Hierarchical Clustering

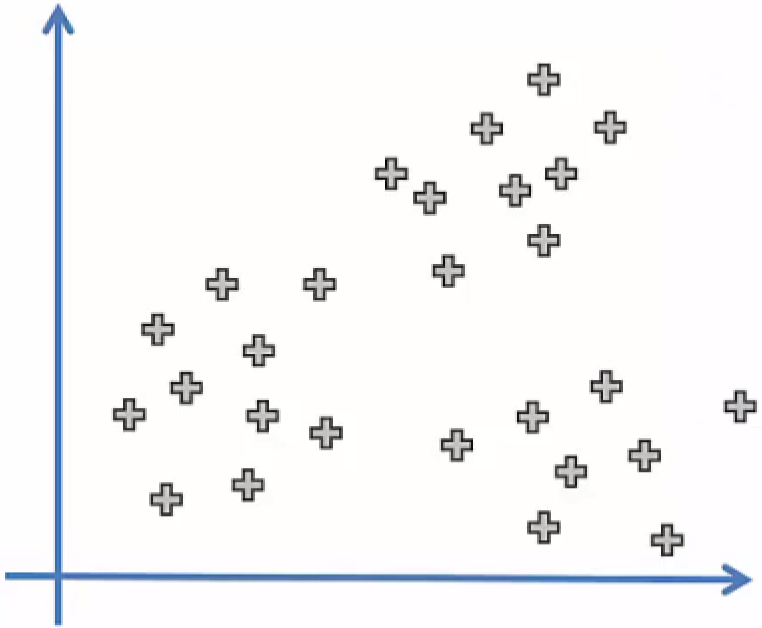
In Machine Learning

Hierarchical Clustering

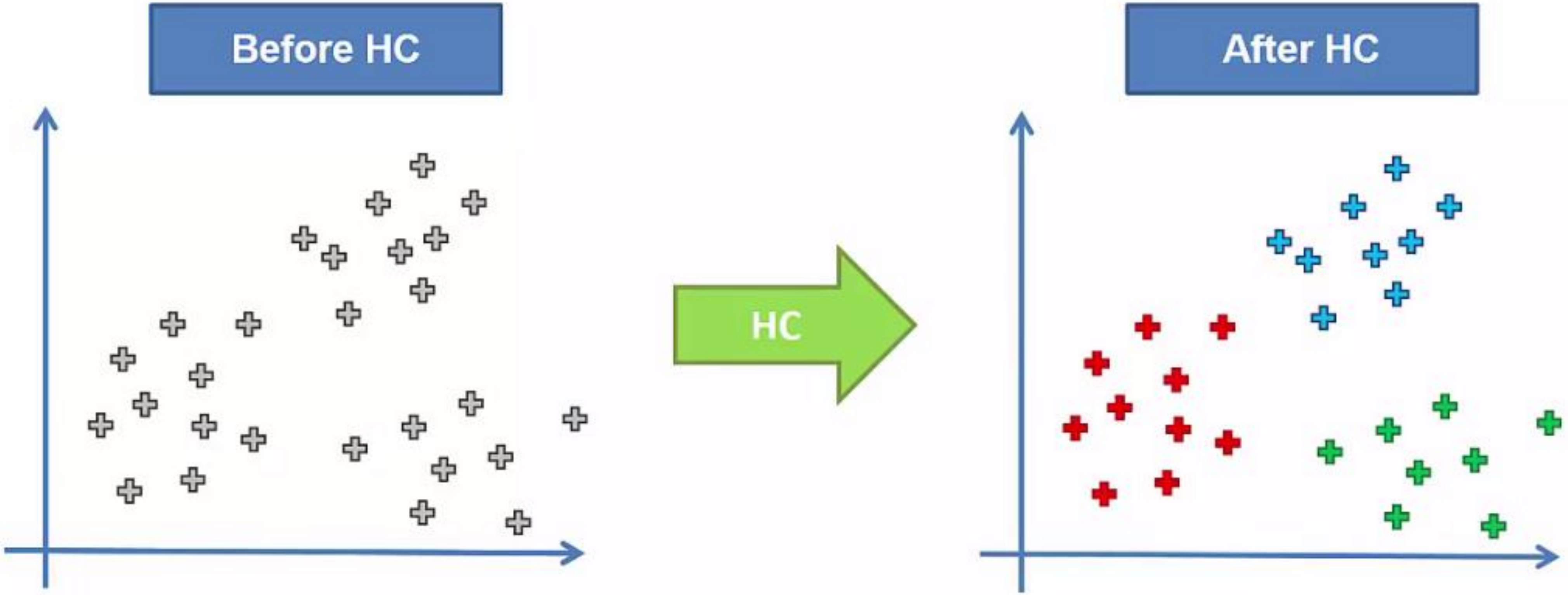
Find successive clusters using previously established clusters

What HC does for you

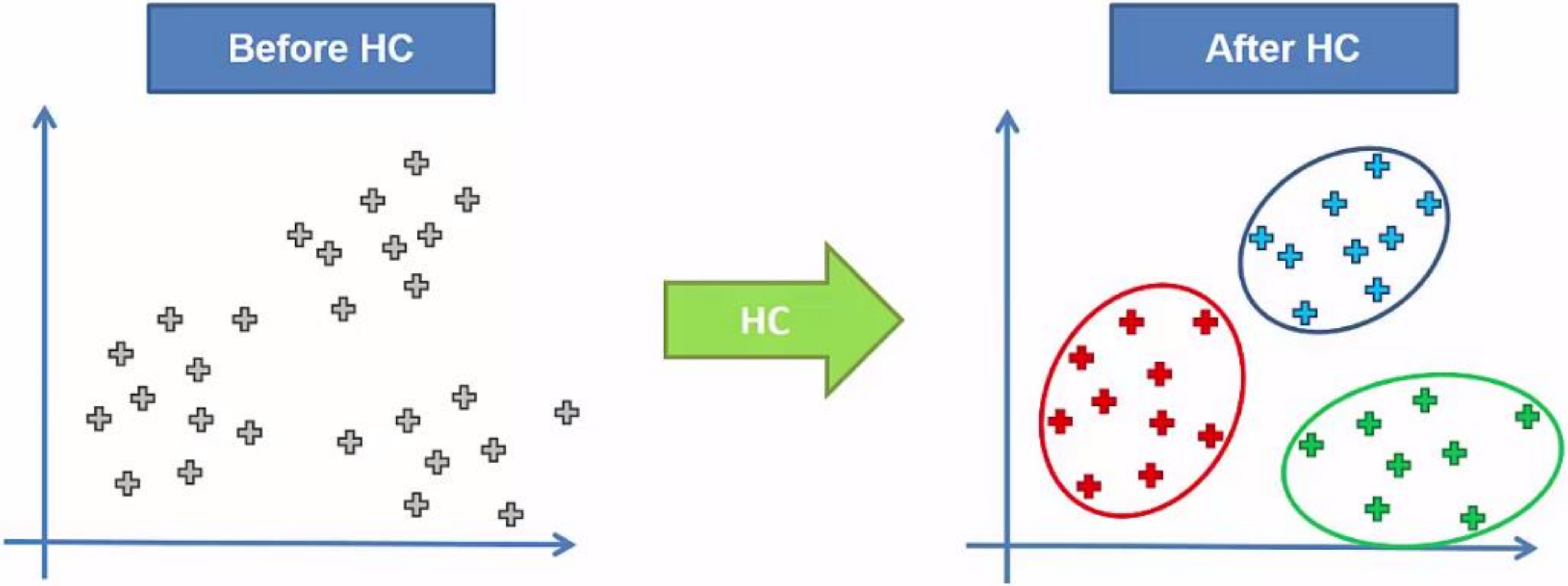
Before HC



What HC does for you



What HC does for you



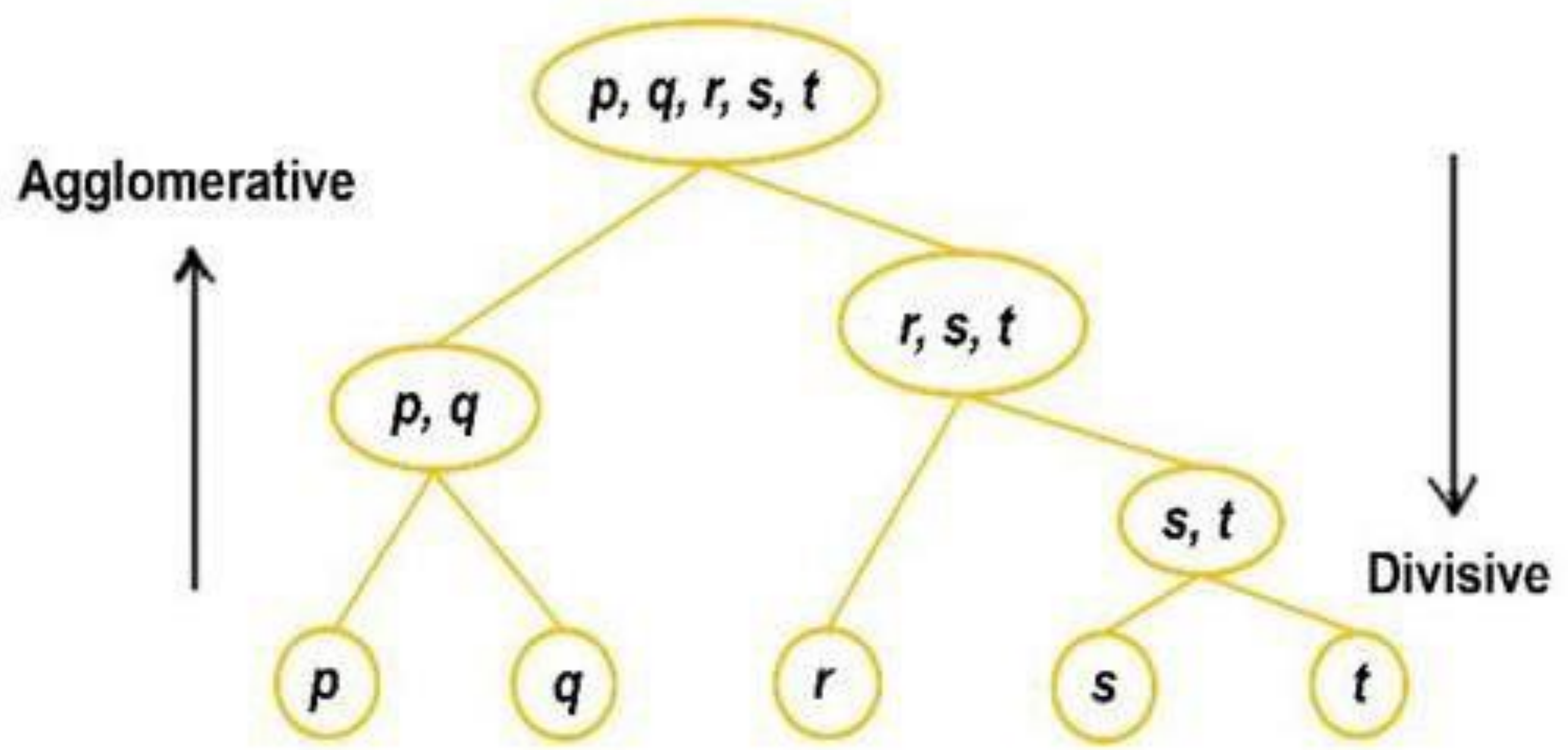
Same as K-Means but different process

Hierarchical Clustering

- Two types
  - Agglomerative (bottom-up)
  - Divisive (top-down)

- Agglomerative Algorithm
  - Begin with each data element as a separate cluster and merge them into successively larger cluster

- Divisive Algorithm
  - Begins with the whole set and proceed to divide it into successive smaller clusters


p, q, r, s, t
Agglomerative
r, s, t
p, q
s, t
Divisive
p
q
r
s
t

Applications

- **Bioinformatics**: grouping animals according to their biological features to reconstruct phylogeny trees
- **Business**: dividing customers into segments or forming a hierarchy of employees based on salary.
- **Image processing**: grouping handwritten characters in text recognition based on the similarity of the character shapes.
- **Information Retrieval**: categorizing search results based on the query

