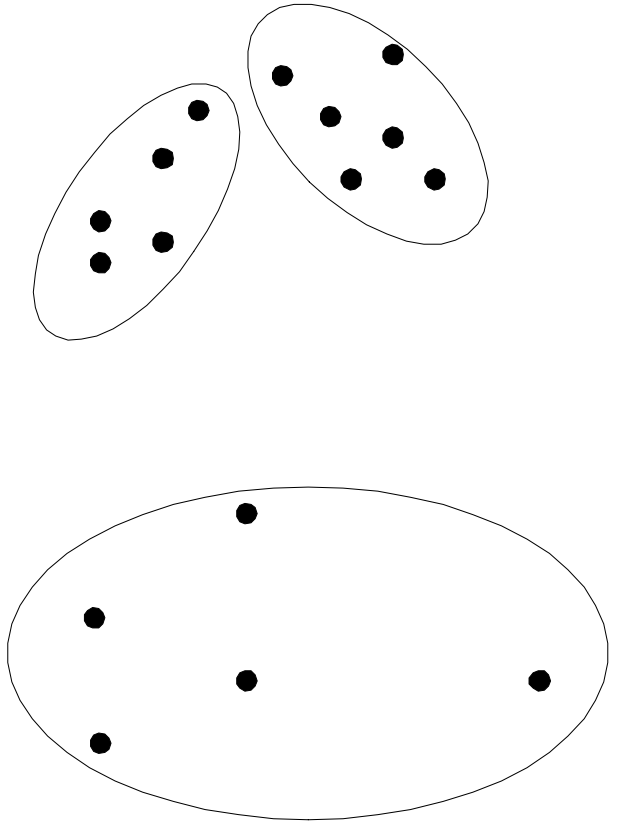
Types of Clustering

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

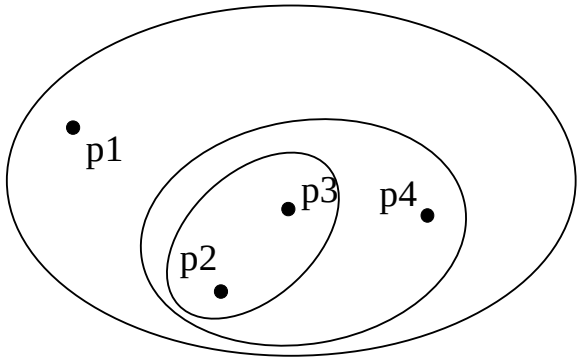


Original Points

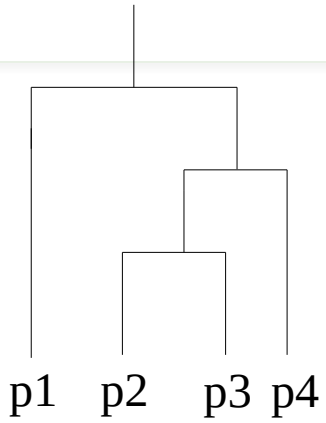


A Partitional Clustering

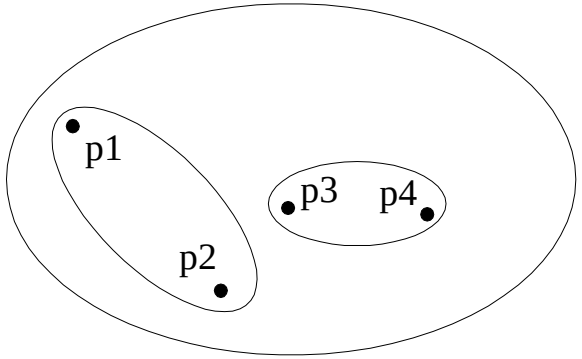
Hierarchical Clustering



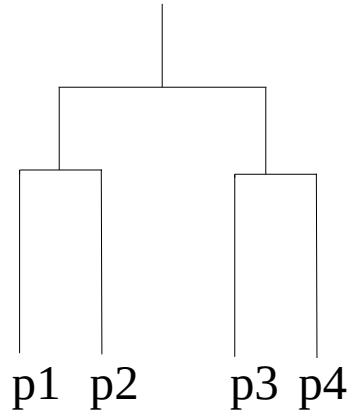
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Map Clustering Problem to a Different Problem

- Map the clustering problem to a different domain and solve a related problem in that domain
  - Proximity matrix defines a weighted graph, where the nodes are the points being clustered, and the weighted edges represent the proximities between points
  - Clustering is equivalent to breaking the graph into connected components, one for each cluster.
  - Want to minimize the edge weight between clusters and maximize the edge weight within clusters

Characteristics of the Input Data Are Important

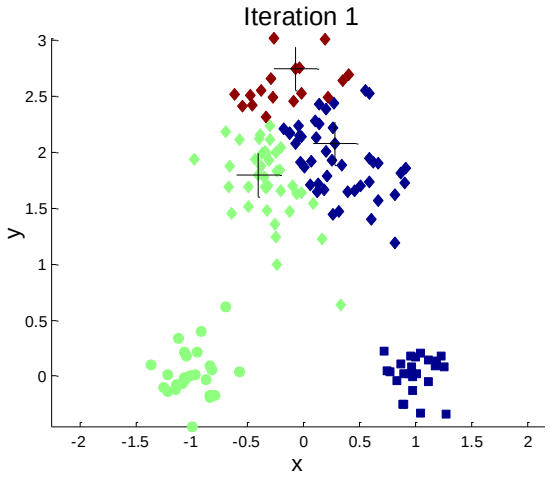
- Type of proximity or density measure
  - Central to clustering
  - Depends on data and application
- Data characteristics that affect proximity and/or density are
  - Dimensionality
    - Sparseness
  - Attribute type
  - Special relationships in the data
    - For example, autocorrelation
  - Distribution of the data
- Noise and Outliers
  - Often interfere with the operation of the clustering algorithm

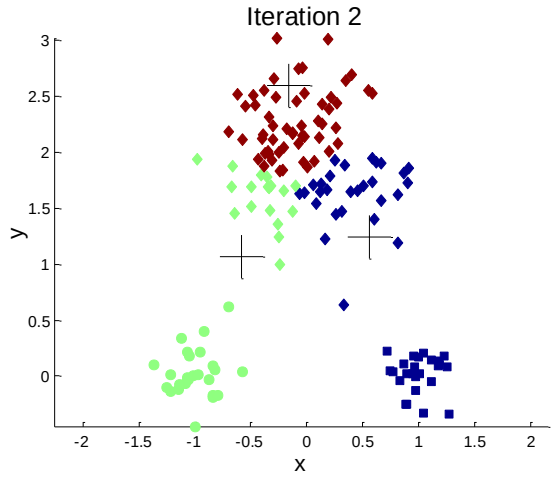
K-means Clustering

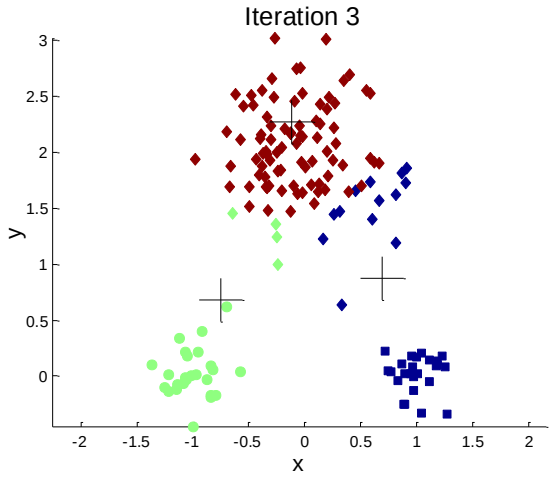
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

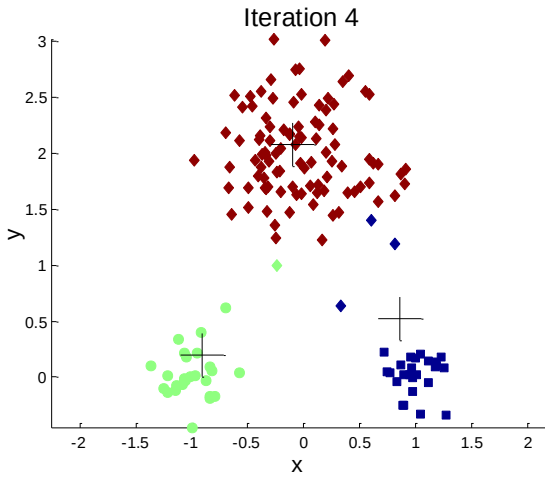
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

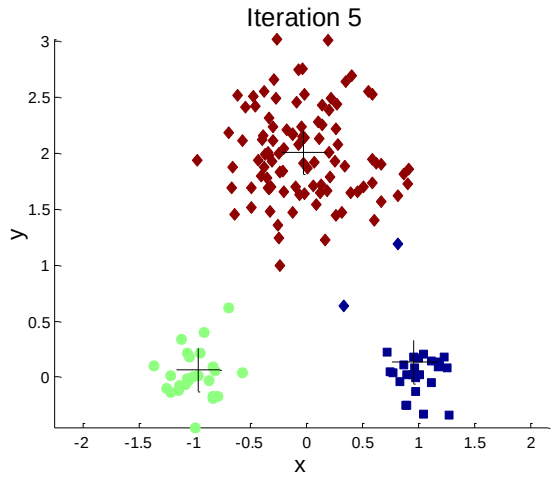
Example of K-means Clustering

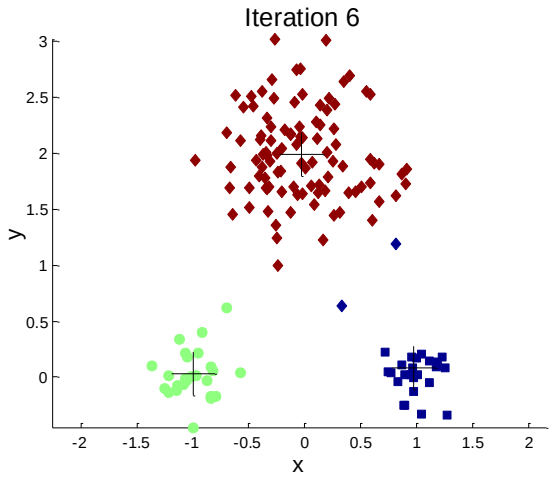












K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

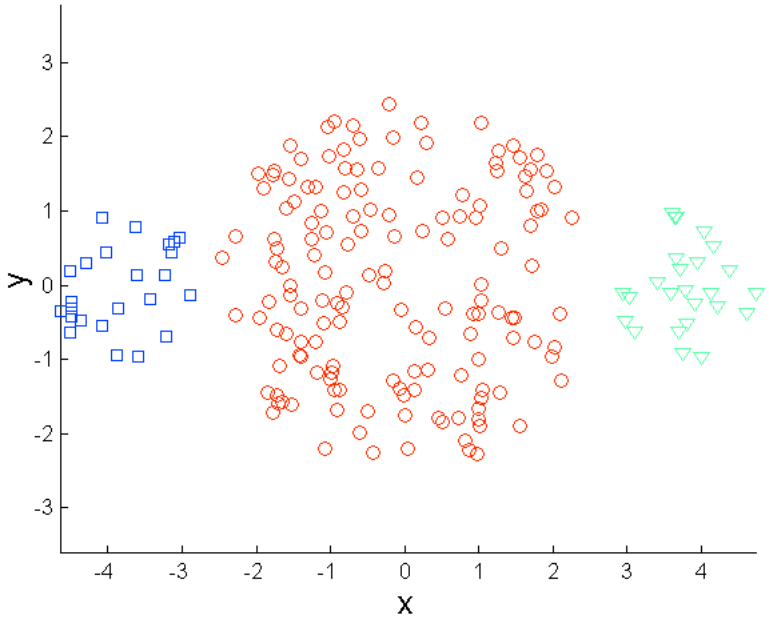
  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two sets of clusters, we prefer the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Limitations of K-means

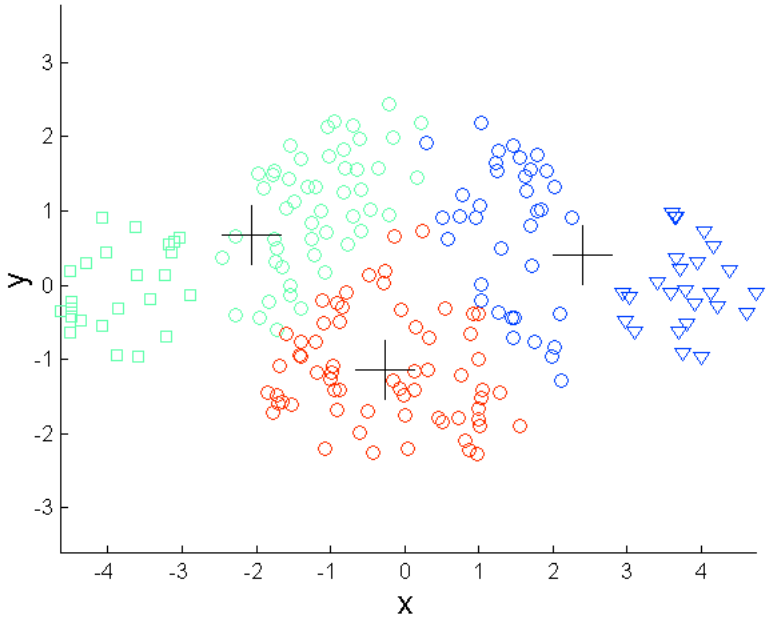
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

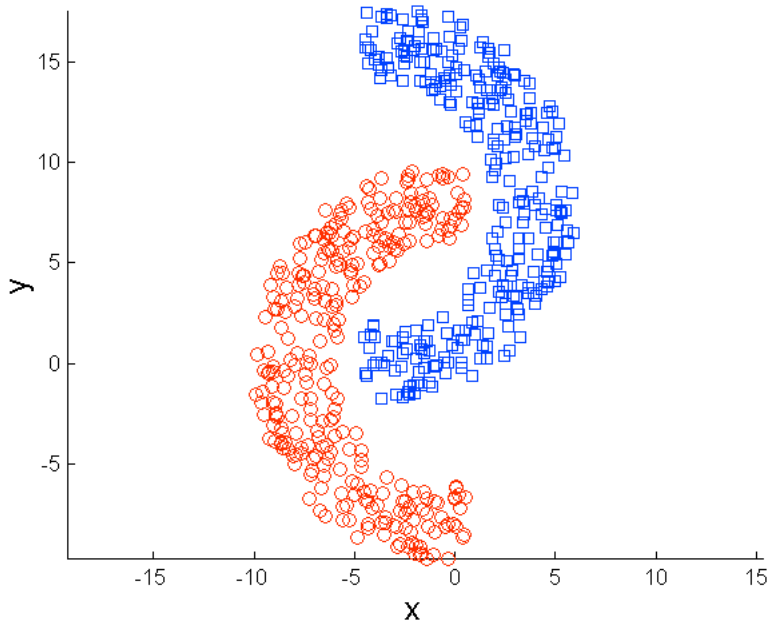


Original Points

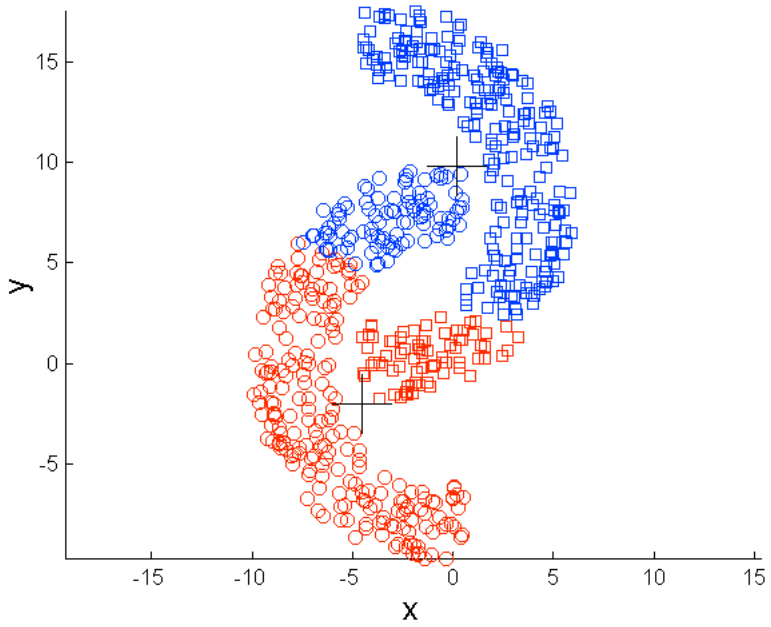


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points



K-means (2 Clusters)