Agglomerative HC

STEP 1: Make each data point a single-point cluster ➡ That forms N clusters

STEP 2: Take the two closest data points and make them one cluster ➡ That forms N-1 clusters

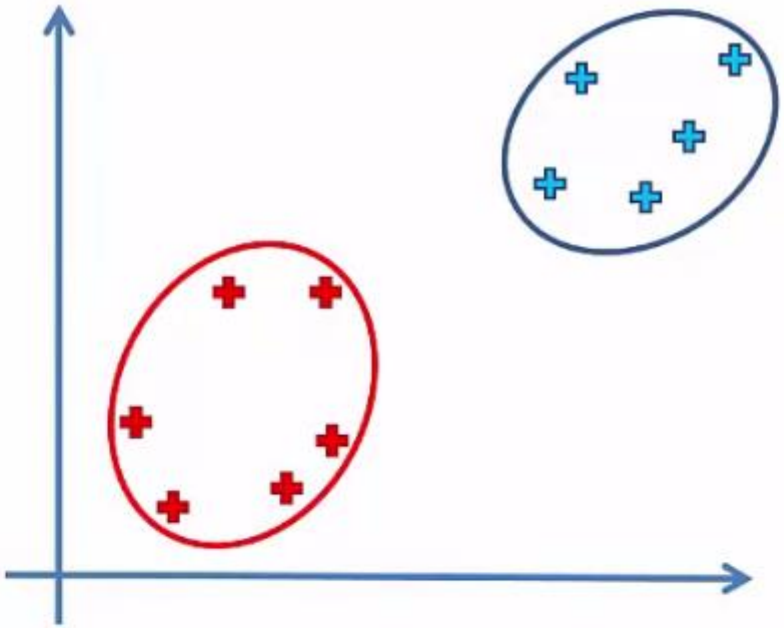
STEP 3: Take the two closest clusters and make them one cluster ➡ That forms N - 2 clusters

STEP 4: Repeat STEP 3 until there is only one cluster

FIN

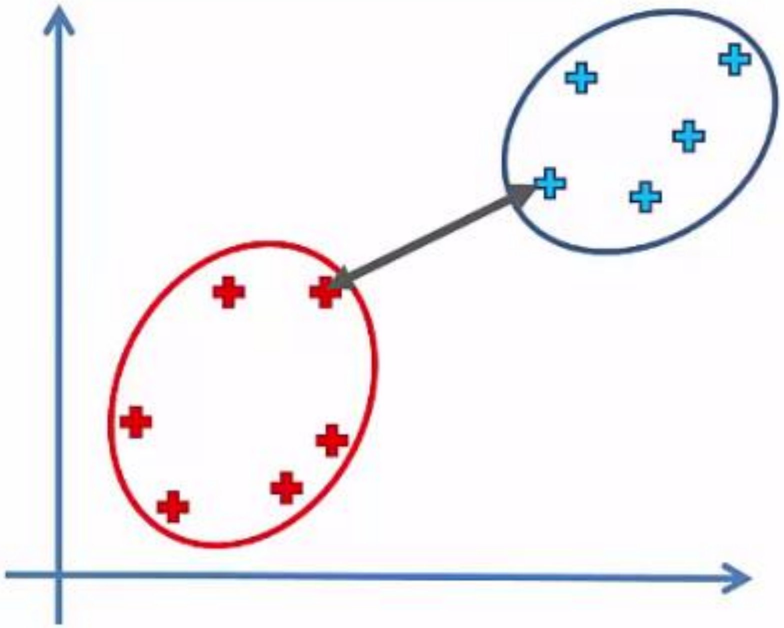
- Closest clusters - Euclidean distance or Manhattan distance

Distance Between Clusters



Distance Between Two Clusters:

Distance Between Clusters

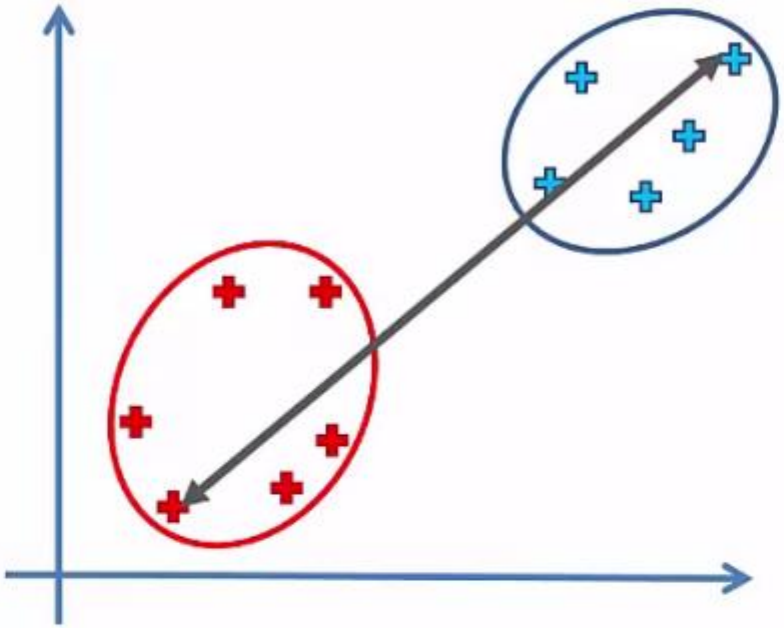


Distance Between Two Clusters:

- Option 1: Closest Points

Min (Single) Linkage

Distance Between Clusters

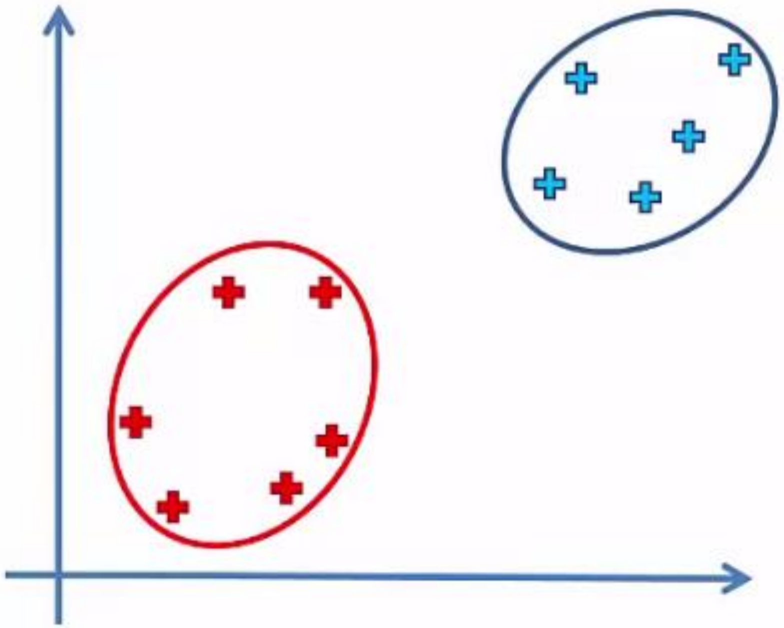


Distance Between Two Clusters:

- Option 1: Closest Points
- Option 2: Furthest Points

Max (Complete) Linkage

Distance Between Clusters

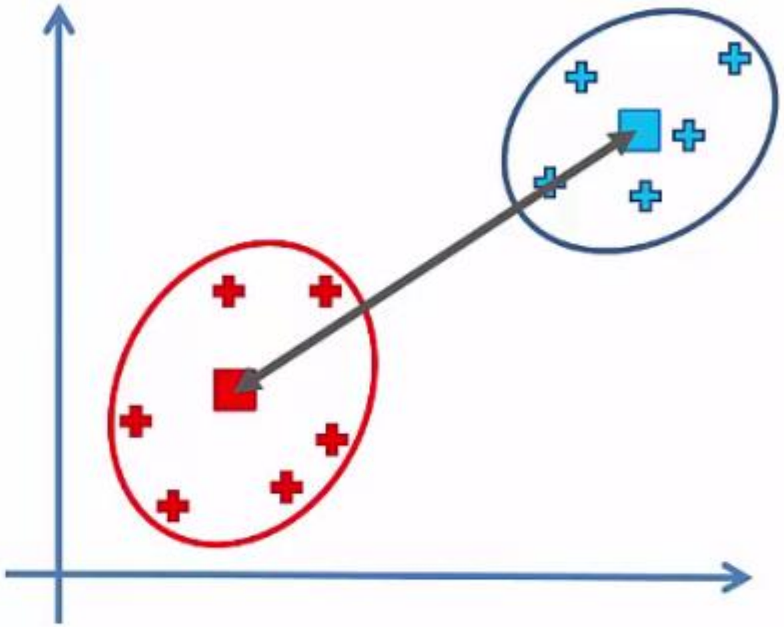


Distance Between Two Clusters:

- Option 1: Closest Points
- Option 2: Furthest Points
- Option 3: Average Distance

Average Linkage

Distance Between Clusters



Distance Between Two Clusters:

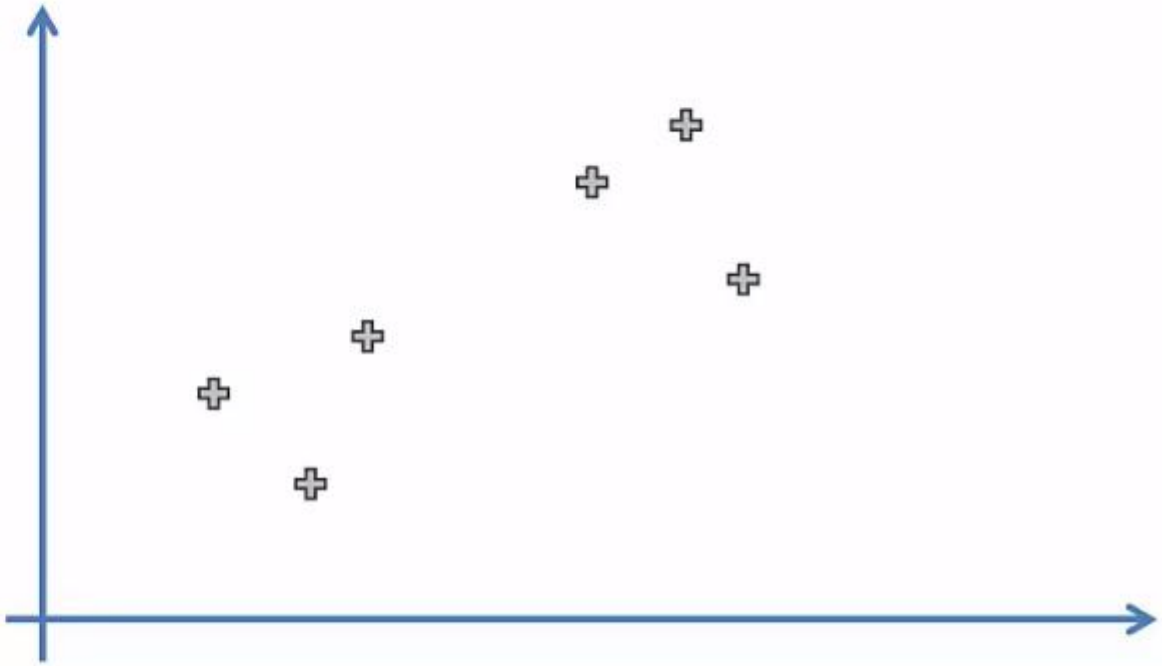
- Option 1: Closest Points
- Option 2: Furthest Points
- Option 3: Average Distance
- Option 4: Distance Between Centroids