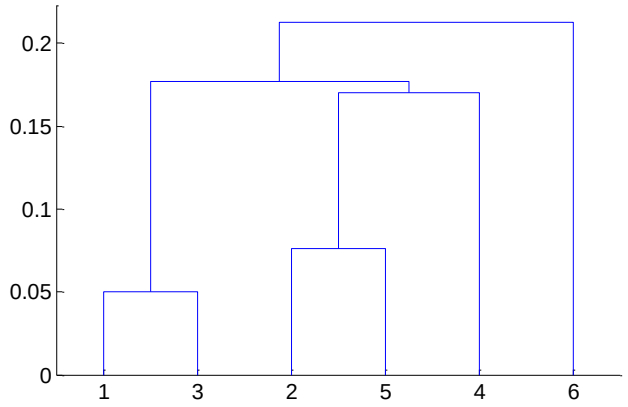
K-means

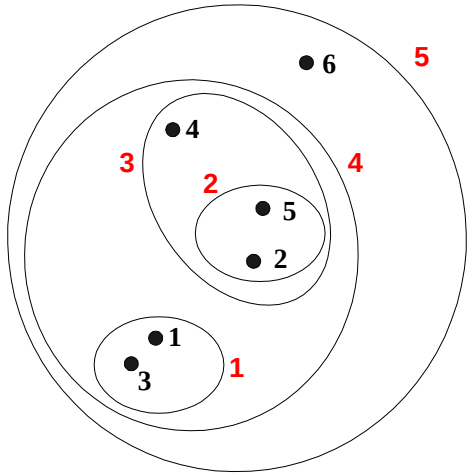
Solutions of choosing K:

1. Trial-and-error
2. Iterative heuristics

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

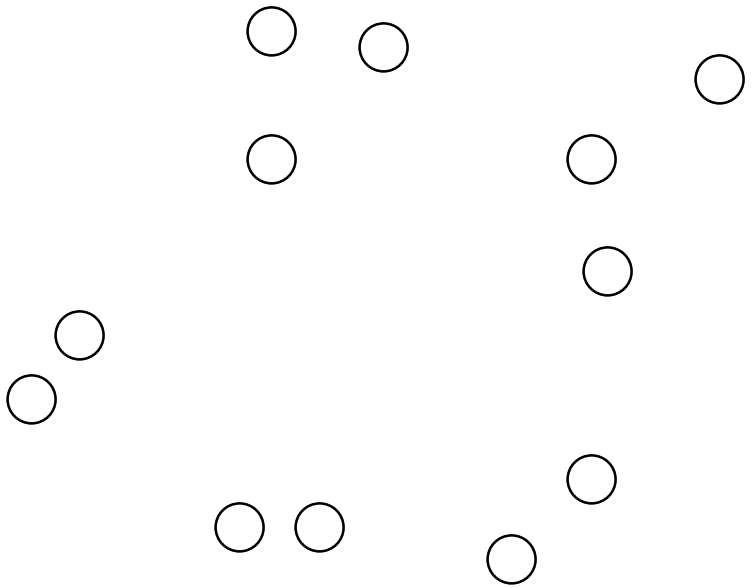
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains an individual point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

- Most popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

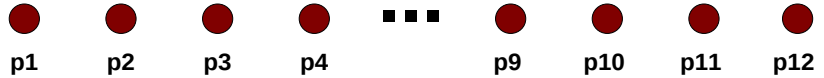
Starting Situation

- Start with clusters of individual points and a proximity matrix



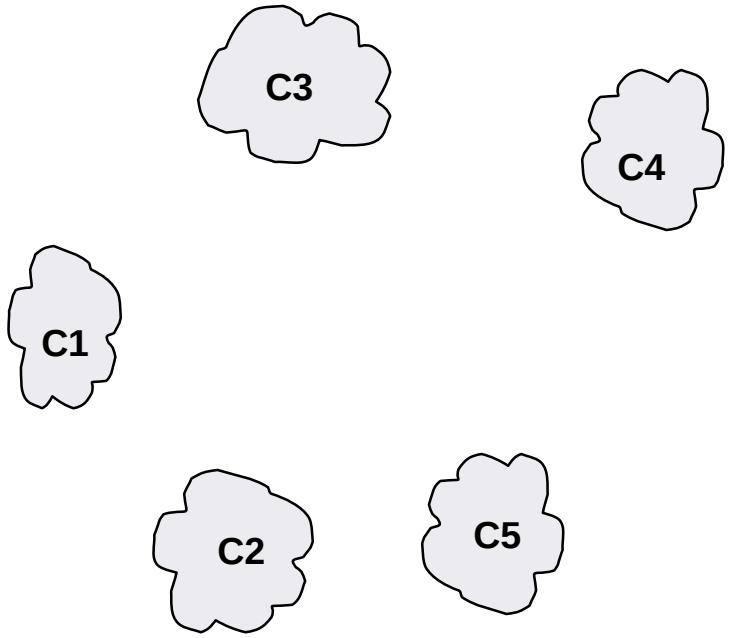
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



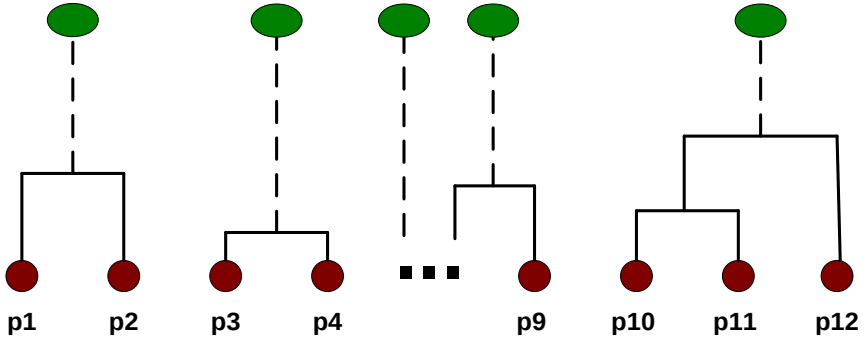
Intermediate Situation

- After some merging steps, we have some clusters



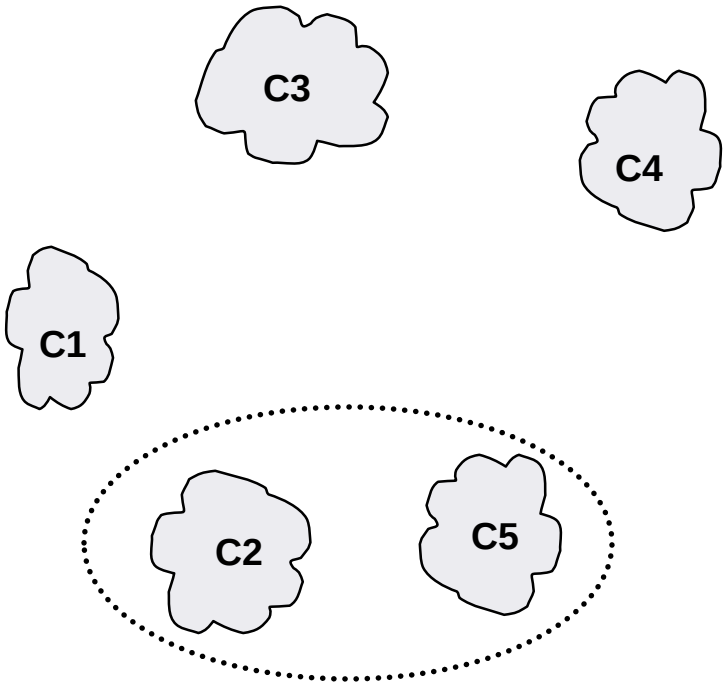
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



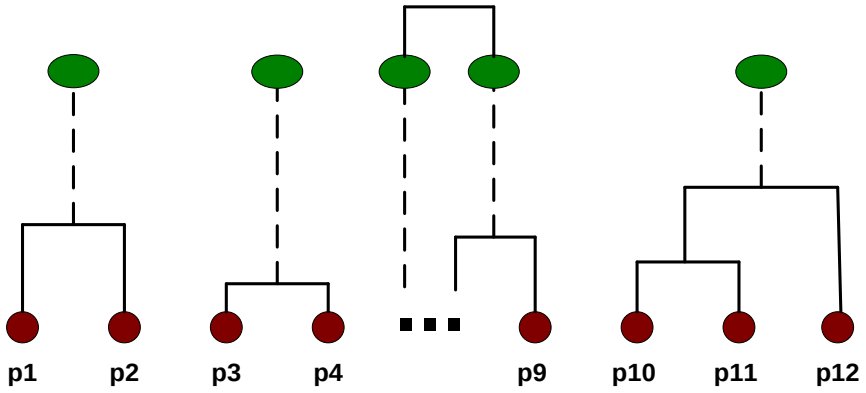
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



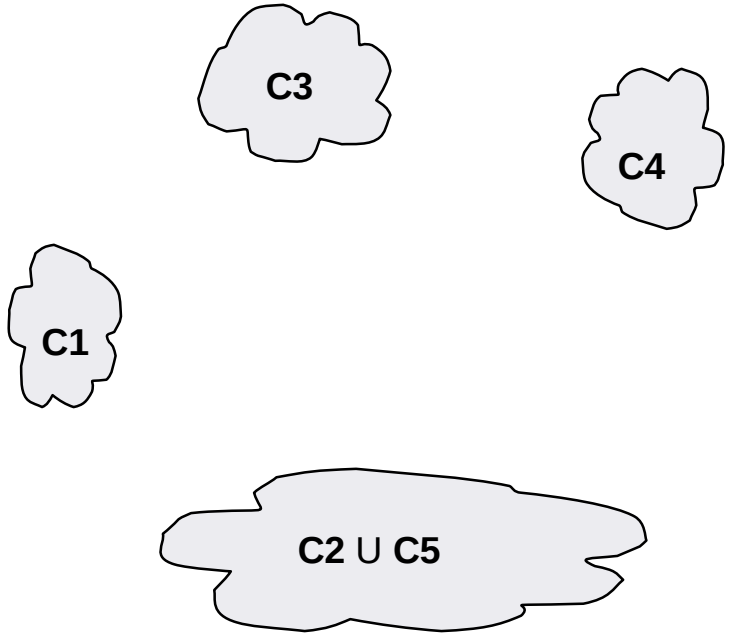
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



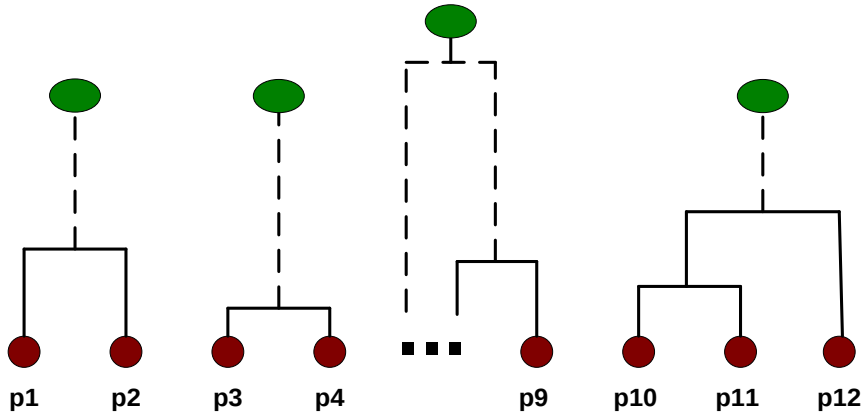
After Merging

- The question is “How do we update the proximity matrix?”

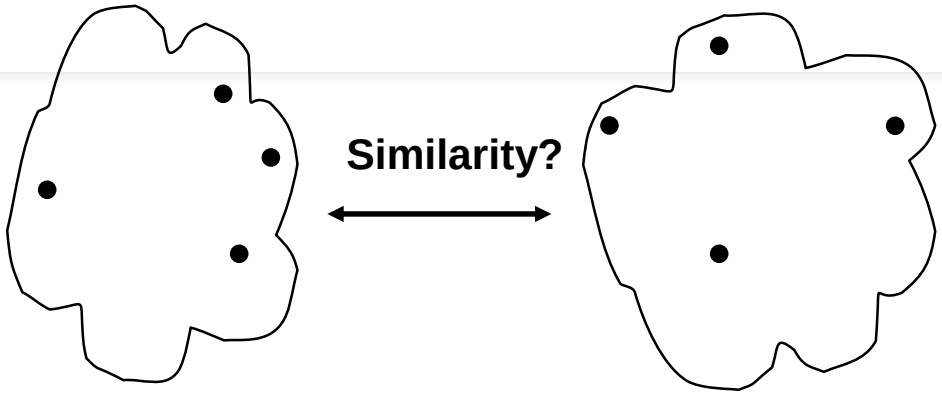


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Distance

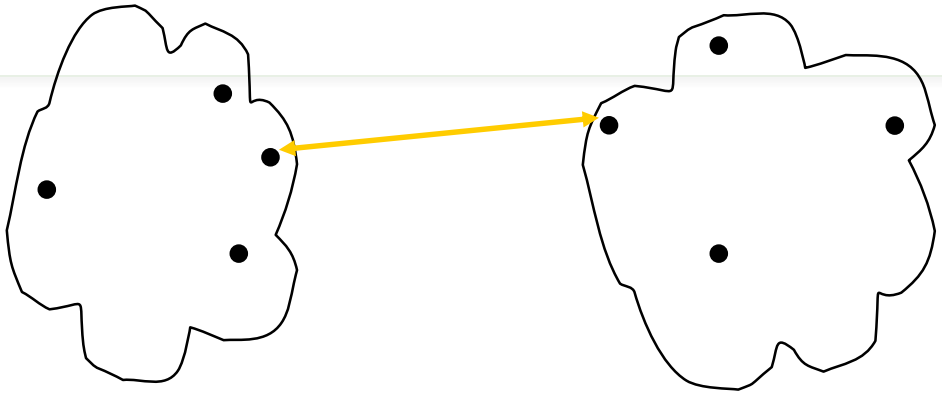


- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity

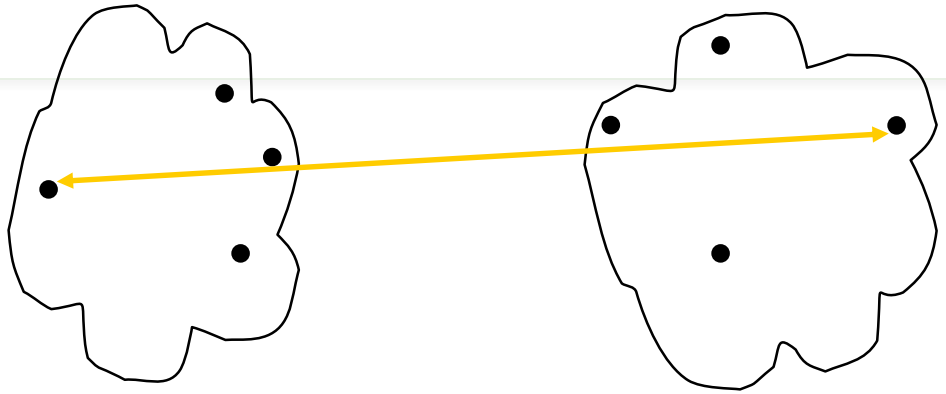


- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity

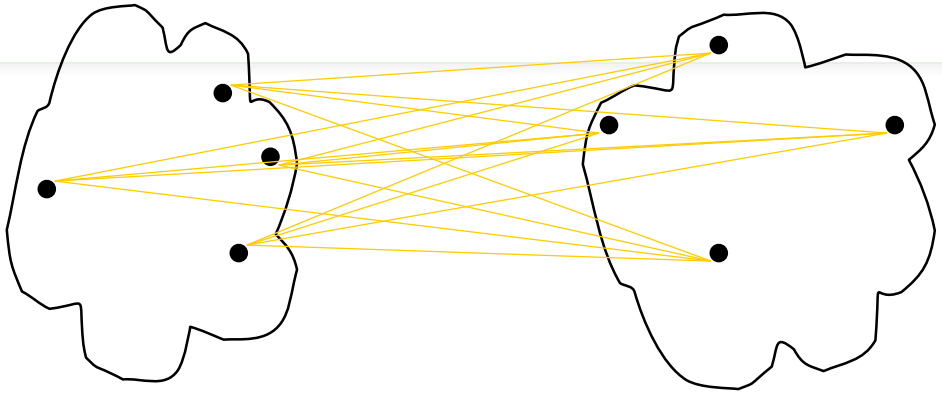


- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity

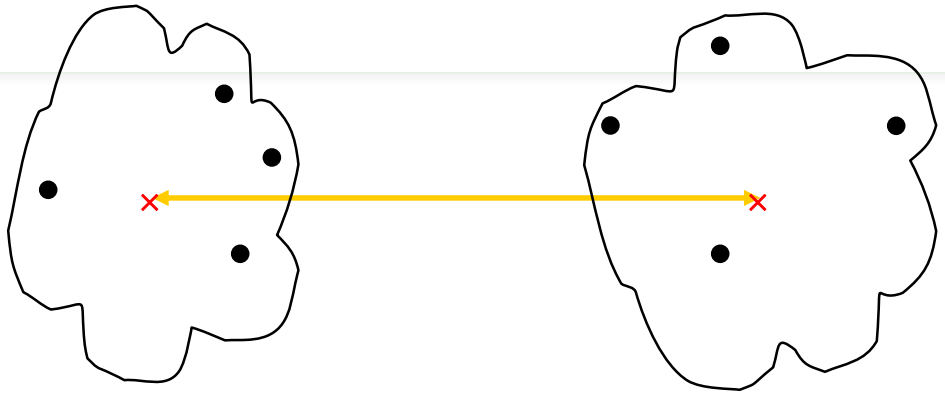


- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity



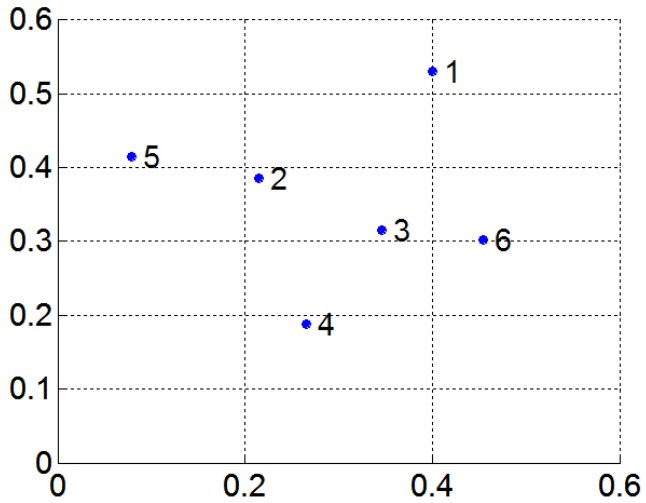
- MIN
- MAX
- Group Average
- Distance Between Centroids