Centroid Linkage

Ward Linkage

Agglomerative HC

Consider the following dataset of N = 6 data points

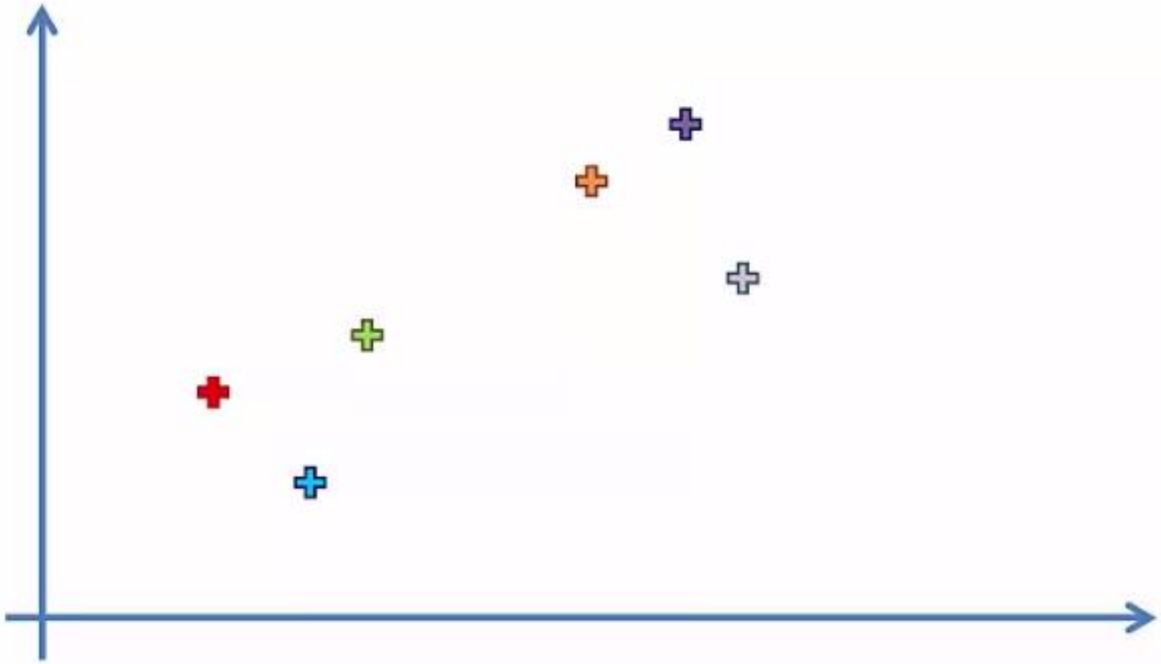


Agglomerative HC

STEP 1: Make each data point a single-point cluster ➡ That forms 6 clusters

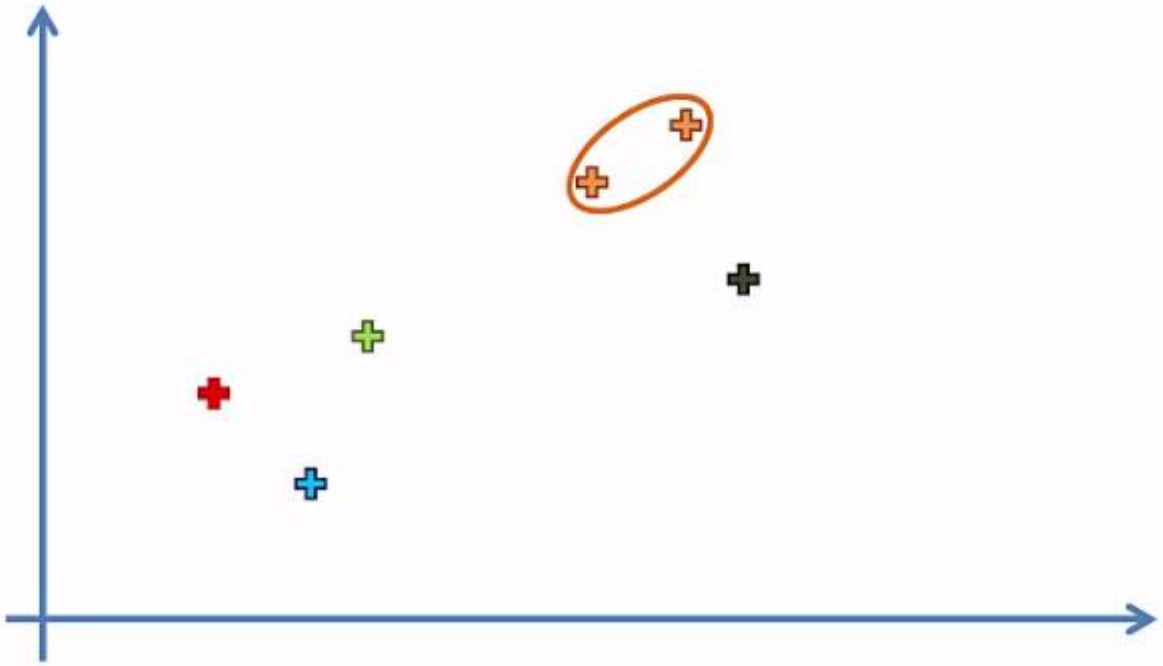
Agglomerative HC

STEP 1: Make each data point a single-point cluster ➡ That forms 6 clusters



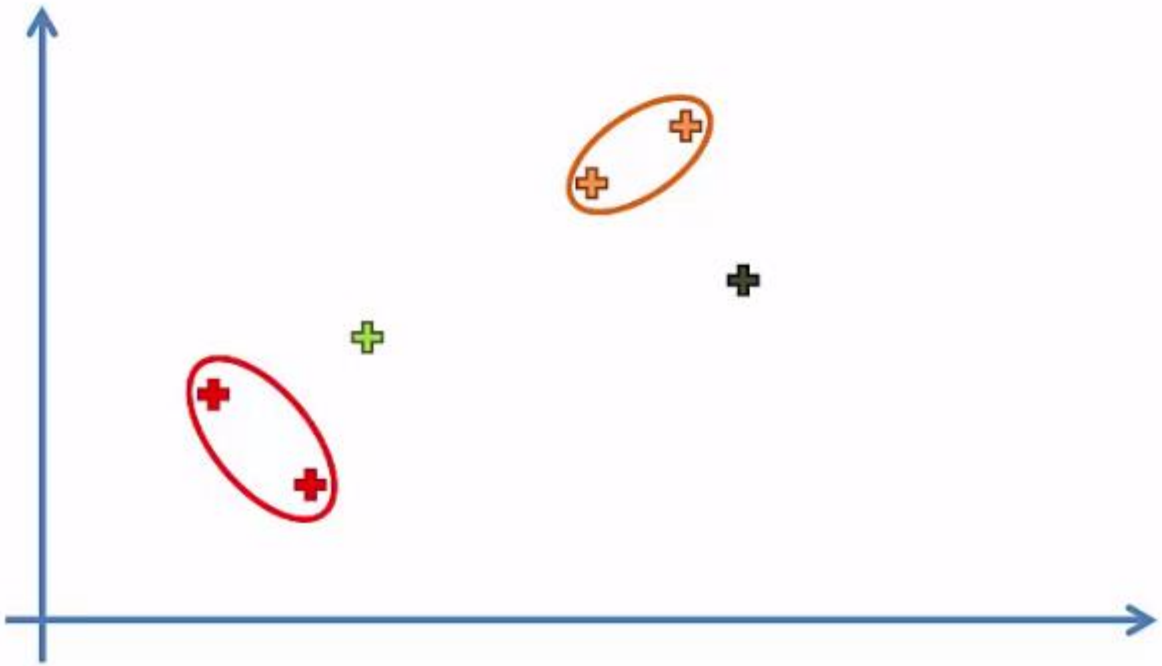
Agglomerative HC

STEP 2: Take the two closest data points and make them one cluster
➡ That forms 5 clusters



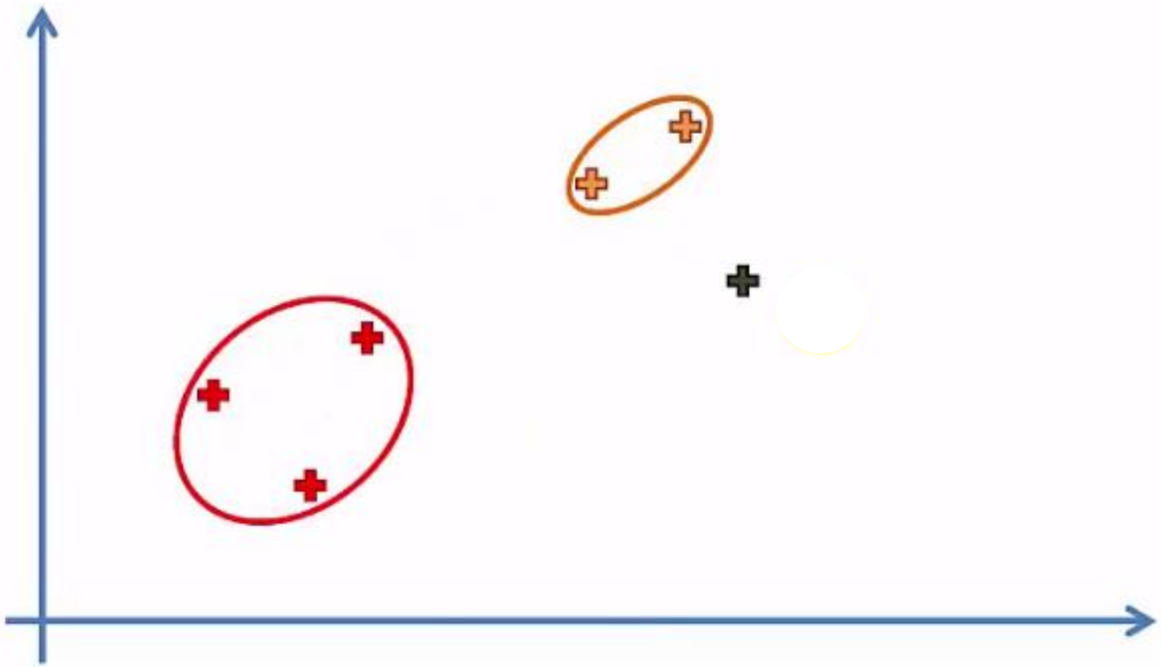
Agglomerative HC

STEP 3: Take the two closest clusters and make them one cluster
➡ That forms 4 clusters