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

MIN or Single Link

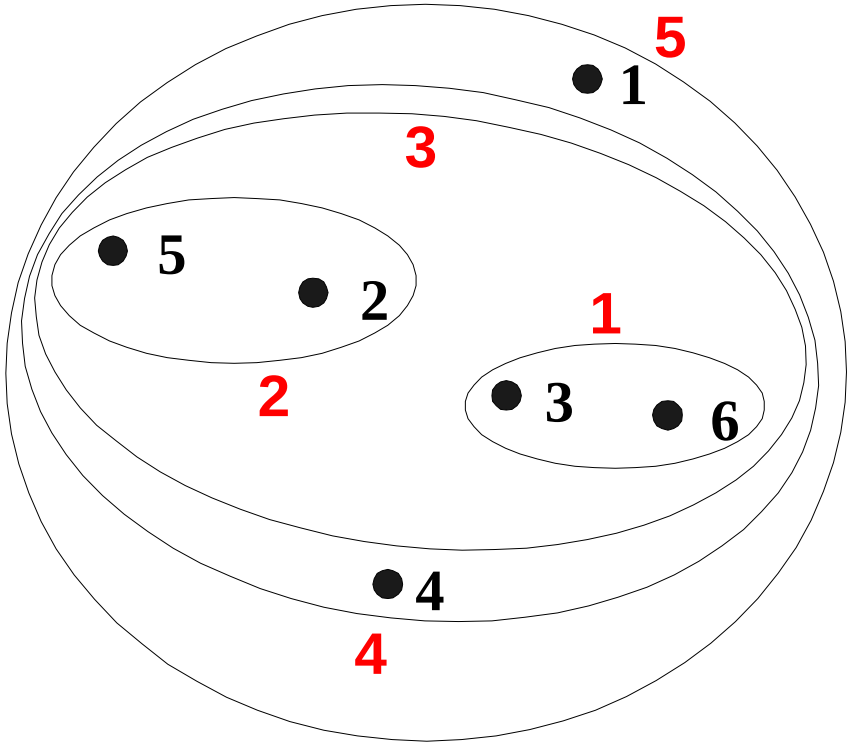
- Proximity of two clusters is based on the two closest points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph
- Example:



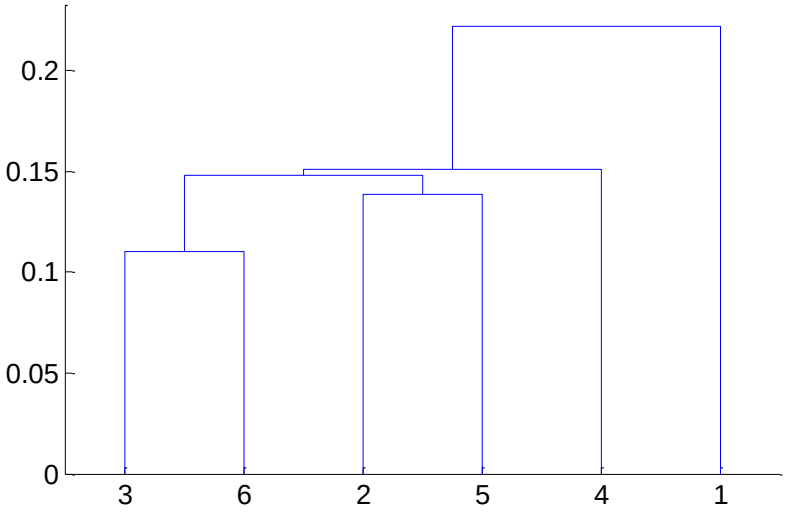
Distance Matrix:

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MIN

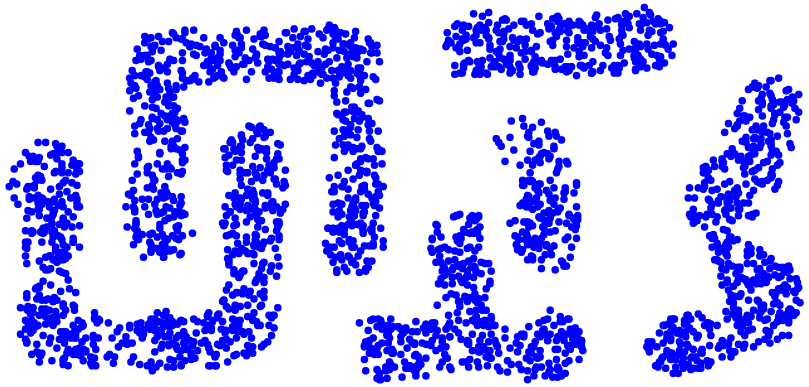


Nested Clusters

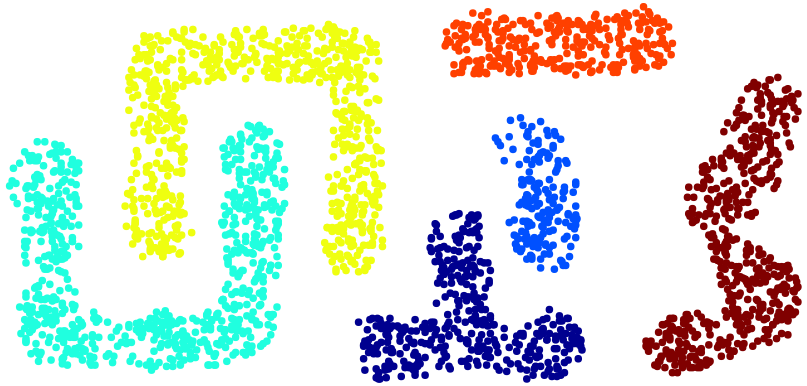


Dendrogram

Strength of MIN



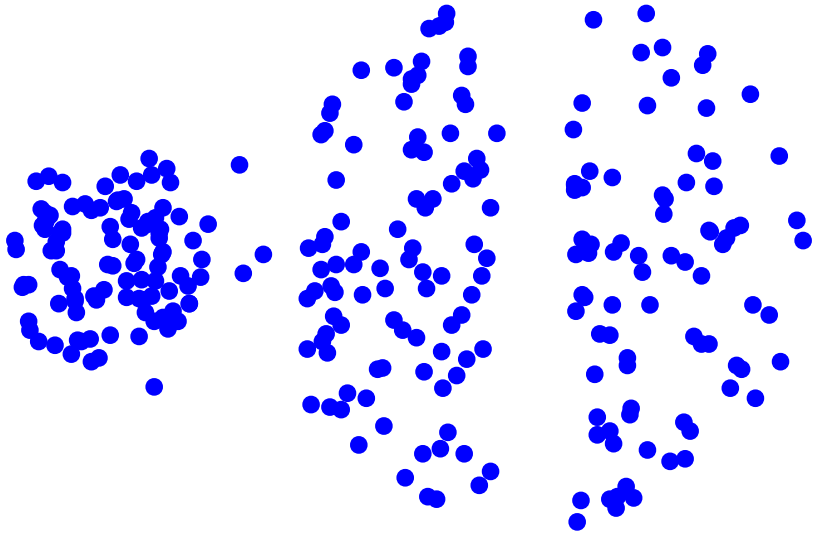
Original Points



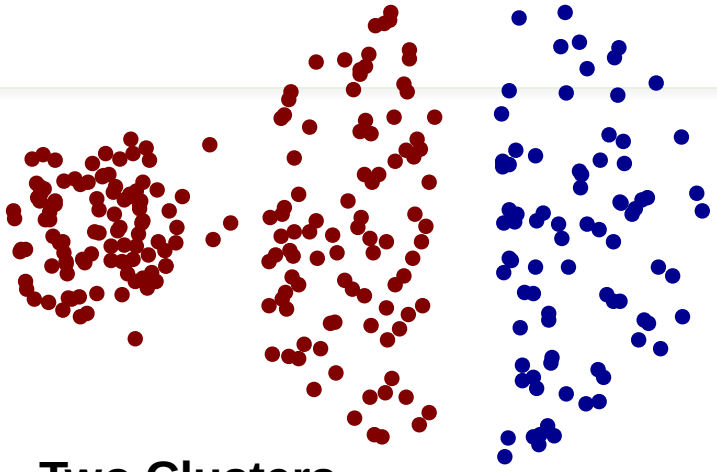
Six Clusters

- **Can handle non-elliptical shapes**

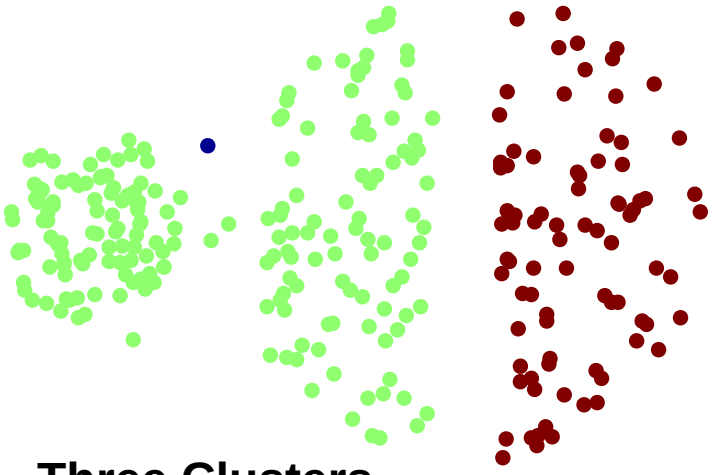
Limitations of MIN



Original Points



Two Clusters

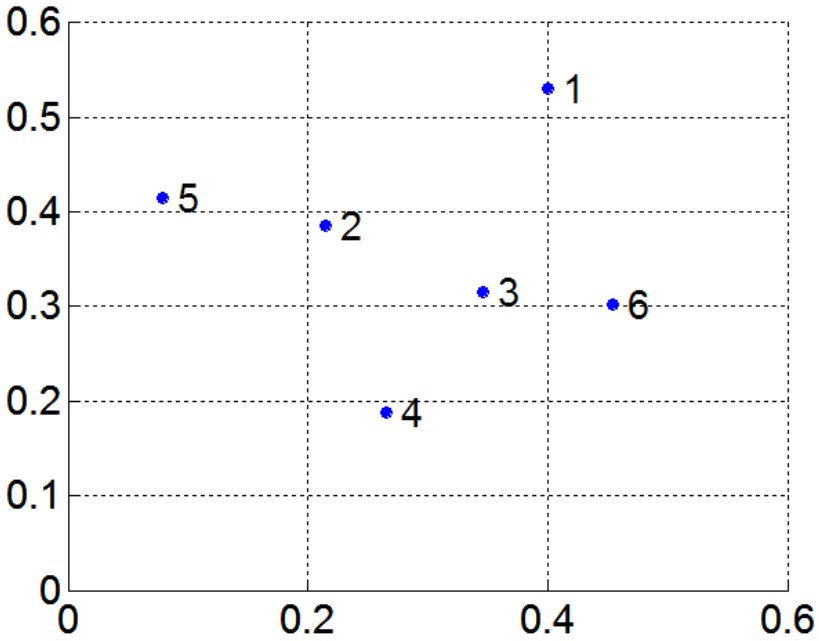


Three Clusters

- Sensitive to noise and outliers

MAX or Complete Linkage

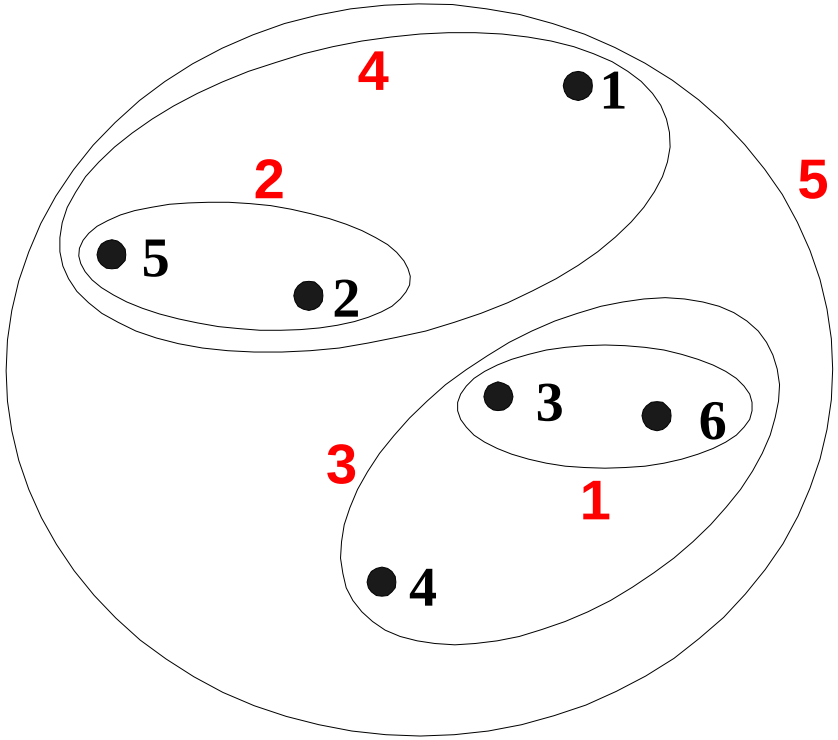
- Proximity of two clusters is based on the two most distant points in the different clusters
  - Determined by all pairs of points in the two clusters



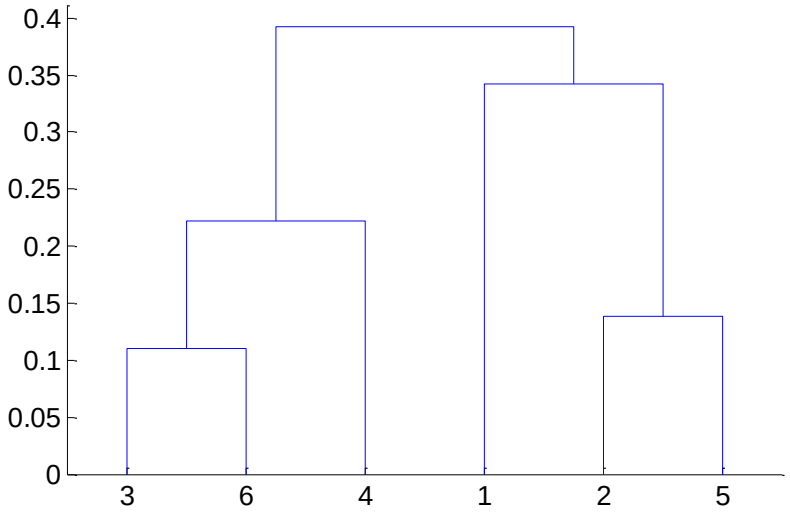
Distance Matrix:

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: MAX

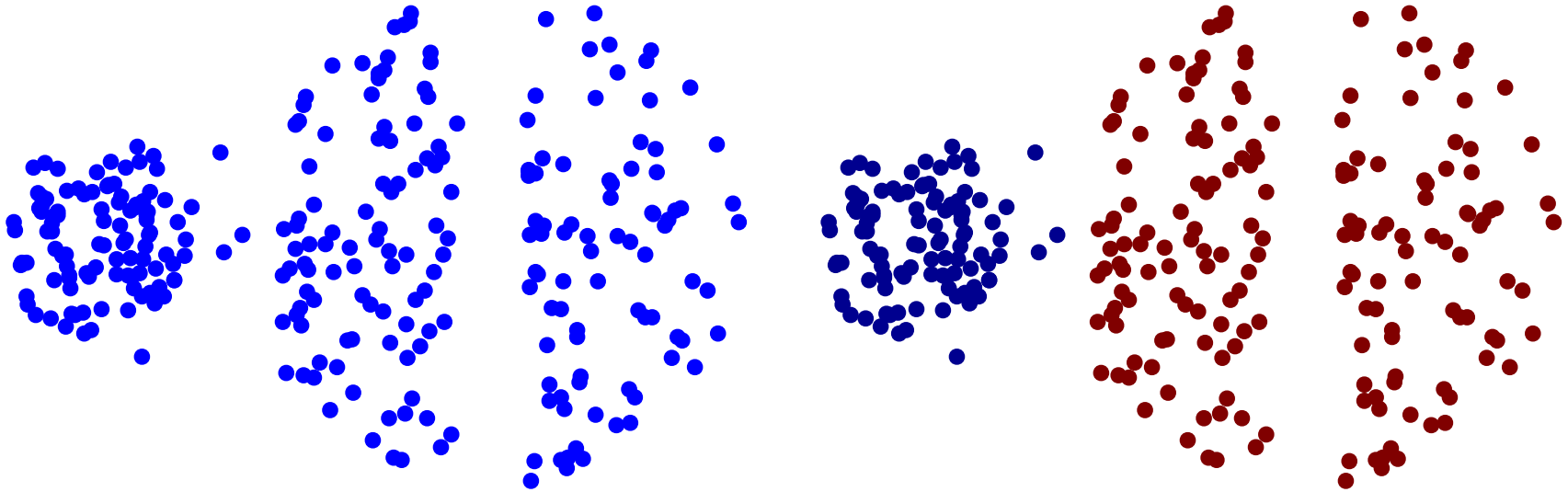


Nested Clusters



Dendrogram

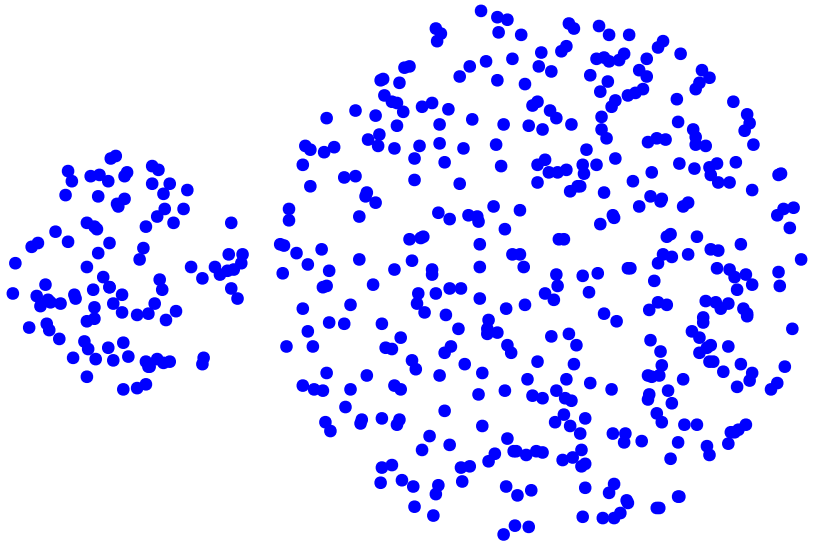
Strength of MAX



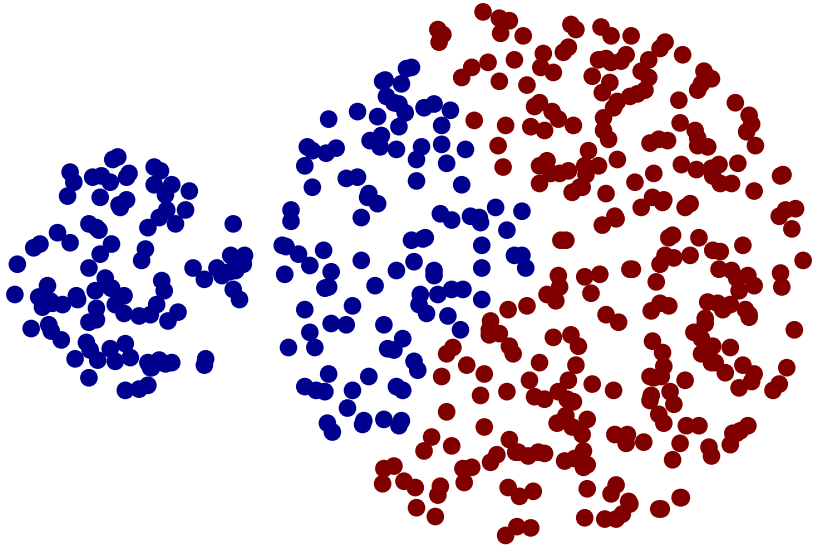
Original Points **Two Clusters**

- **Less susceptible to noise and outliers**

Limitations of MAX



Original Points



Two Clusters

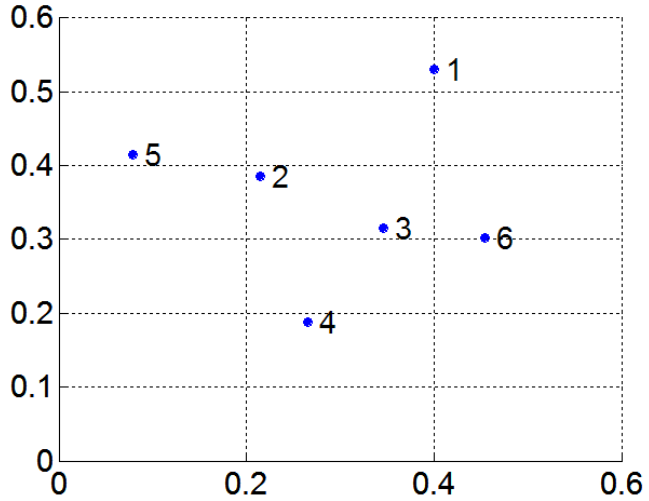
- **Tends to break large clusters**
- **Biased towards globular clusters**

Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| \times |\text{Cluster}_j|}$$

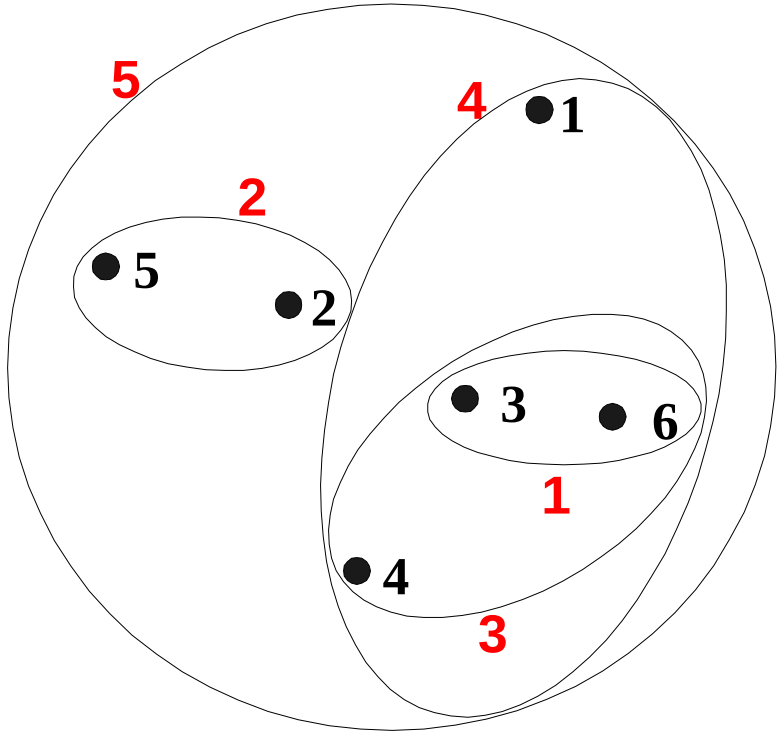
- Need to use average connectivity for scalability since total proximity favors large clusters



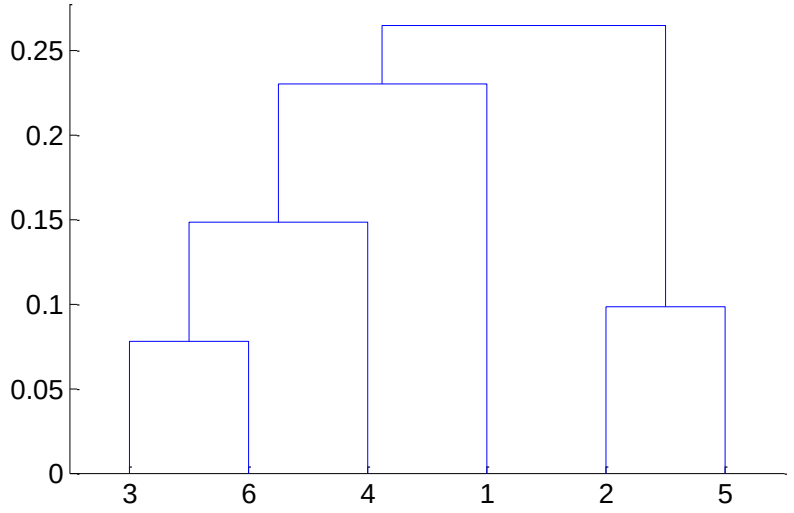
Distance Matrix:

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

Hierarchical Clustering: Group Average

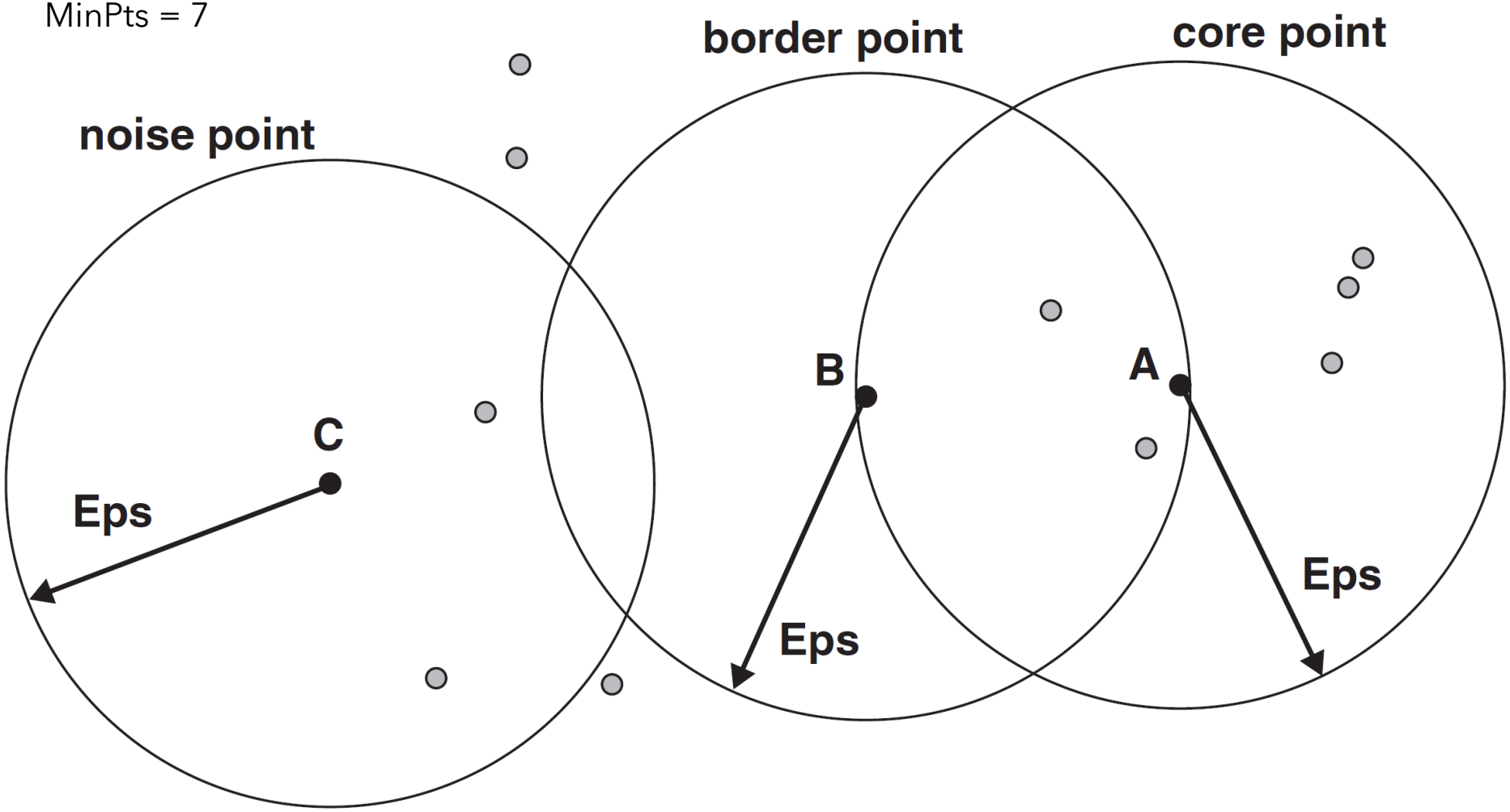
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has at least a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
    - Counts the point itself
  - A border point is not a core point, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point