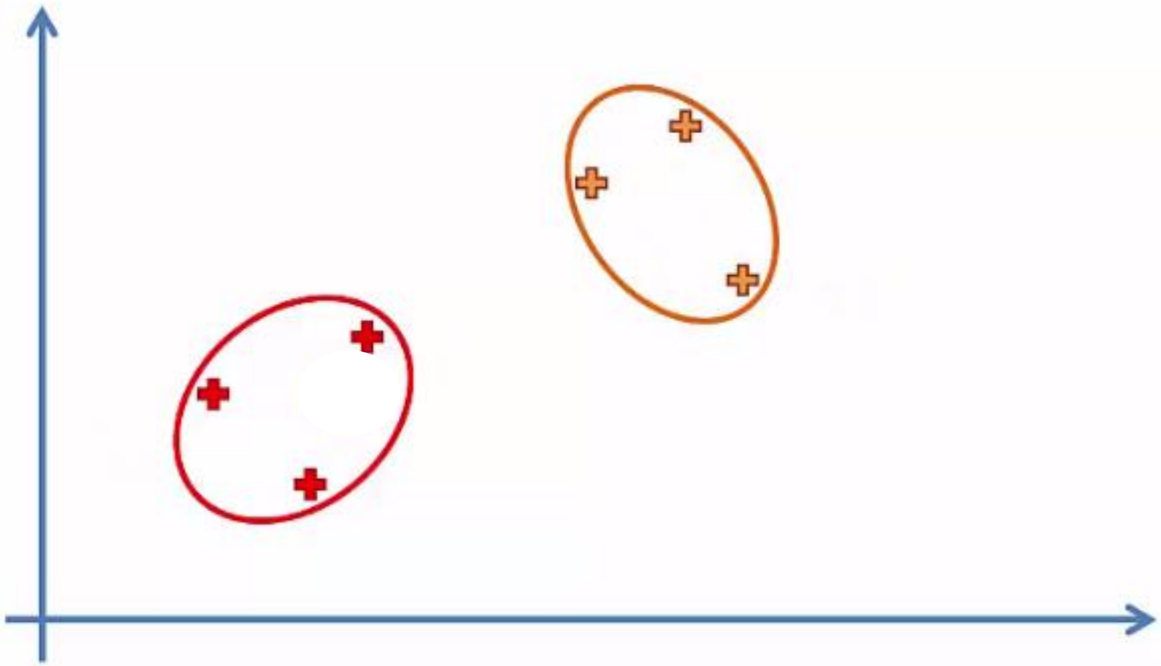
Agglomerative HC

STEP 4: Repeat STEP 3 until there is only one cluster



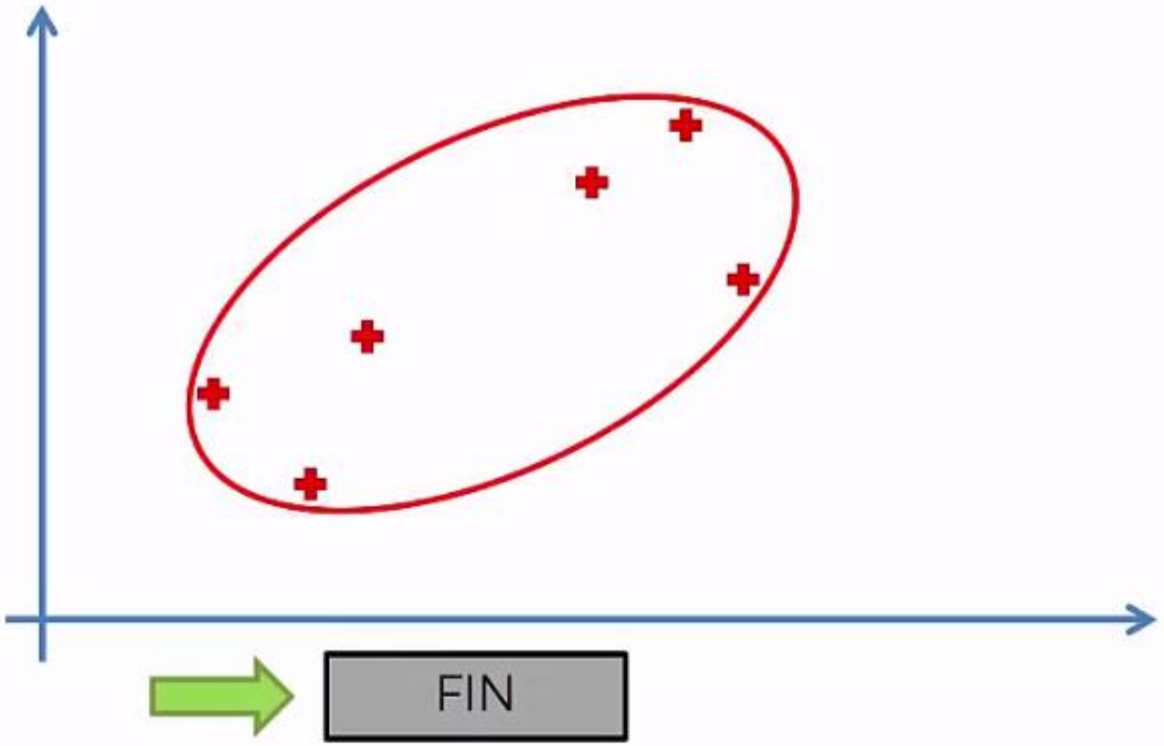
Agglomerative HC

STEP 4: Repeat STEP 3 until there is only one cluster



Agglomerative HC

STEP 4: Repeat STEP 3 until there is only one cluster

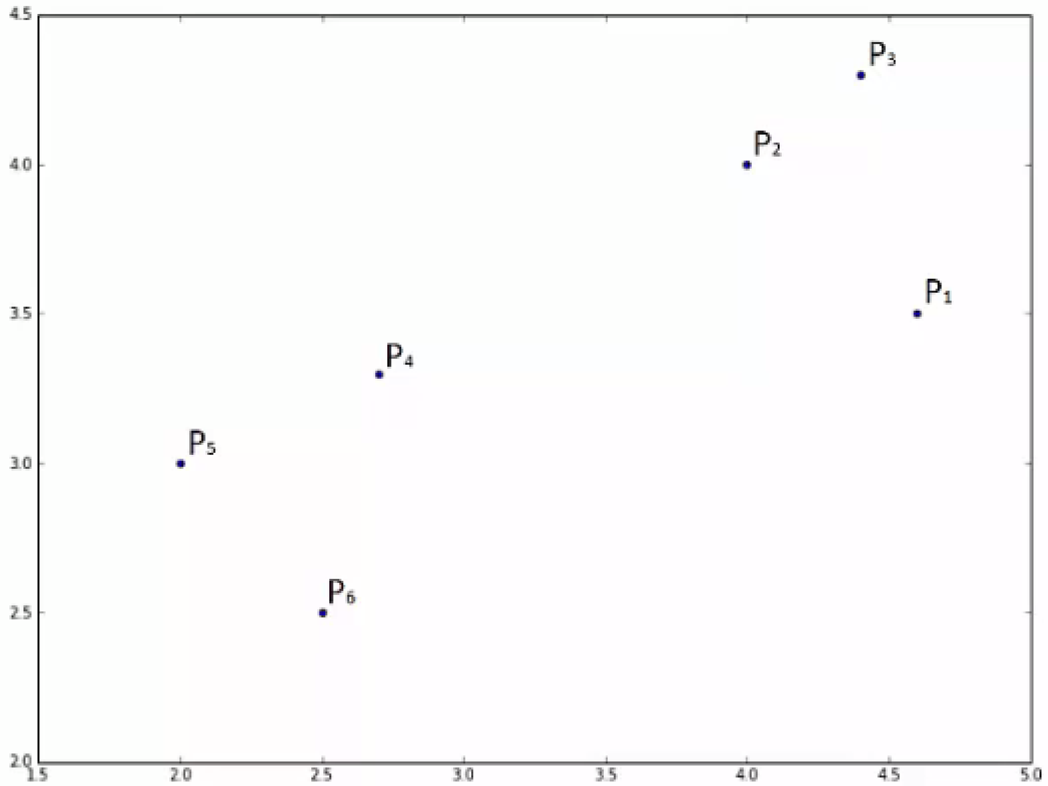


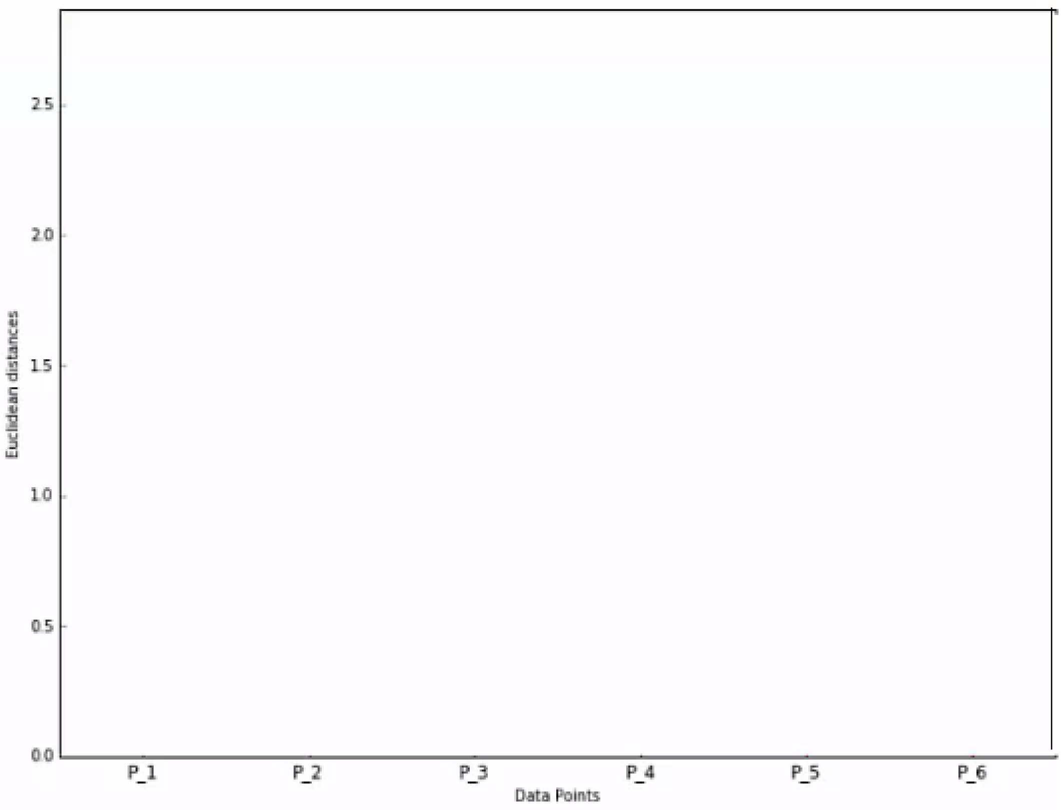
Hierarchical Clustering

Dendrograms

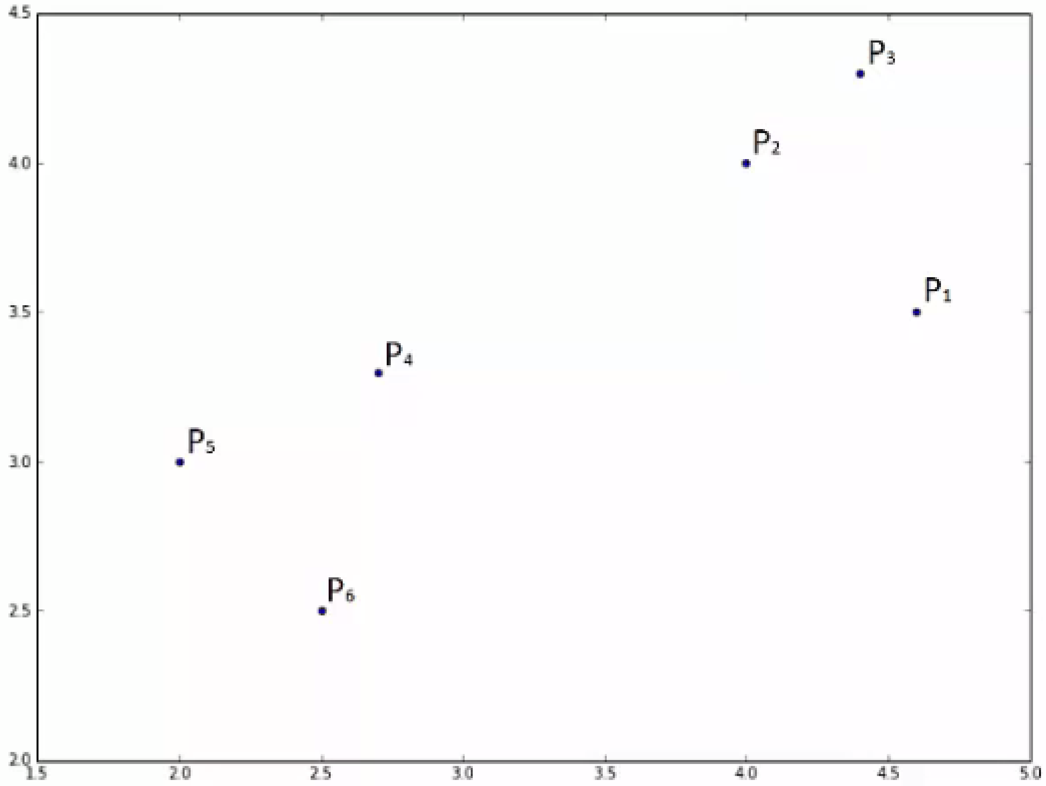
How do Dendrograms Works?

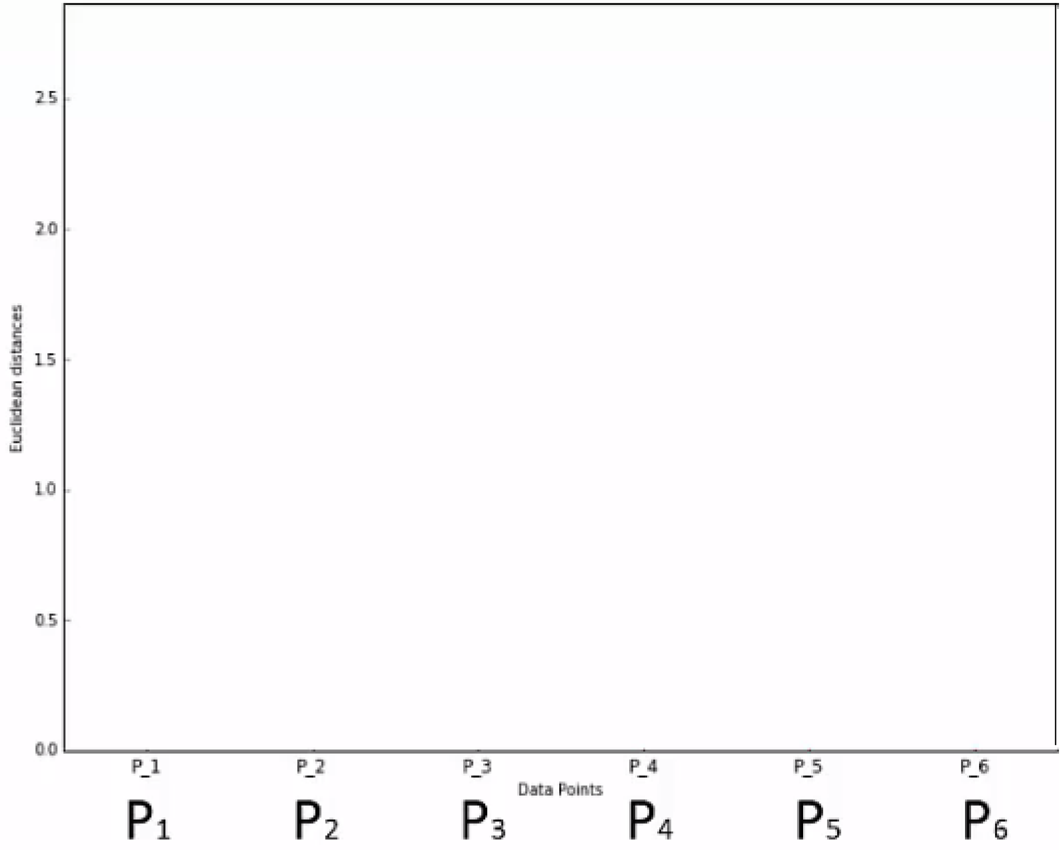
How do Dendrograms Works



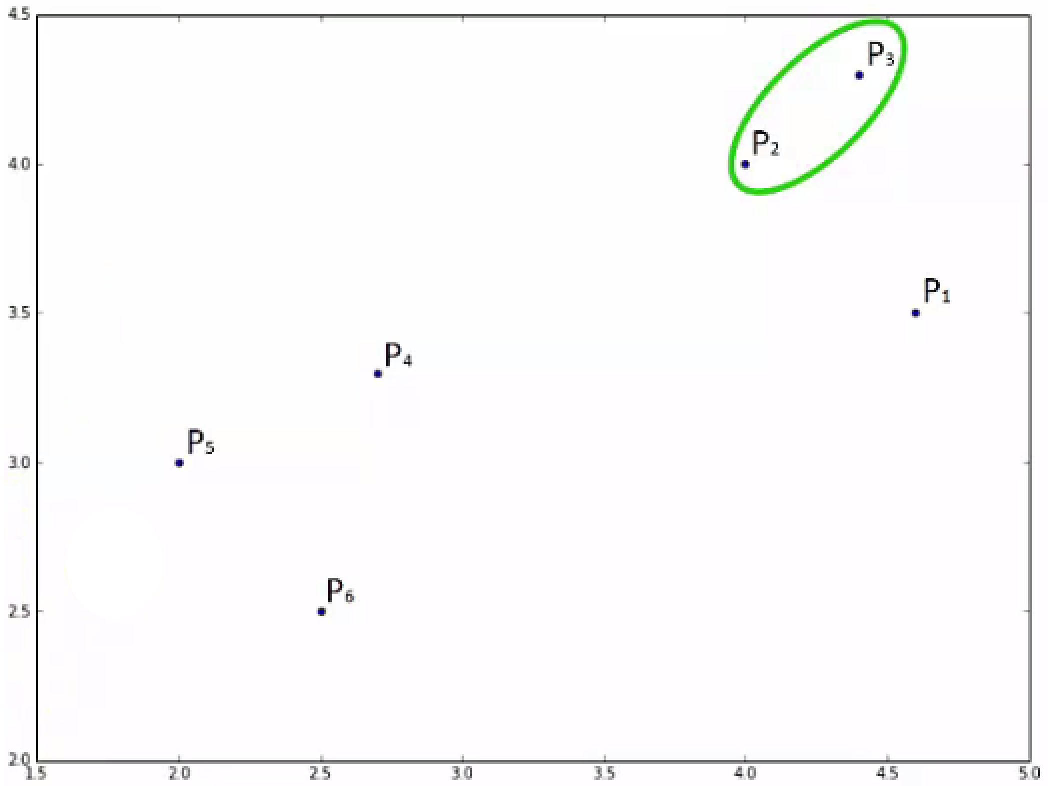


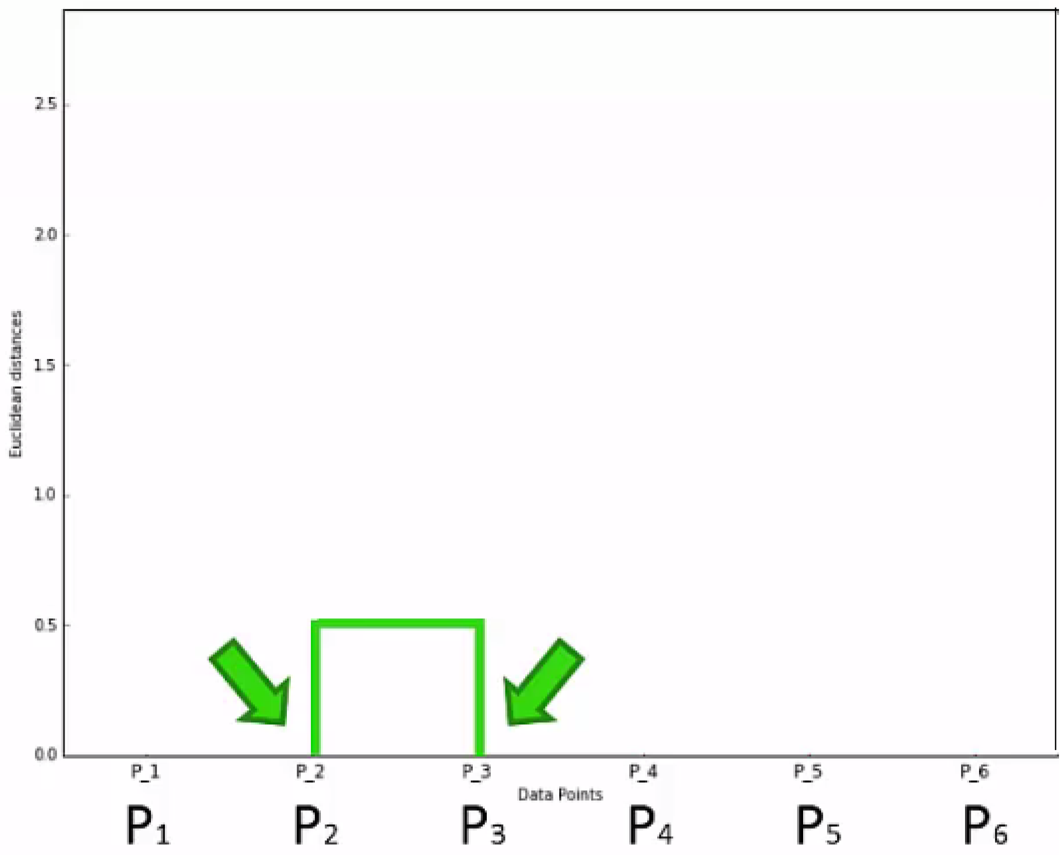
How do Dendrograms Works



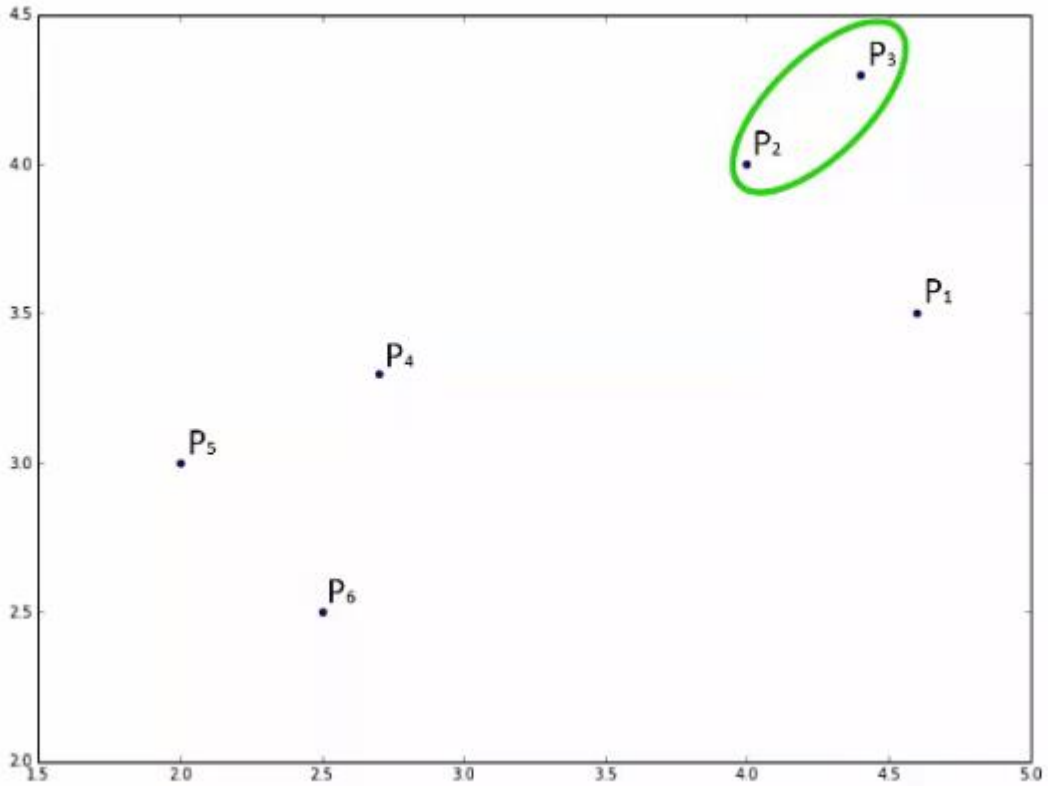


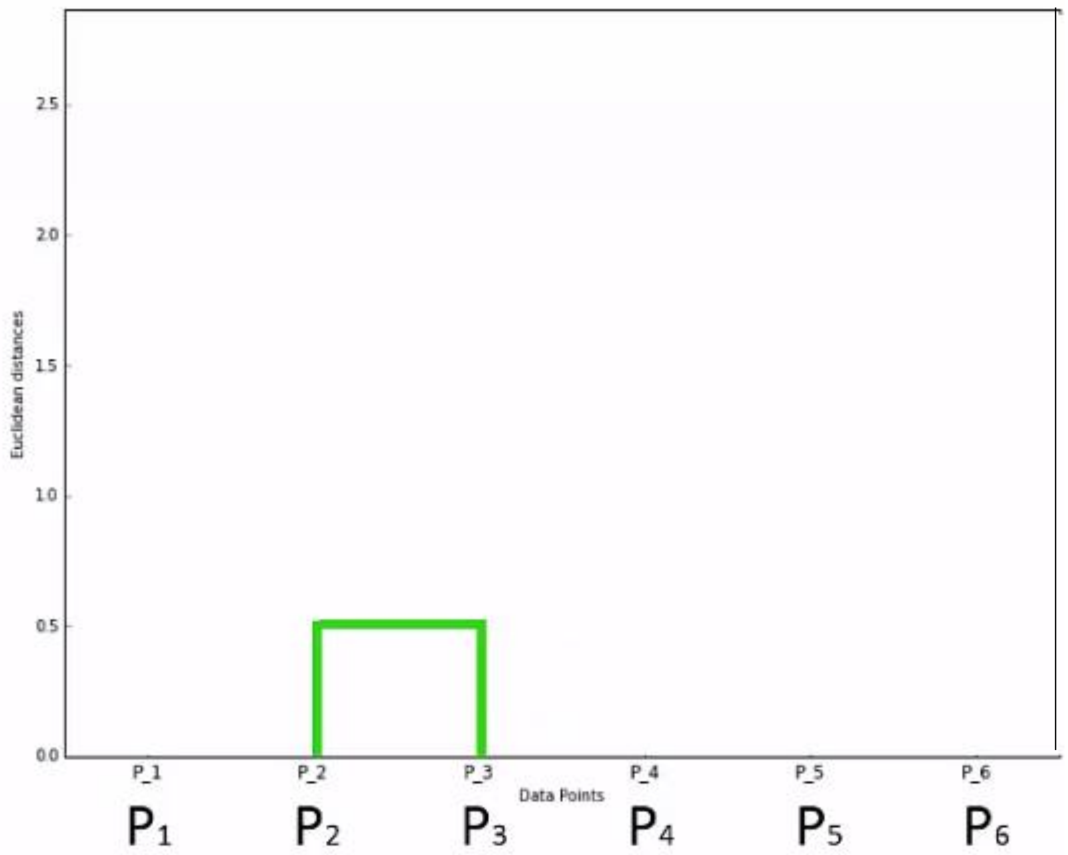