DBSCAN: Core, Border, and Noise Points

MinPts = 7



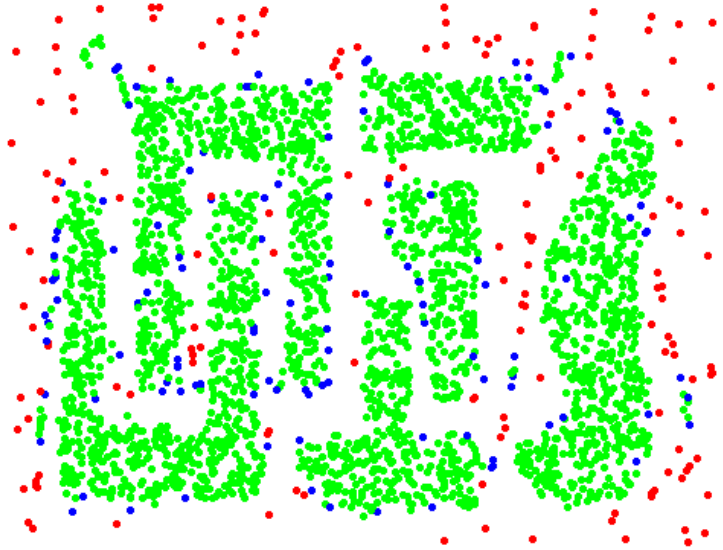
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
end for
```

DBSCAN: Core, Border and Noise Points

Original Points

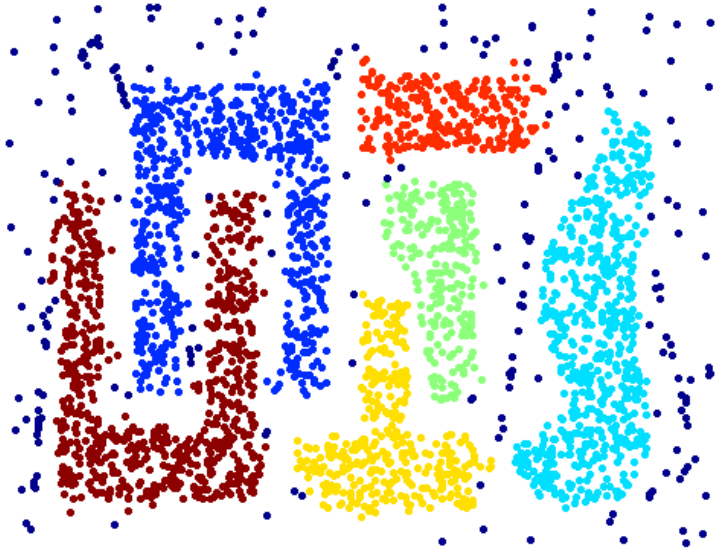


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

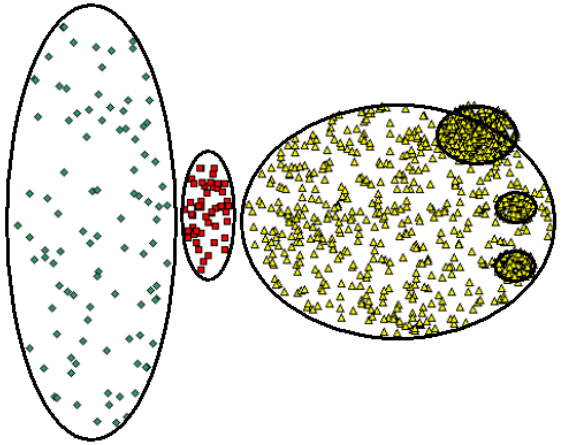
Original Points



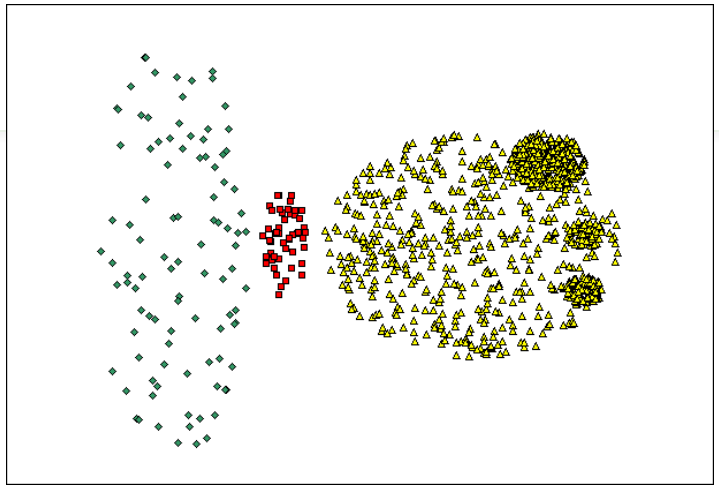
Clusters

- **Resistant to Noise**
- **Can handle clusters of different shapes and sizes**

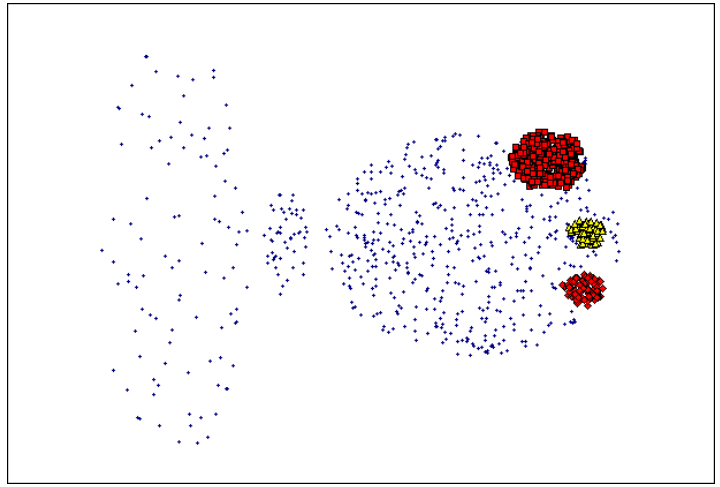
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).

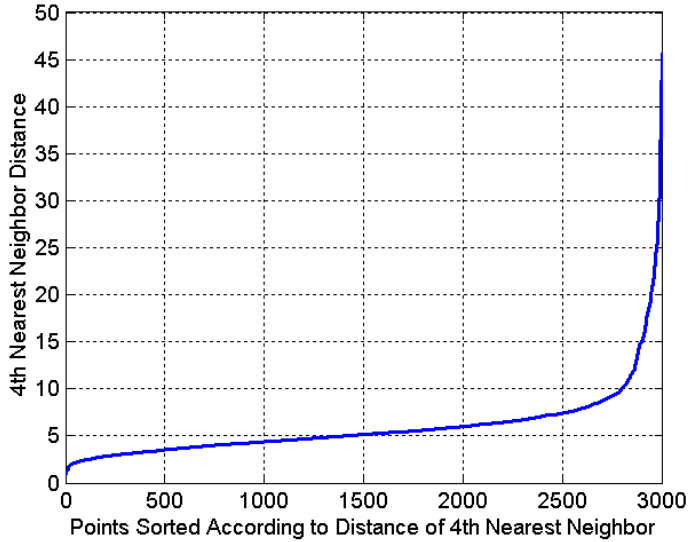


(MinPts=4, Eps=9.92)

- **Varying densities**
- **High-dimensional data**

DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
    - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

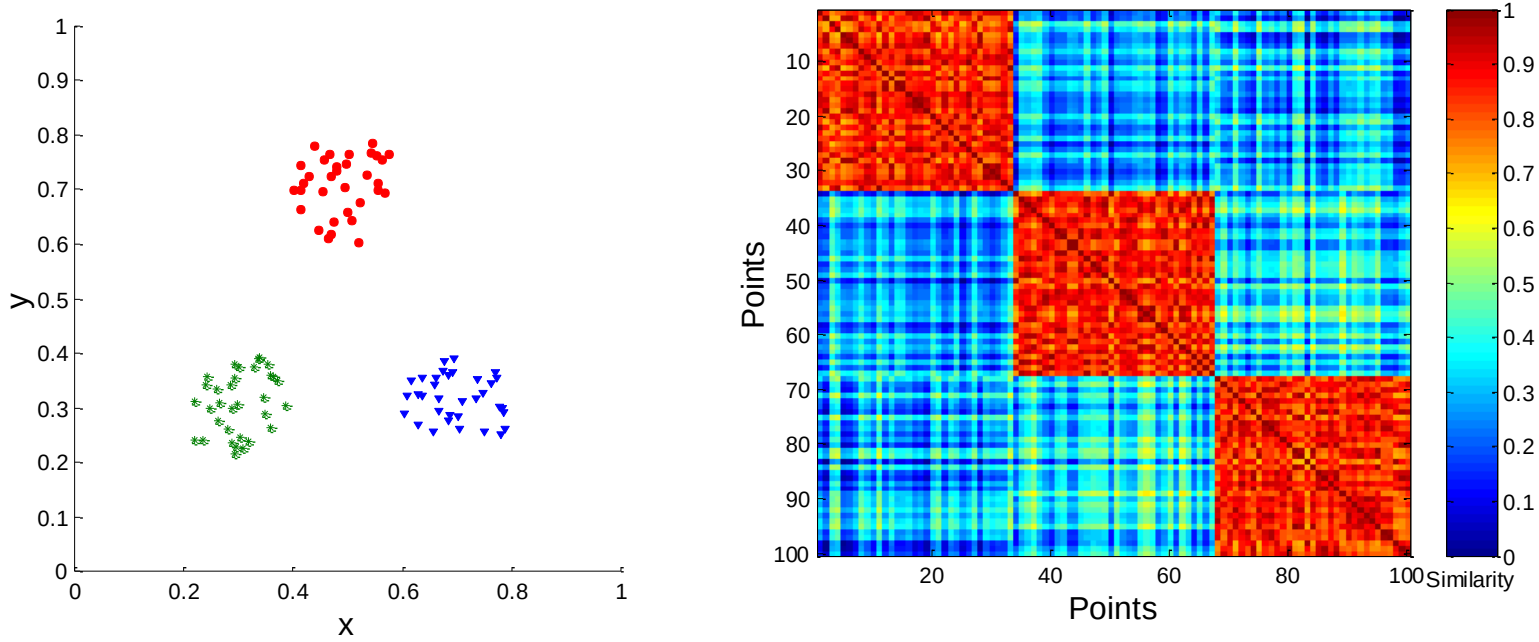
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measuring Cluster Validity Via Correlation

- Two matrices
  - Proximity Matrix
  - Ideal Similarity Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.



Internal Measures: Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$SSE = WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

    - Where $|C_i|$ is the size of cluster i

SDSC 6004

Laboratory Section – Macro Enabled Data Process and Analytics

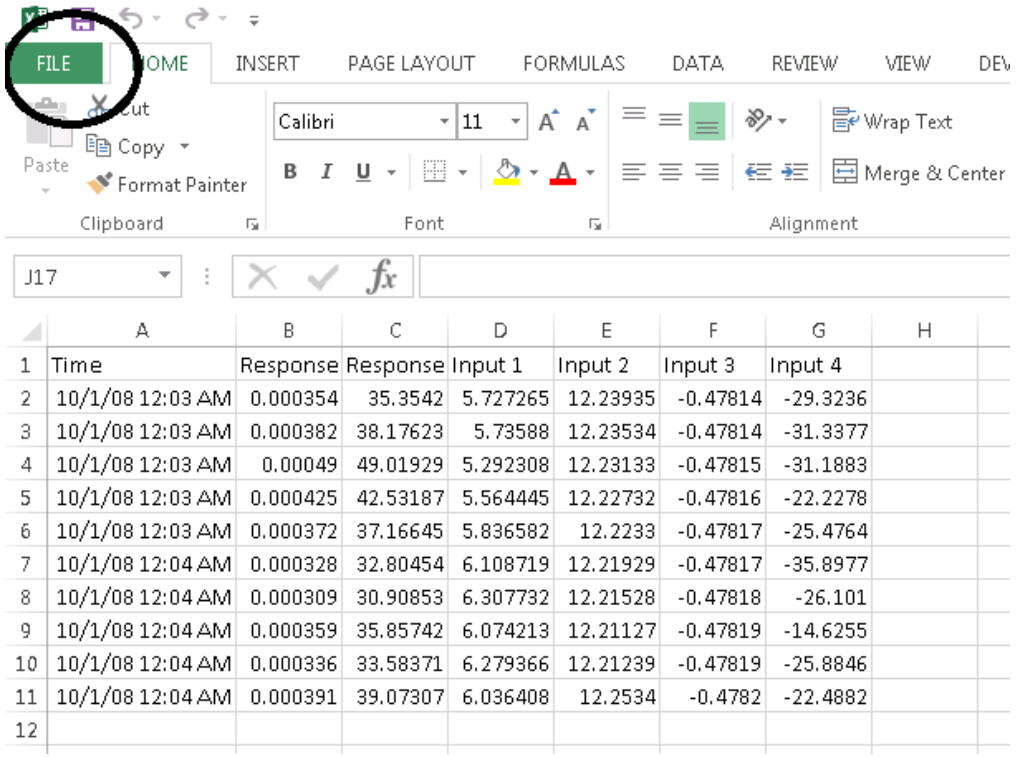
Aims

- A. Learn develop macro via recording
  - Develop a macro for routine excel operations;
  - Apply the macro to process data.
- B. Learn customized macro via visual basic programming
  - Define variables;
  - Understand basic syntax;
  - Complete proposed data processing tasks.

Develop Macro via Recording

- Excel setup

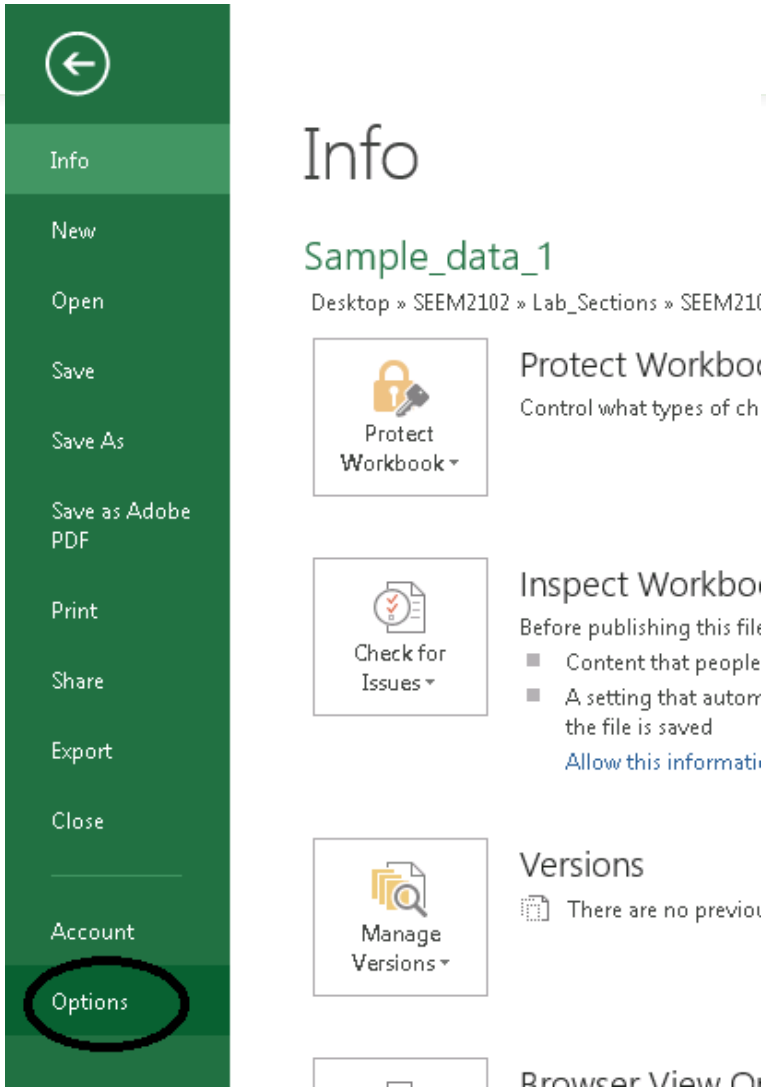
Step 1: left-click File tag



	A	B	C	D	E	F	G	H
1	Time	Response	Response	Input 1	Input 2	Input 3	Input 4	
2	10/1/08 12:03 AM	0.000354	35.3542	5.727265	12.23935	-0.47814	-29.3236	
3	10/1/08 12:03 AM	0.000382	38.17623	5.73588	12.23534	-0.47814	-31.3377	
4	10/1/08 12:03 AM	0.00049	49.01929	5.292308	12.23133	-0.47815	-31.1883	
5	10/1/08 12:03 AM	0.000425	42.53187	5.564445	12.22732	-0.47816	-22.2278	
6	10/1/08 12:03 AM	0.000372	37.16645	5.836582	12.2233	-0.47817	-25.4764	
7	10/1/08 12:04 AM	0.000328	32.80454	6.108719	12.21929	-0.47817	-35.8977	
8	10/1/08 12:04 AM	0.000309	30.90853	6.307732	12.21528	-0.47818	-26.101	
9	10/1/08 12:04 AM	0.000359	35.85742	6.074213	12.21127	-0.47819	-14.6255	
10	10/1/08 12:04 AM	0.000336	33.58371	6.279366	12.21239	-0.47819	-25.8846	
11	10/1/08 12:04 AM	0.000391	39.07307	6.036408	12.2534	-0.4782	-22.4882	
12								

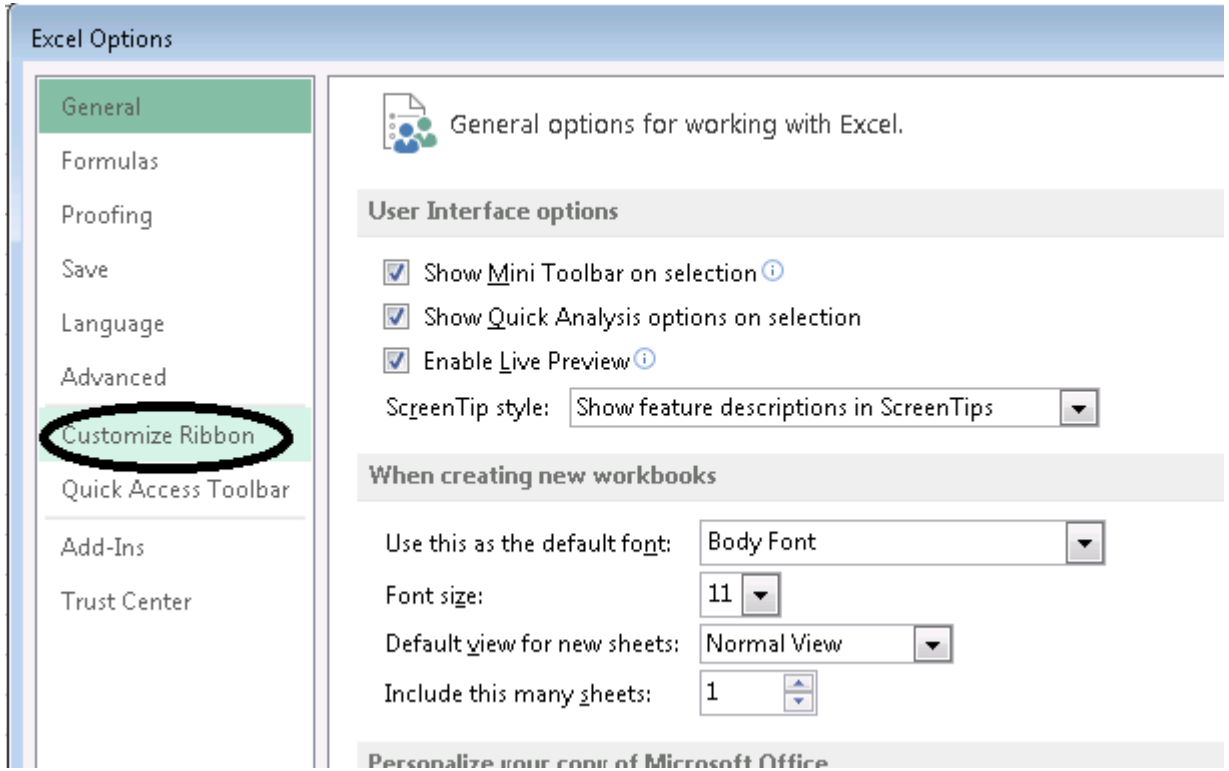
Develop Macro via Recording

Step 2: left-click the Options tag



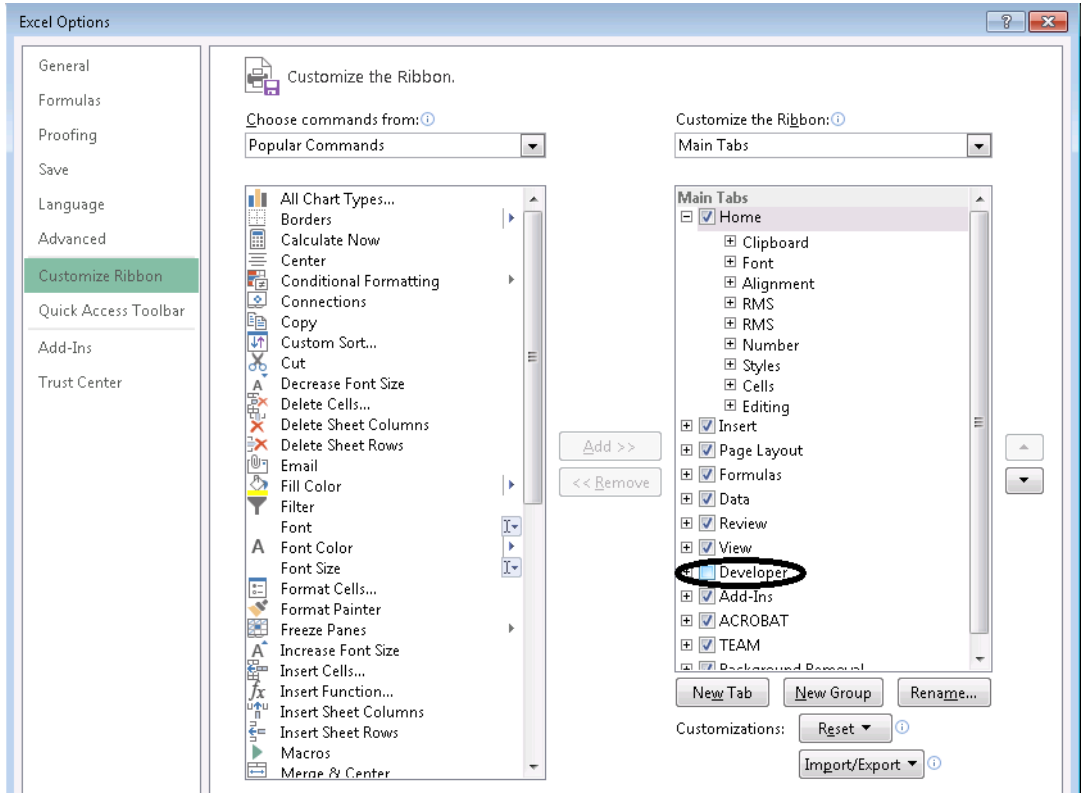
Develop Macro via Recording

Step 3: left-click the Customize Ribbon tag



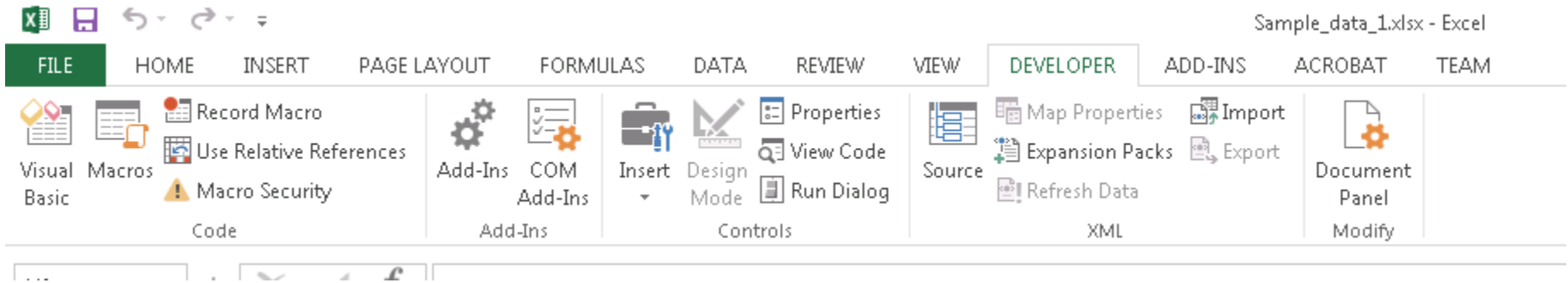
Develop Macro via Recording

Step 4: Tick Developer option and Click OK.



Develop Macro via Recording

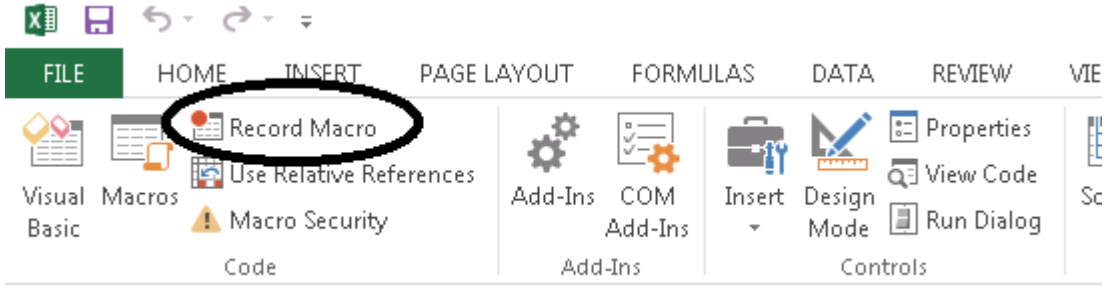
Step 5: Check the environment has been setup as follows



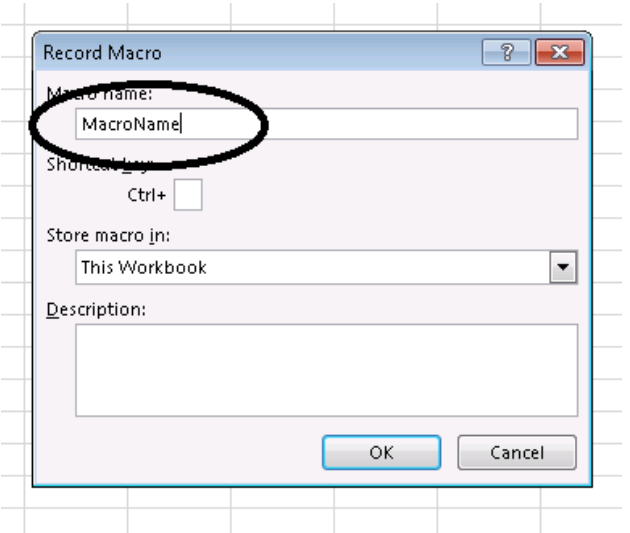
Develop Macro via Recording

- Learn the usage of Macro recording

Step 1: Left click Macro Record



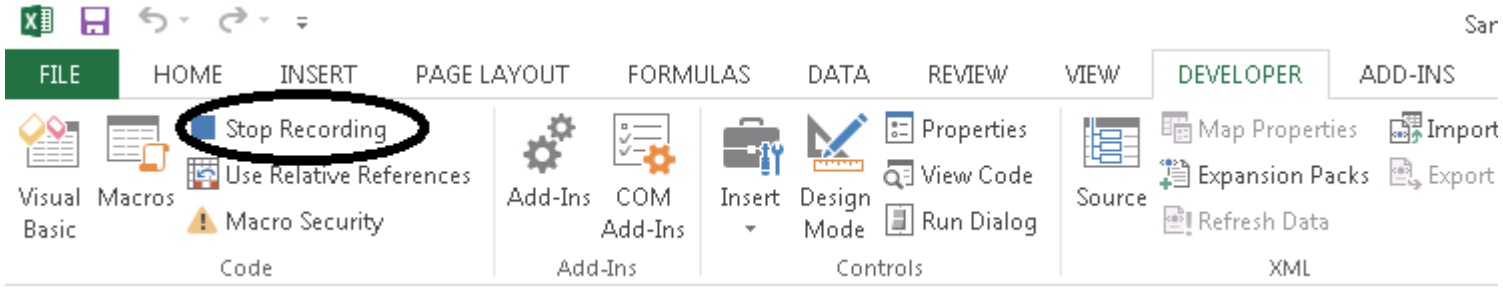
Step 2: Provide a name of Macro



Develop Macro via Recording

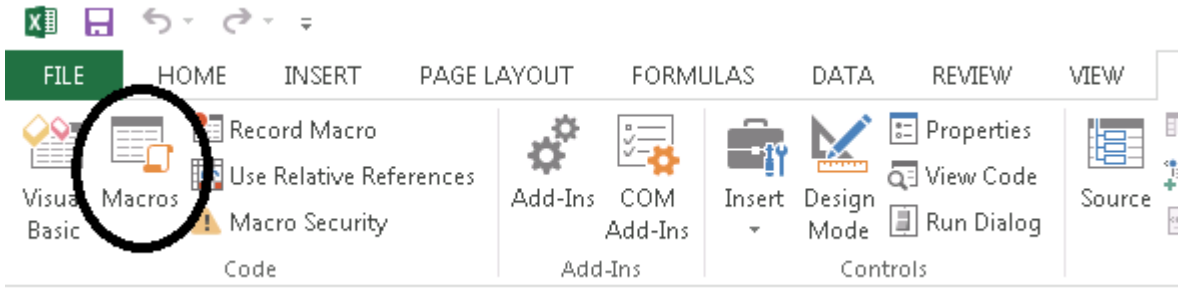
Step 3: Start your excel operations