How do Dendrograms Works



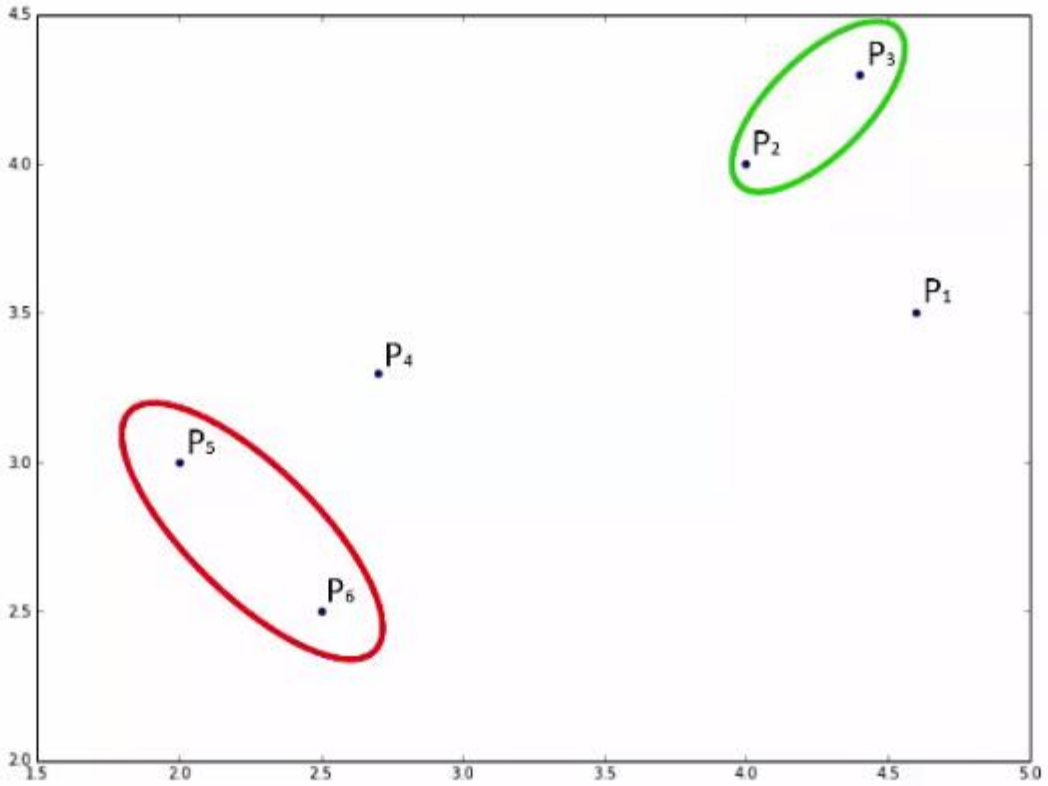


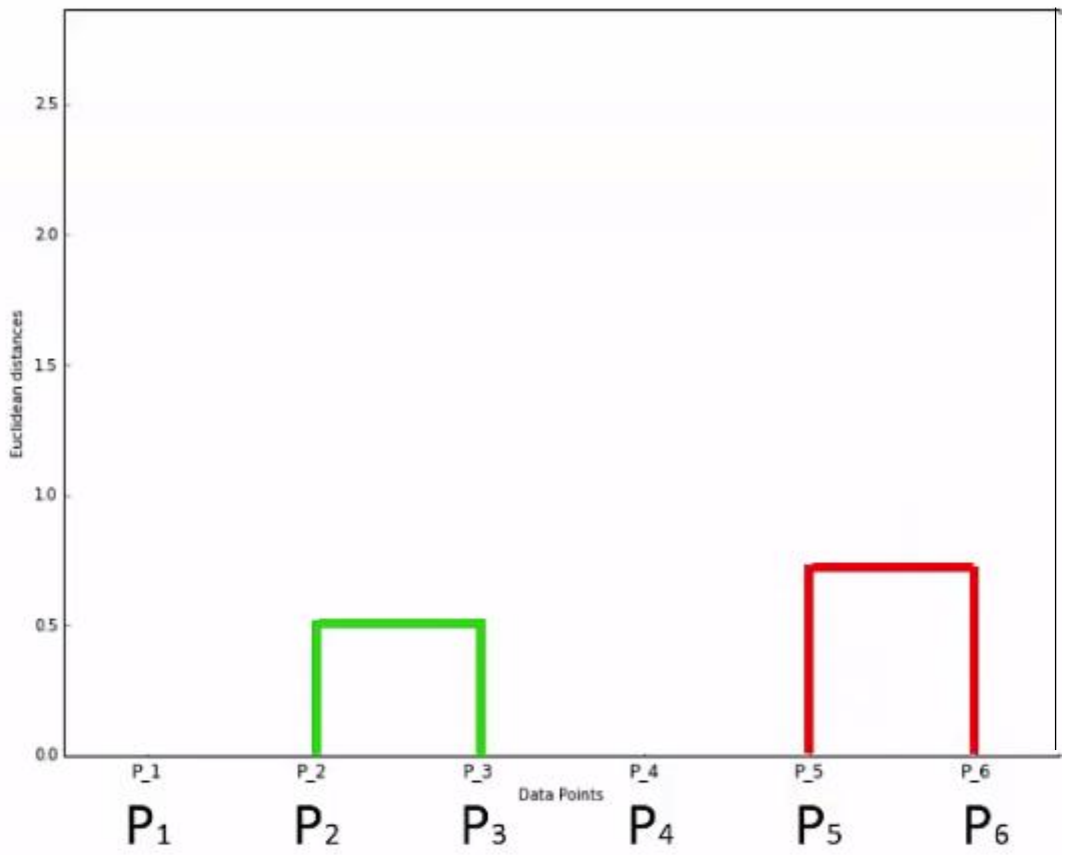
How do Dendrograms Works



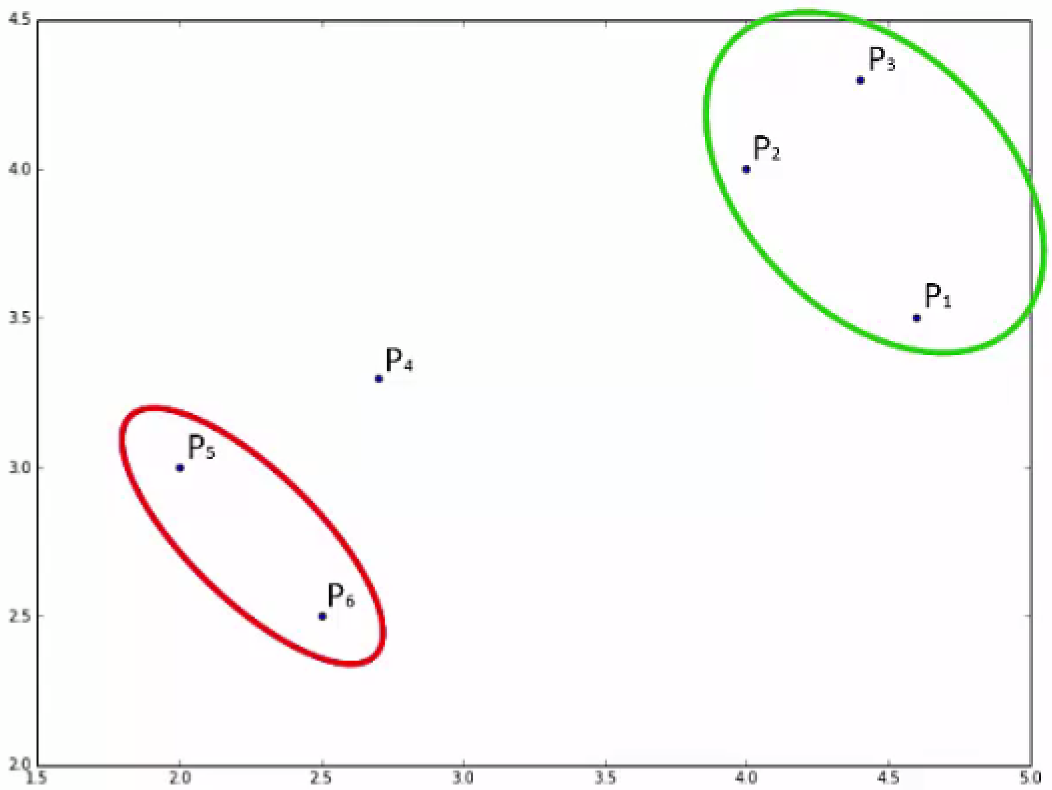


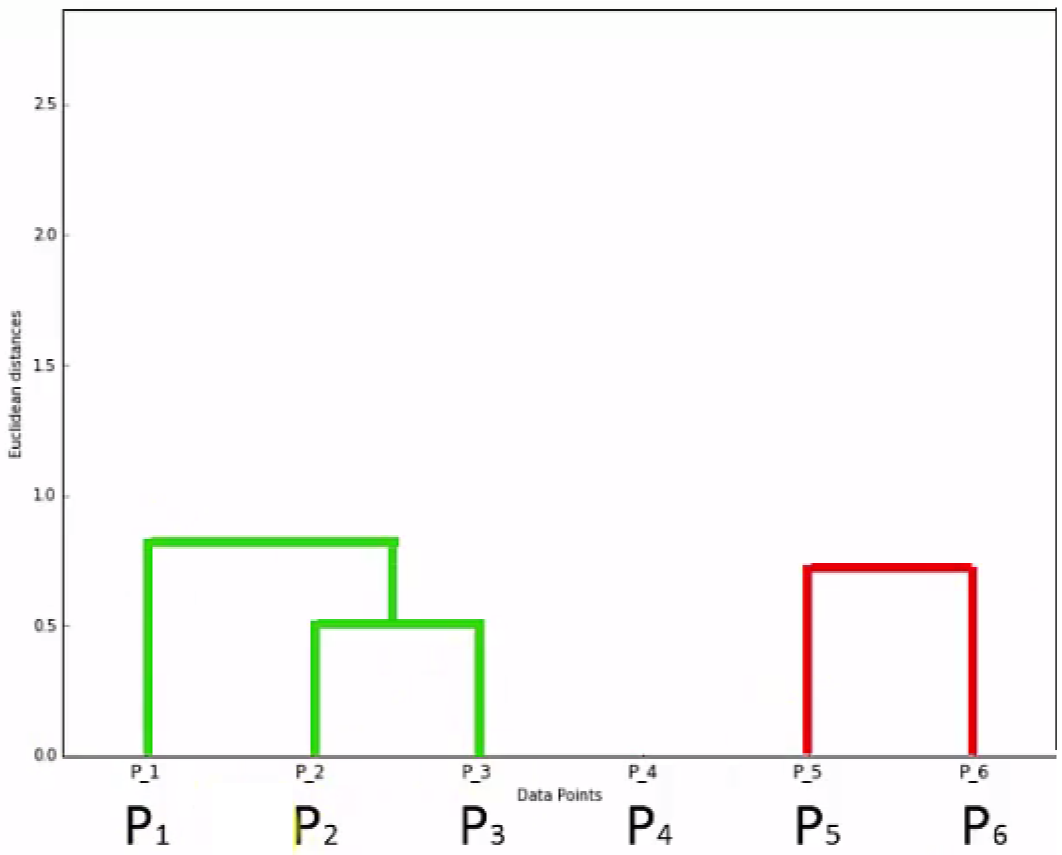
How do Dendrograms Works



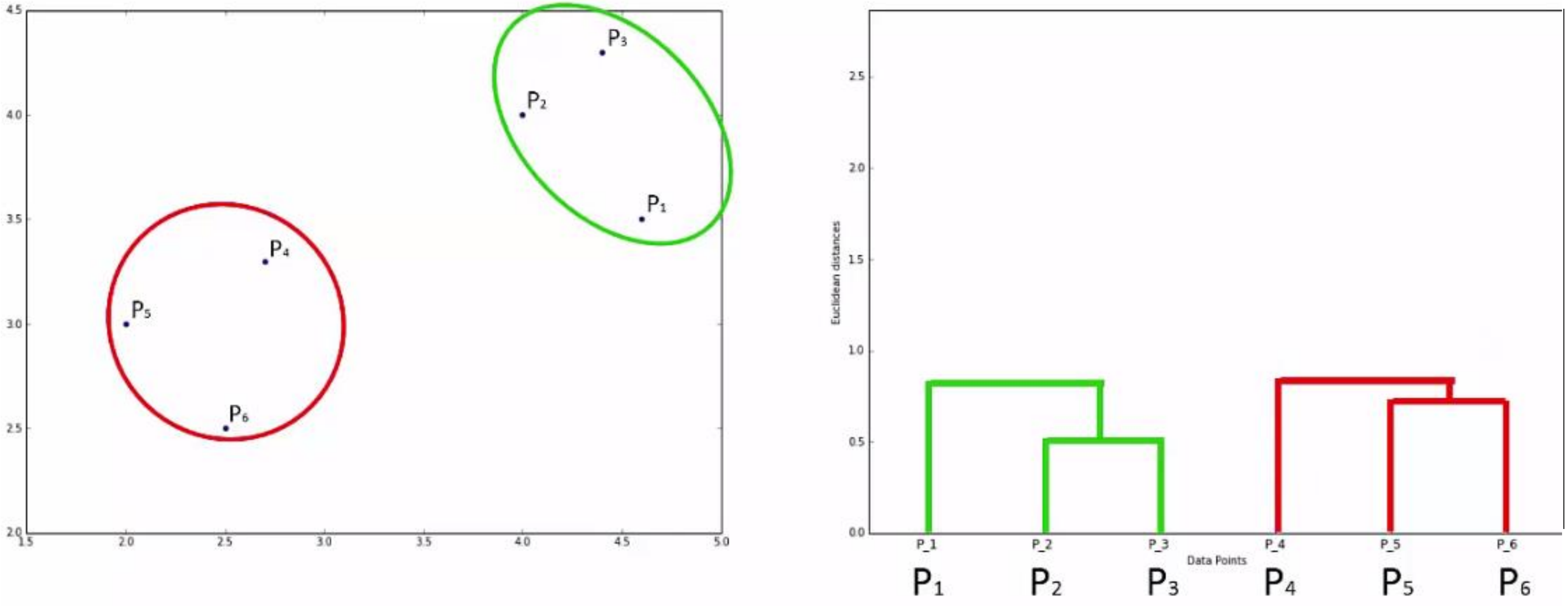


How do Dendrograms Works

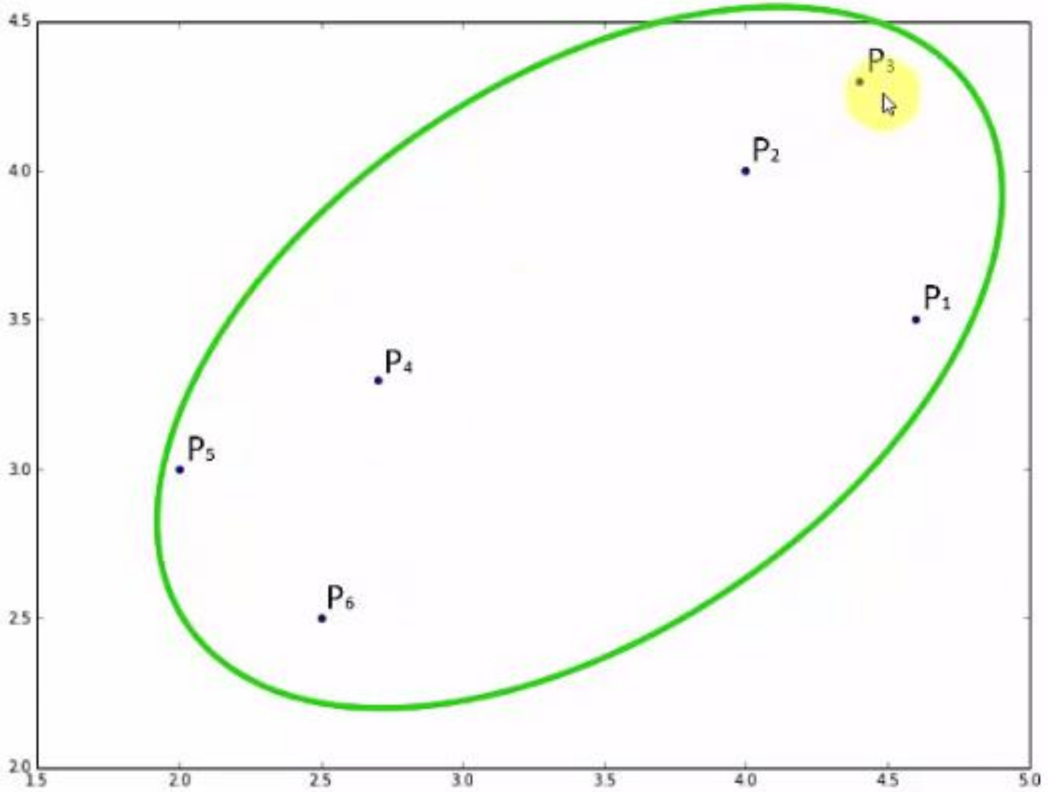


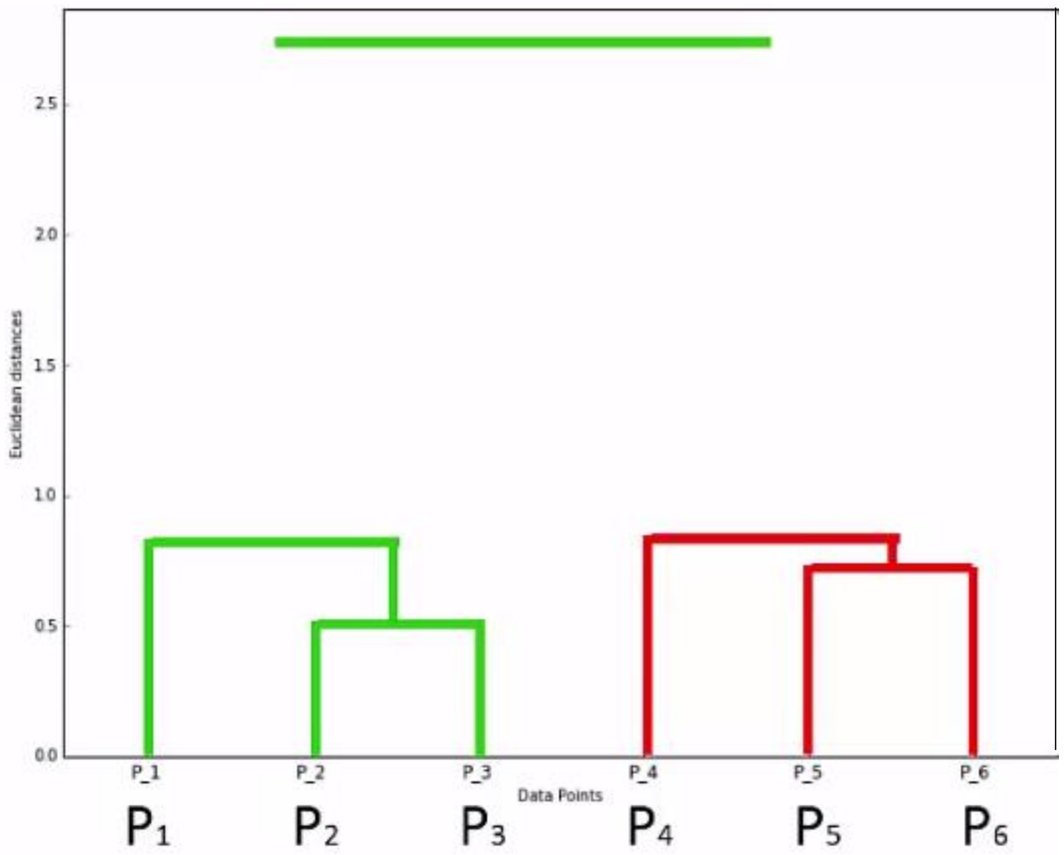


How do Dendrograms Works

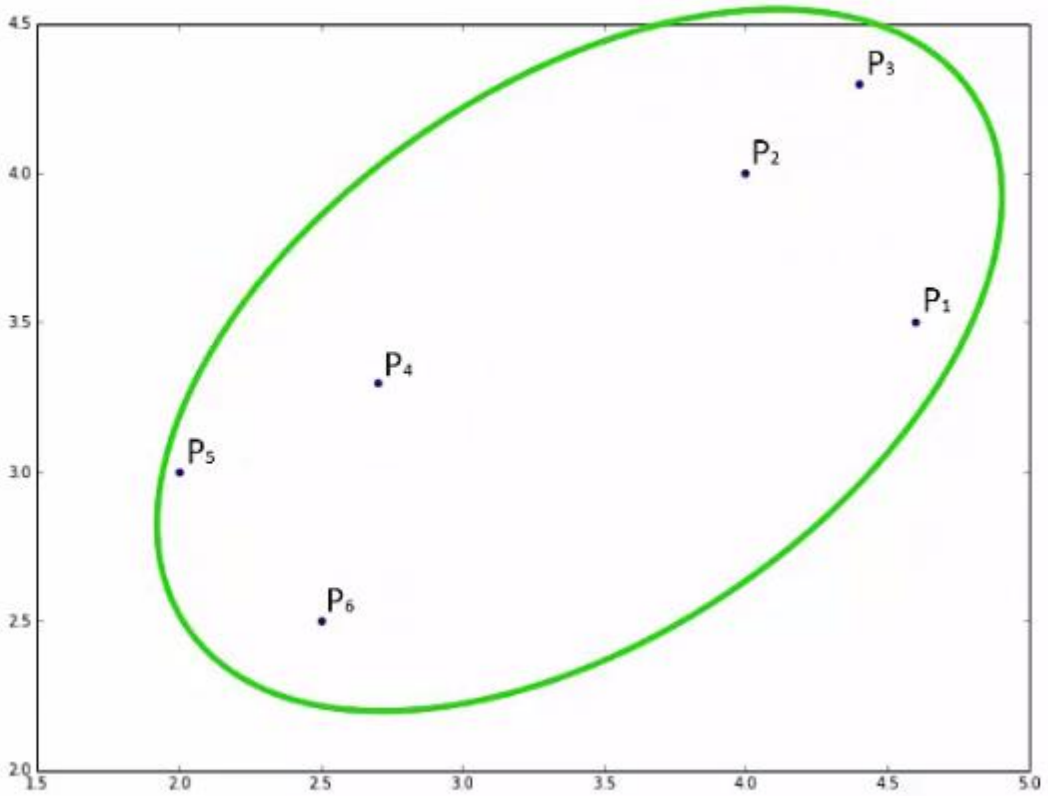