Step 4: Stop Macro Record by left-click Stop Recording button once your operations have been finished



Develop Macro via Recording

Step 5: View your Macro by left-click Macros

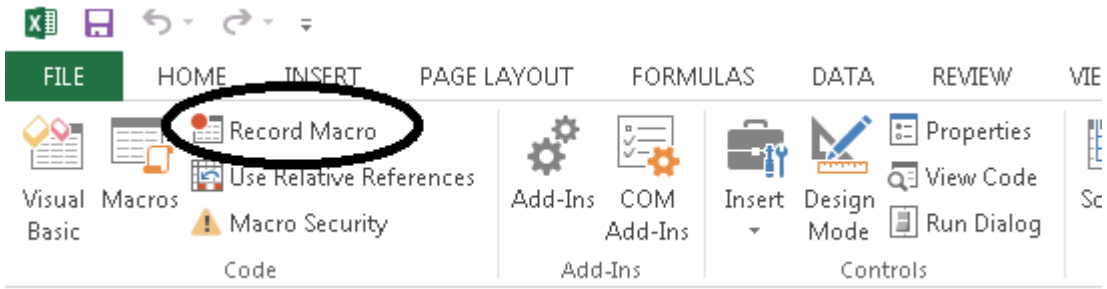


Develop Macro via Recording

- Application 1: Copy original dataset to a new worksheet

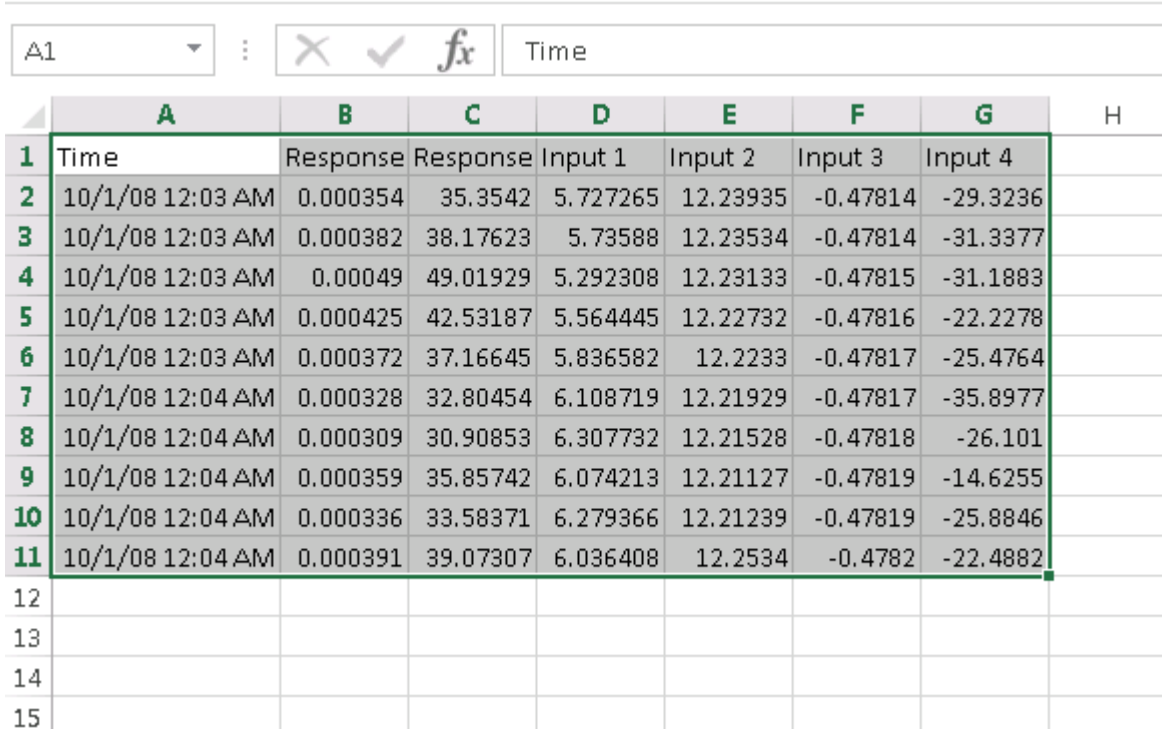
Used Dataset: Sample_data_1.xlsx

A1 – Find the developer tag and click Record Macro button, following you need to provide a name to the new Macro and click ok



Develop Macro via Recording

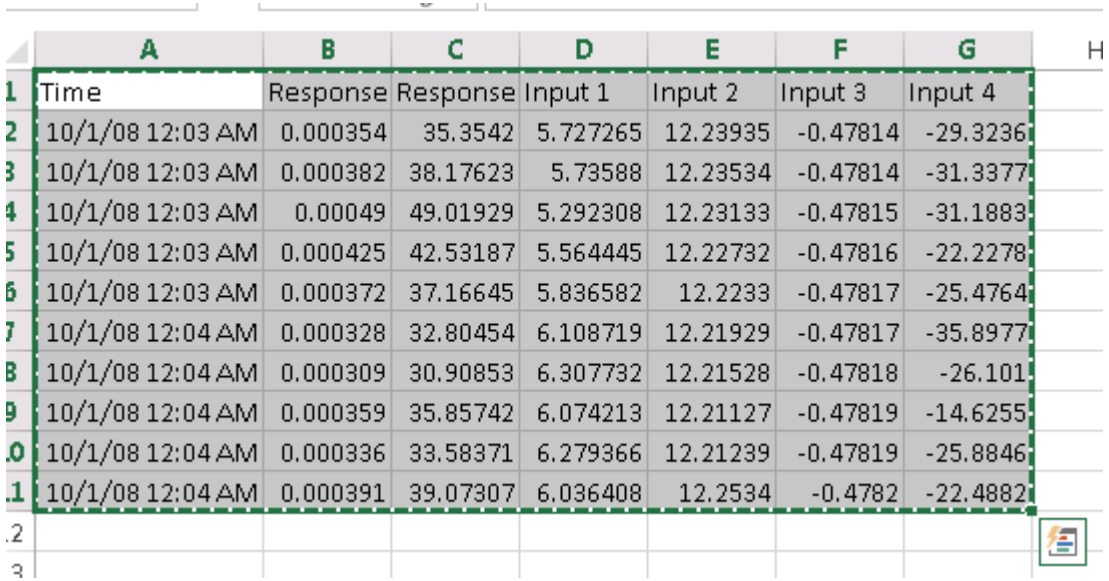
A2 – left-click cell A1 and apply command “Ctrl + A”



	A	B	C	D	E	F	G	H
1	Time	Response	Response	Input 1	Input 2	Input 3	Input 4	
2	10/1/08 12:03 AM	0.000354	35.3542	5.727265	12.23935	-0.47814	-29.3236	
3	10/1/08 12:03 AM	0.000382	38.17623	5.73588	12.23534	-0.47814	-31.3377	
4	10/1/08 12:03 AM	0.00049	49.01929	5.292308	12.23133	-0.47815	-31.1883	
5	10/1/08 12:03 AM	0.000425	42.53187	5.564445	12.22732	-0.47816	-22.2278	
6	10/1/08 12:03 AM	0.000372	37.16645	5.836582	12.2233	-0.47817	-25.4764	
7	10/1/08 12:04 AM	0.000328	32.80454	6.108719	12.21929	-0.47817	-35.8977	
8	10/1/08 12:04 AM	0.000309	30.90853	6.307732	12.21528	-0.47818	-26.101	
9	10/1/08 12:04 AM	0.000359	35.85742	6.074213	12.21127	-0.47819	-14.6255	
10	10/1/08 12:04 AM	0.000336	33.58371	6.279366	12.21239	-0.47819	-25.8846	
11	10/1/08 12:04 AM	0.000391	39.07307	6.036408	12.2534	-0.4782	-22.4882	
12								
13								
14								
15								

Develop Macro via Recording

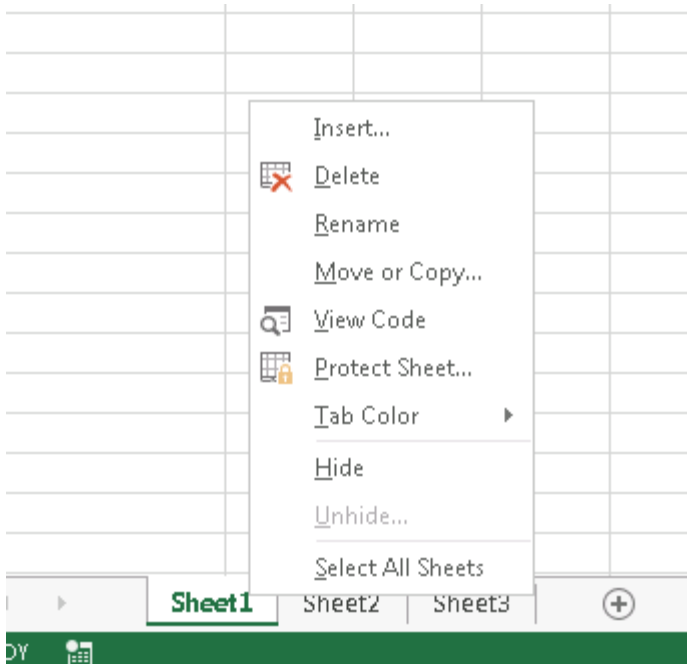
A3: Apply command “Ctrl + C”



	A	B	C	D	E	F	G
1	Time	Response	Response	Input 1	Input 2	Input 3	Input 4
2	10/1/08 12:03 AM	0.000354	35.3542	5.727265	12.23935	-0.47814	-29.3236
3	10/1/08 12:03 AM	0.000382	38.17623	5.73588	12.23534	-0.47814	-31.3377
4	10/1/08 12:03 AM	0.00049	49.01929	5.292308	12.23133	-0.47815	-31.1883
5	10/1/08 12:03 AM	0.000425	42.53187	5.564445	12.22732	-0.47816	-22.2278
6	10/1/08 12:03 AM	0.000372	37.16645	5.836582	12.2233	-0.47817	-25.4764
7	10/1/08 12:04 AM	0.000328	32.80454	6.108719	12.21929	-0.47817	-35.8977
8	10/1/08 12:04 AM	0.000309	30.90853	6.307732	12.21528	-0.47818	-26.101
9	10/1/08 12:04 AM	0.000359	35.85742	6.074213	12.21127	-0.47819	-14.6255
10	10/1/08 12:04 AM	0.000336	33.58371	6.279366	12.21239	-0.47819	-25.8846
11	10/1/08 12:04 AM	0.000391	39.07307	6.036408	12.2534	-0.4782	-22.4882
12							
13							

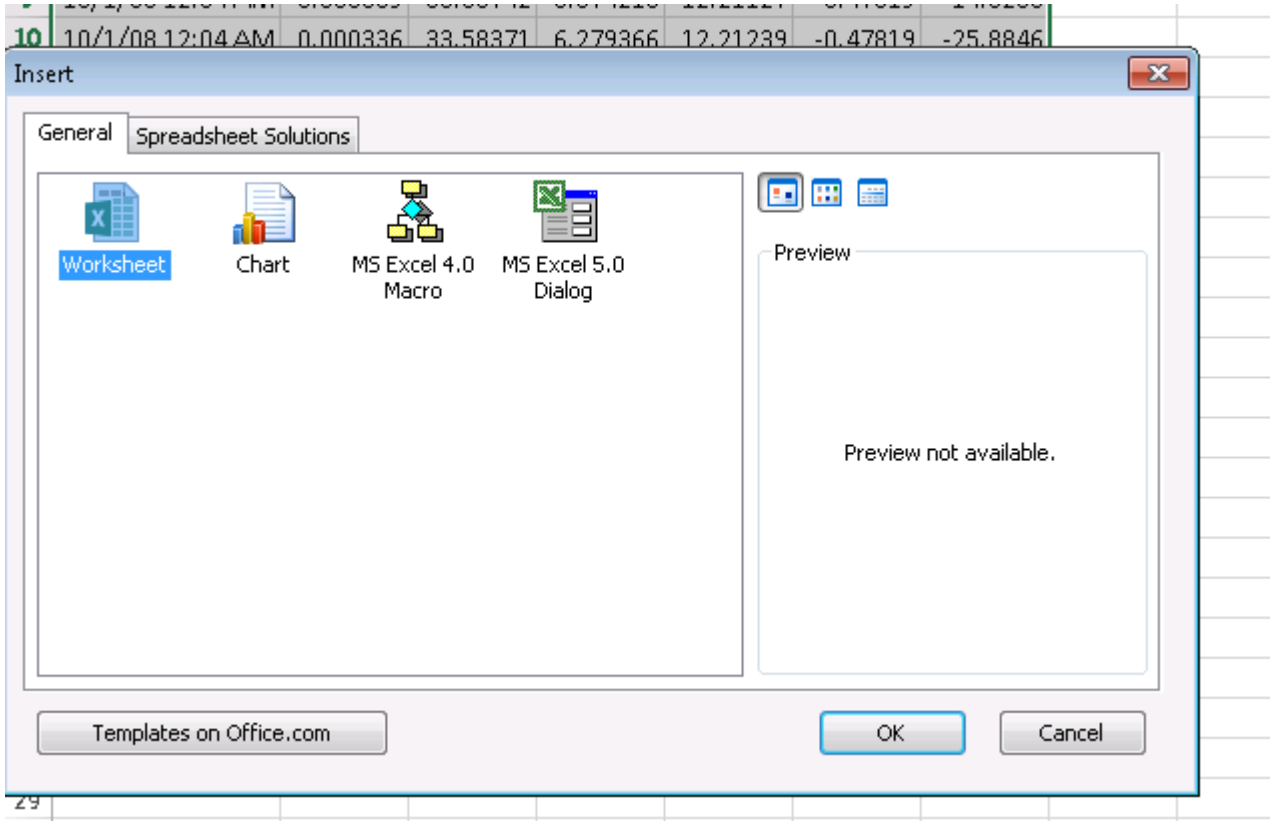
Develop Macro via Recording

A4: Move your mouse to tag of sheets, right-click, and select Insert



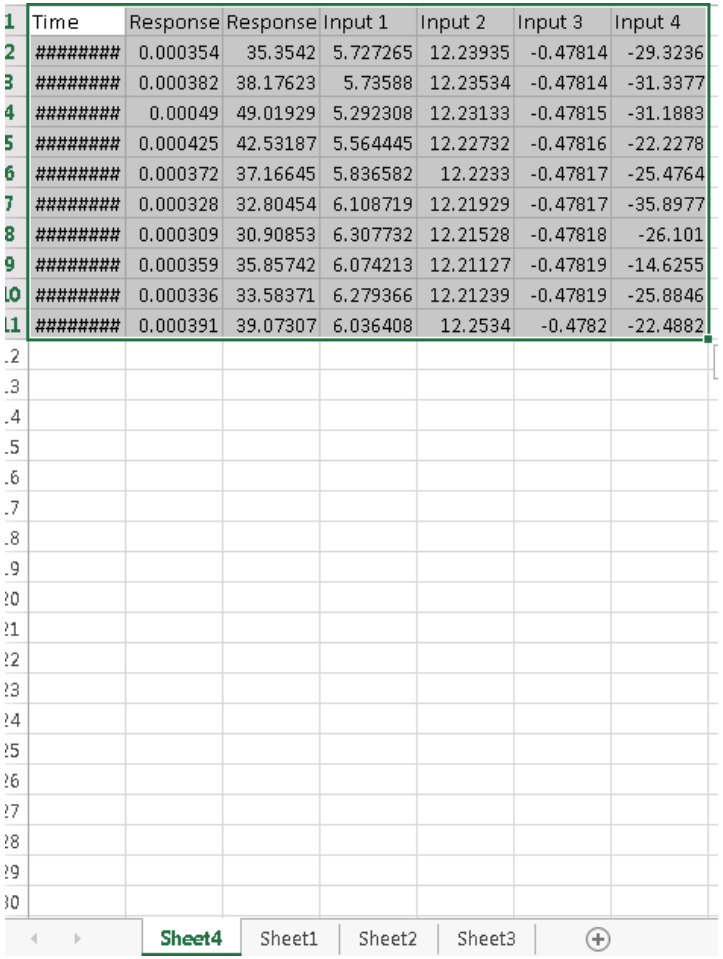
Develop Macro via Recording

A5 – Insert a new worksheet



Develop Macro via Recording

A6 – Apply command “Ctrl + V”

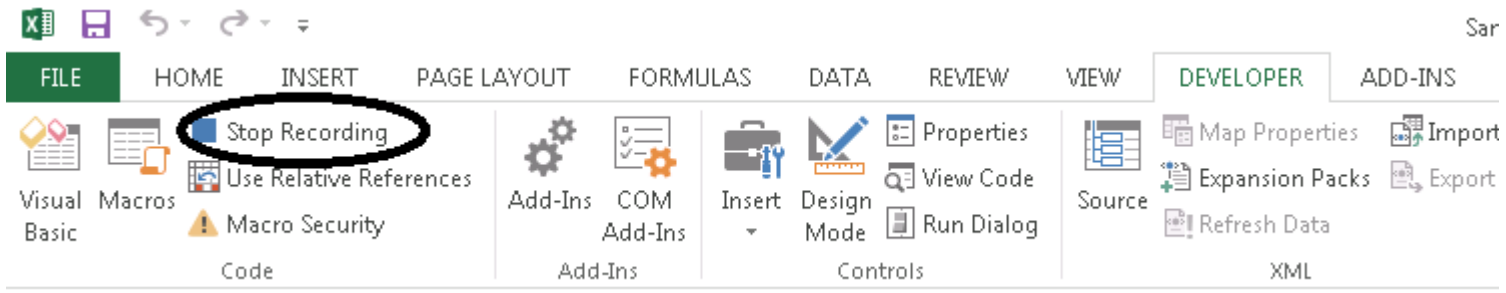


	Time	Response	Response	Input 1	Input 2	Input 3	Input 4
2	########	0.000354	35.3542	5.727265	12.23935	-0.47814	-29.3236
3	########	0.000382	38.17623	5.73588	12.23534	-0.47814	-31.3377
4	########	0.00049	49.01929	5.292308	12.23133	-0.47815	-31.1883
5	########	0.000425	42.53187	5.564445	12.22732	-0.47816	-22.2278
6	########	0.000372	37.16645	5.836582	12.2233	-0.47817	-25.4764
7	########	0.000328	32.80454	6.108719	12.21929	-0.47817	-35.8977
8	########	0.000309	30.90853	6.307732	12.21528	-0.47818	-26.101
9	########	0.000359	35.85742	6.074213	12.21127	-0.47819	-14.6255
10	########	0.000336	33.58371	6.279366	12.21239	-0.47819	-25.8846
11	########	0.000391	39.07307	6.036408	12.2534	-0.4782	-22.4882

Sheet4 Sheet1 Sheet2 Sheet3

Develop Macro via Recording

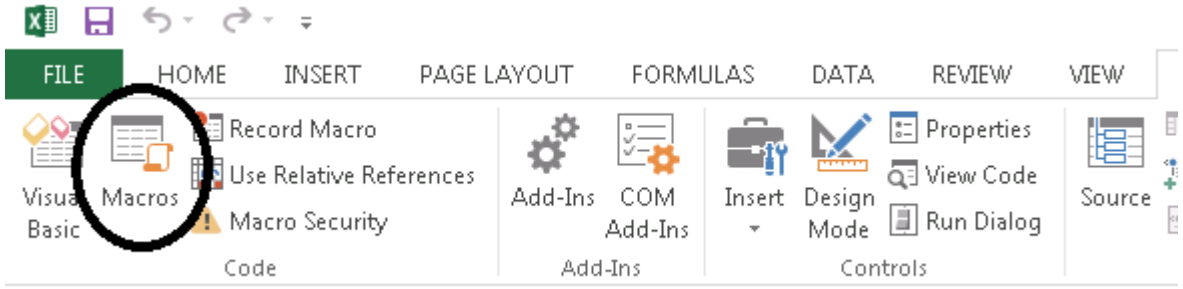
A7 – Go back to Sheet 1, click any cell, and click Stop Recording



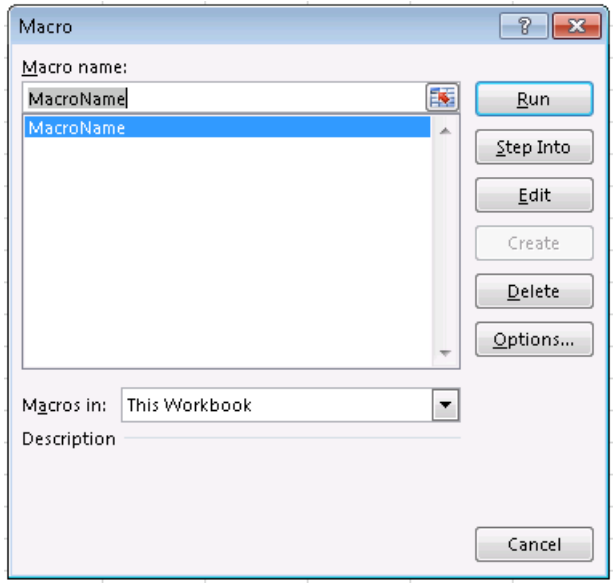
The Macro for Application 1 has been developed. To apply the Macro, we use following steps:

Develop Macro via Recording

A8 – Find your developed Macro by clicking Macros



A9 – Select Macro your created and click Run.



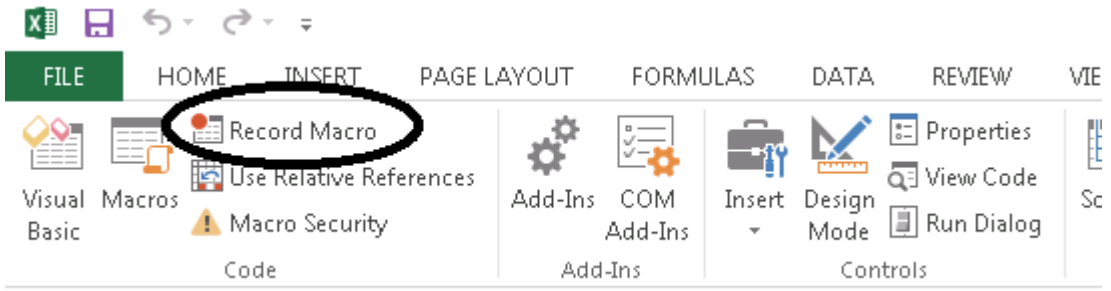
Each time, the Macro repeats your recorded operations

Develop Macro via Recording

Application 2: Repeat estimations of statistical metrics for data in each sheet

Continue this application based on previous dataset

B1 – Find the developer tag and click Record Macro button, following you need to provide a name to the new Macro and click ok



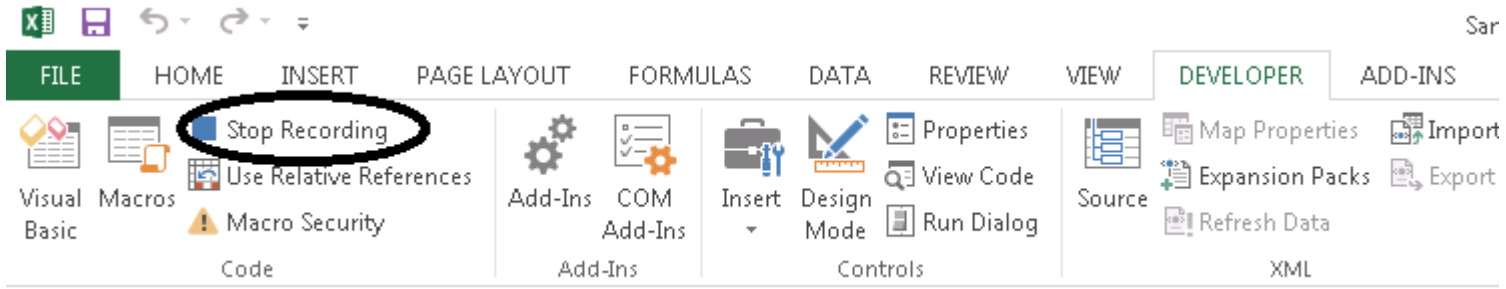
Develop Macro via Recording

B2 – Select a cell to start your operations in estimating mean, standard deviation, maximum, and minimum for each parameter, e.g.:

	A	B	C	D	E	F	G	H
1	Time	Response	Response	Input 1	Input 2	Input 3	Input 4	
2	10/1/08 12:03 AM	0.000353542	35.3542	5.727265	12.23935	-0.47814	-29.3236	
3	10/1/08 12:03 AM	0.000381762	38.17623	5.73588	12.23534	-0.47814	-31.3377	
4	10/1/08 12:03 AM	0.000490193	49.01929	5.292308	12.23133	-0.47815	-31.1883	
5	10/1/08 12:03 AM	0.000425319	42.53187	5.564445	12.22732	-0.47816	-22.2278	
6	10/1/08 12:03 AM	0.000371664	37.16645	5.836582	12.2233	-0.47817	-25.4764	
7	10/1/08 12:04 AM	0.000328045	32.80454	6.108719	12.21929	-0.47817	-35.8977	
8	10/1/08 12:04 AM	0.000309085	30.90853	6.307732	12.21528	-0.47818	-26.101	
9	10/1/08 12:04 AM	0.000358574	35.85742	6.074213	12.21127	-0.47819	-14.6255	
10	10/1/08 12:04 AM	0.000335837	33.58371	6.279366	12.21239	-0.47819	-25.8846	
11	10/1/08 12:04 AM	0.000390731	39.07307	6.036408	12.2534	-0.4782	-22.4882	
12								
13	Average	0.000374475	37.44753	5.896292	12.22683	-0.47817	-26.4551	
14	Standard deviatio	5.26164E-05	5.261643	0.324076	0.013369	2.14E-05	5.949696	
15	Maximum	0.000490193	49.01929	6.307732	12.2534	-0.47814	-14.6255	
16	Minimum	0.000309085	30.90853	5.292308	12.21127	-0.4782	-35.8977	
17								

Develop Macro via Recording

B3 – Click any cell and stop the Macro Recording

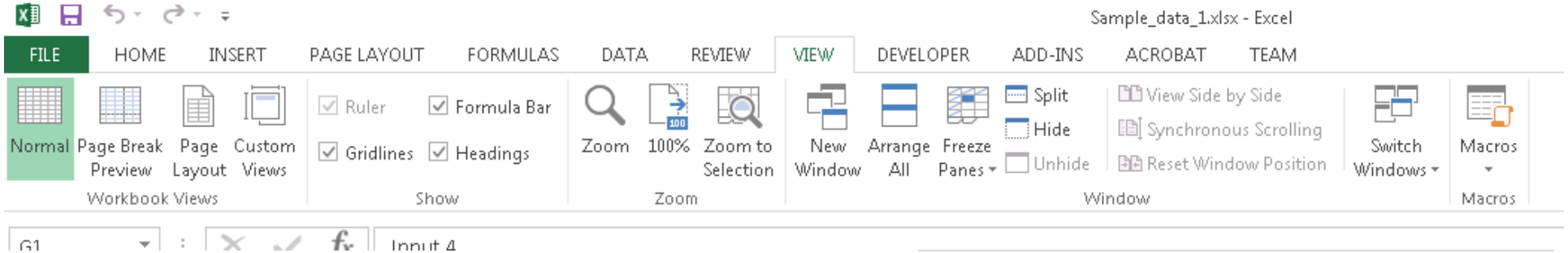


B4 – Go to other worksheets and run the developed Macro, in this application, each run of Macro will repeat the estimation of statistical metrics for data in other worksheets

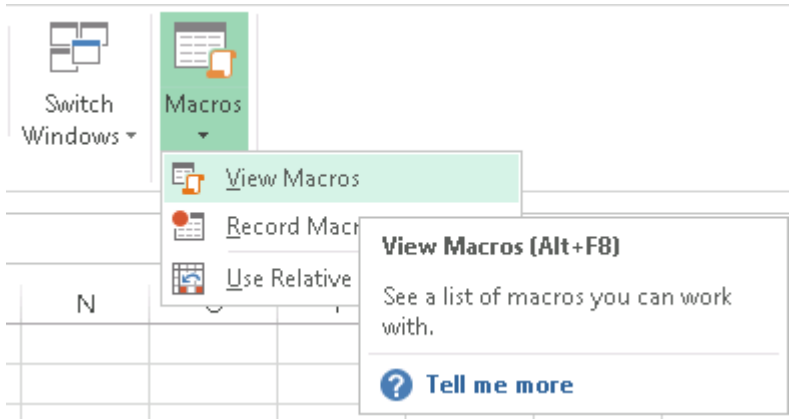
Develop Macro via Visual Basic

- Find the Visual Basic programming interface

Step 1 – Find the View tag and select Macros

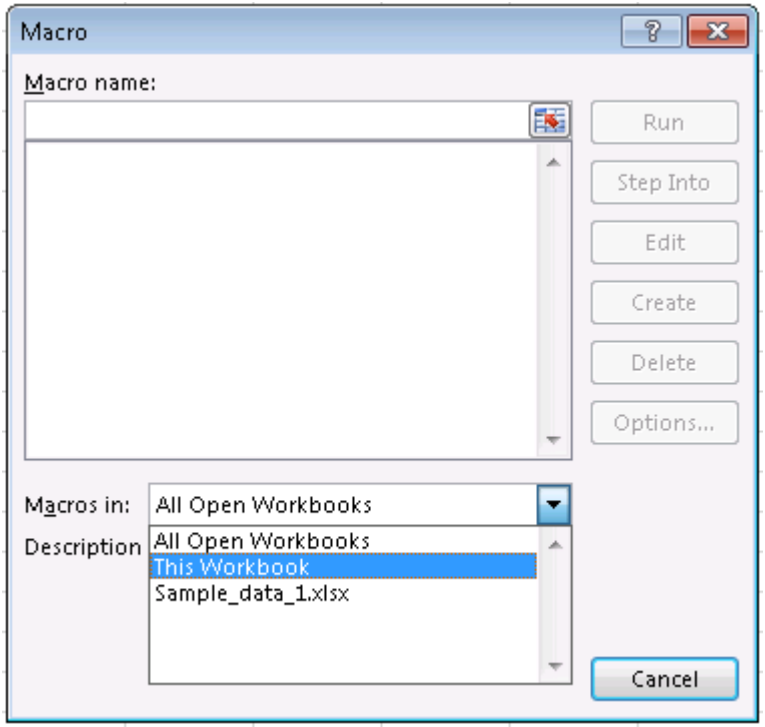


Step 2 – Click view Macros



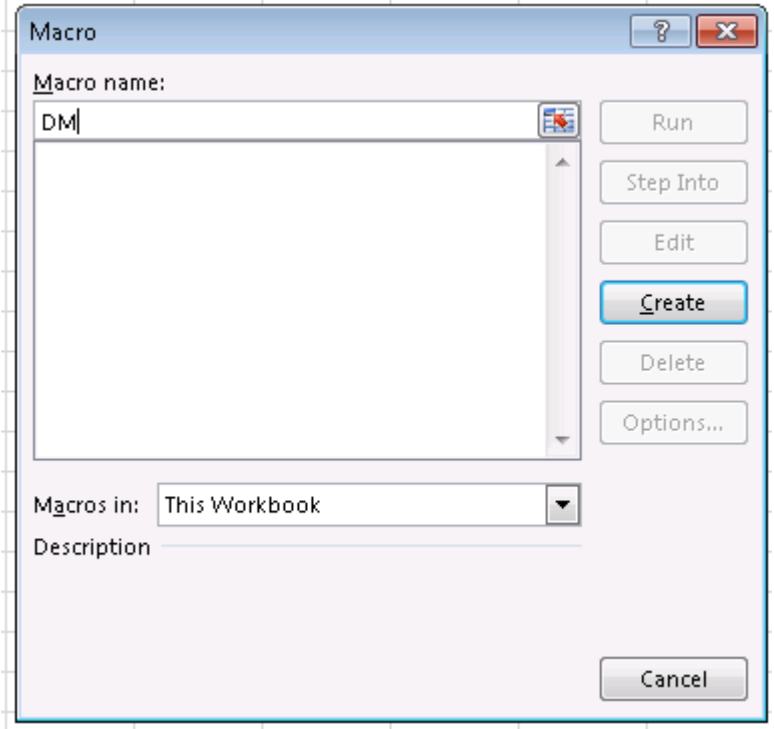
Develop Macro via Visual Basic

Step 3 – Select This Workbook



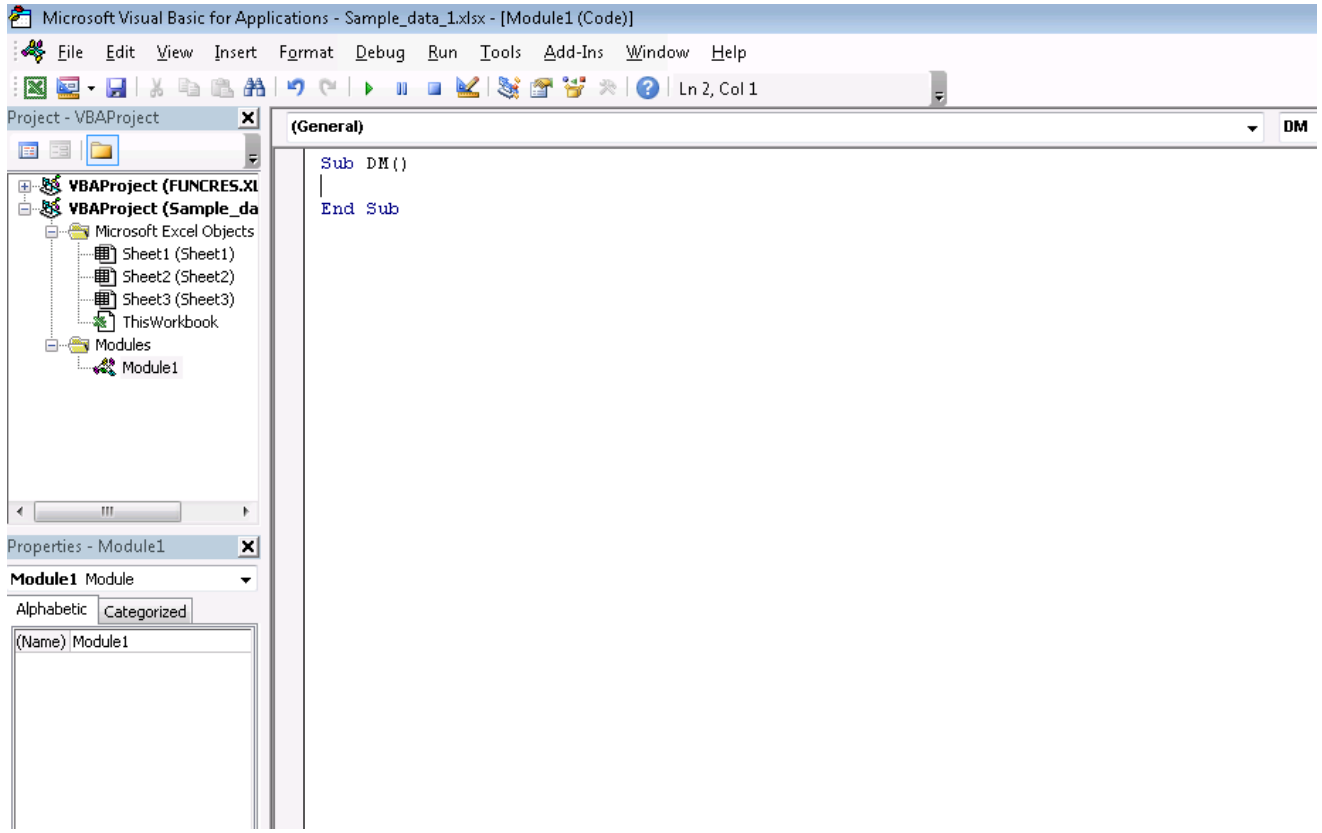
Develop Macro via Visual Basic

Step 4 – Type a name for the new Macro and click Create, e.g.:



Develop Macro via Visual Basic

Step 5 – The programming interface will pop-out



You can program your unique Macro in the interface

Develop Macro via Visual Basic

Some VB syntax

Declare your variable types:

Dim x As Long (Define variable x as integer)

Dim x As Double (Define variable x as real numbers)

Dim x As Boolean (Define variable x as Boolean)

Dim x As String (Define variable x as character strings)

Develop Macro via Visual Basic

Some VB Syntax

Statement of fetch the name of current active workbook:

```
ThisToolName = ActiveWorkbook.Name
```

Develop Macro via Visual Basic

Some VB Syntax

Statement of activating a specified cell of a worksheet

Worksheets(i).Cells(x, y).Activate

i is the index of the worksheet and x,y describe the coordinate of the cell

Develop Macro via Visual Basic

Some VB Syntax

Statements of counting the number of columns and number of rows in the currently activated worksheet

ActiveCell.CurrentRegion.rows.Count

ActiveCell.CurrentRegion.columns.Count

Develop Macro via Visual Basic

Some VB Syntax

Statements of passing the value of an excel cell to a variable

Value = Workbooks(Name).Worksheets(i).Cells(x, y)

Develop Macro via Visual Basic

Some VB Syntax

The if…then statement:

```
If  Then
End If
```

The for loop statement:

```
For i = 1 to 10
Next i
```

Develop Macro via Visual Basic

- Application 3: Integrate data with same format in three files into one file

Used datasets:

Sample_data_1.xlsx

Sample_data_2.xlsx

Sample_data_3.xlsx

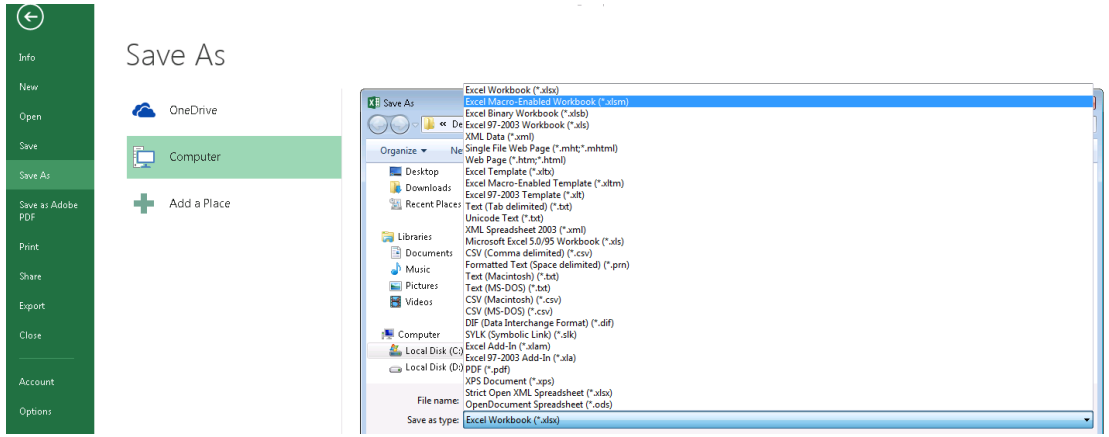
Develop Macro via Visual Basic

Before coding, we need to prepare following items:

Item 1 – Create a macro-enabled excel file by

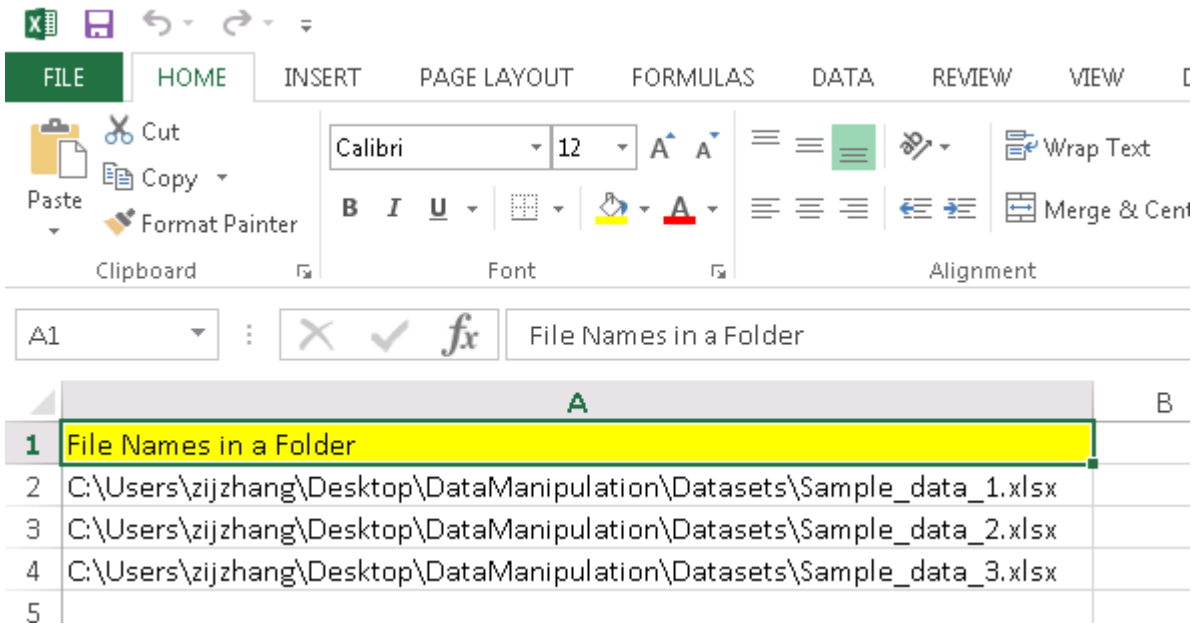
S1: Create a new excel called “Data_Manipulator.xlsx”

S2: Open this excel and save as:



Develop Macro via Visual Basic

In the xlsm file, create a sheet to locate a region containing information of paths for accessing data files to be integrated, e.g.:



Your should have your own paths and cannot directly copy this info.

Develop Macro via Visual Basic

Code Part 1: Declare Variables

```
Dim i, j, k, x As Long
Dim rows, columns As Long
Dim Pointer As Long
Dim Filename As String
Dim ThisToolName, OriginalData, NewData As String
```

Develop Macro via Visual Basic

Code Part 2: Specification of Currently Activated Workbook for integrating sub-datasets

```
ThisToolName = ActiveWorkbook.Name
Worksheets(1).Cells(1, 1).Activate
     rows = ActiveCell.CurrentRegion.rows.Count
     columns = ActiveCell.CurrentRegion.columns.Count
     Pointer = 2
```

Develop Macro via Visual Basic

Code Part 3: Commands for asking excels to collect data from 3 sub-datasets

```
Filename = Workbooks(ThisToolName).Worksheets(1).Cells(i, 1)
If Cells(i, 1) <> Empty Then
   Workbooks.Open Filename
   OriginalData = ActiveWorkbook.Name
   Worksheets(1).Cells(1, 1).Activate
   OriginalDatarows = ActiveCell.CurrentRegion.rows.Count
   OriginalDatacolumns = ActiveCell.CurrentRegion.columns.Count

   For k = 2 To OriginalDatarows
       For j = 1 To OriginalDatacolumns
             Workbooks(ThisToolName).Worksheets("NewData").Cells(Pointer, j) = Workbooks(OriginalData).Worksheets(1).Cells(k, j)
       Next j
       Pointer = Pointer + 1
   Next k

   Workbooks(OriginalData).Close SaveChanges:=False
End If
```

Develop Macro via Visual Basic

Q&A