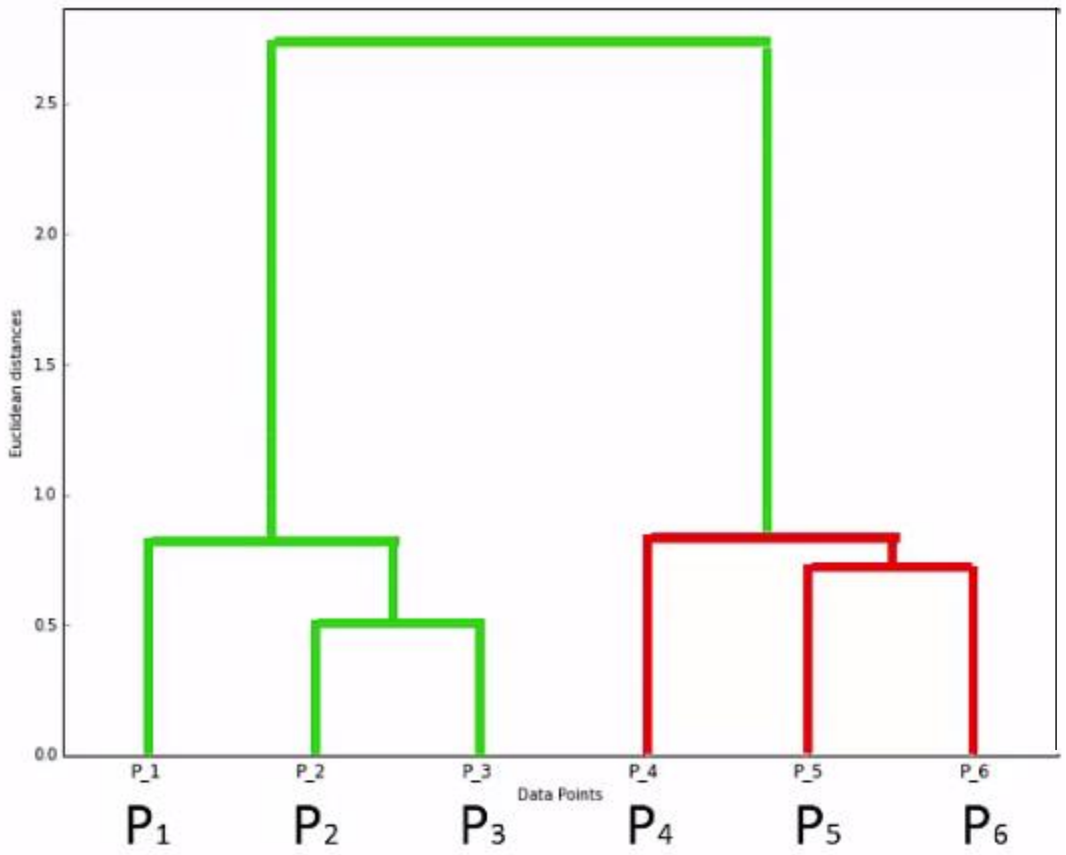
How do Dendrograms Works



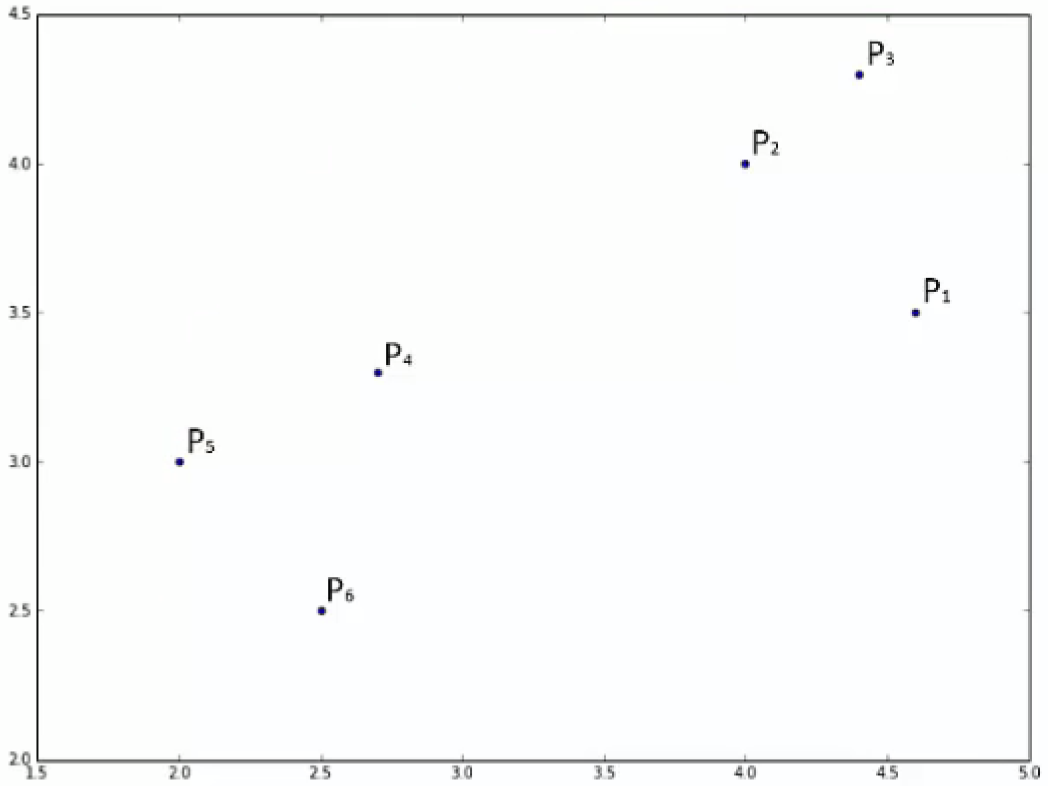


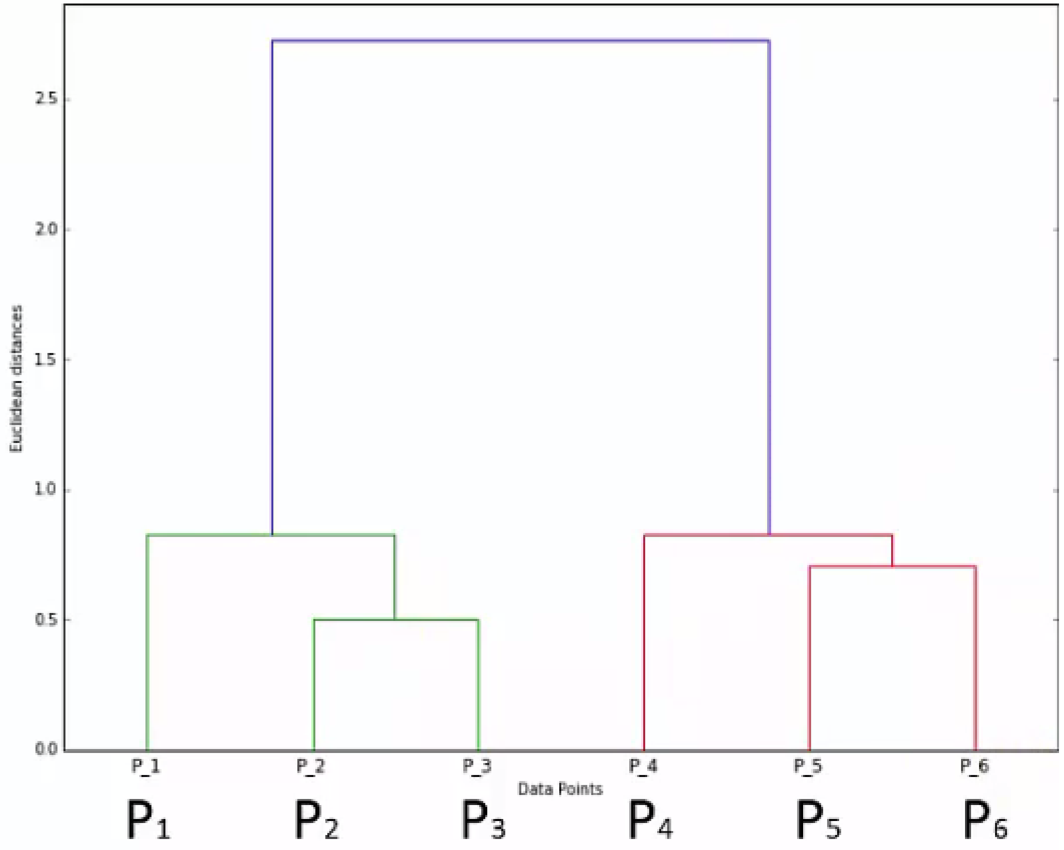
How do Dendrograms Works





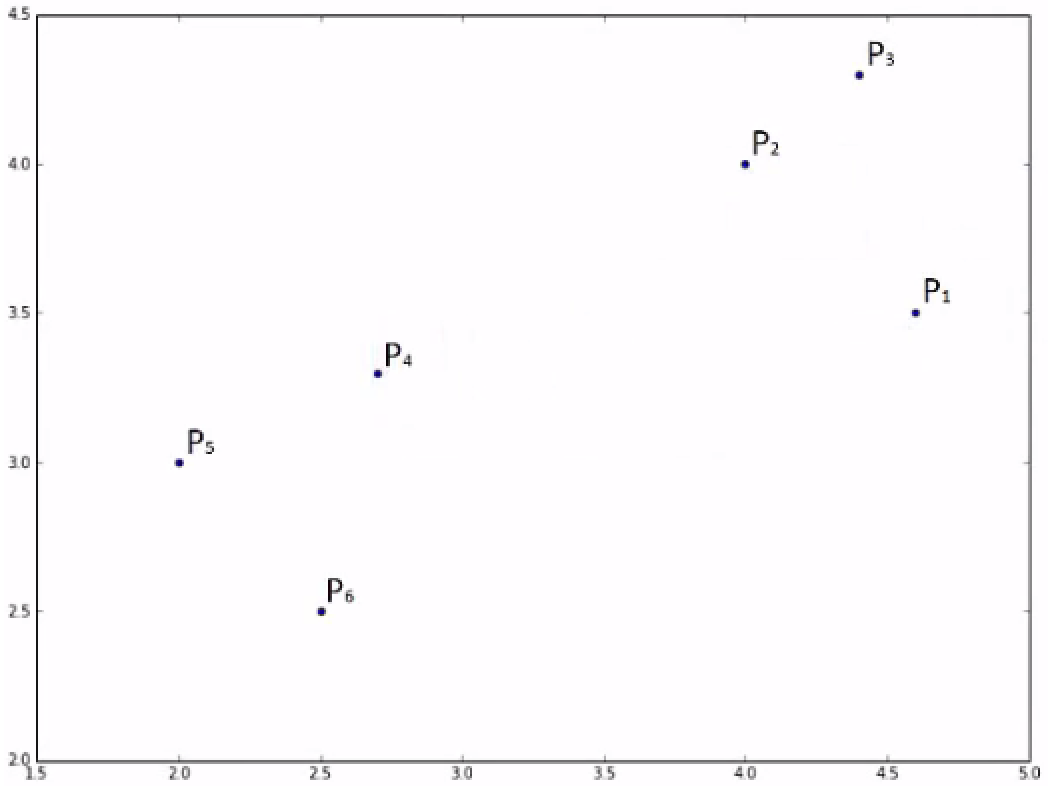
How do Dendrograms Works

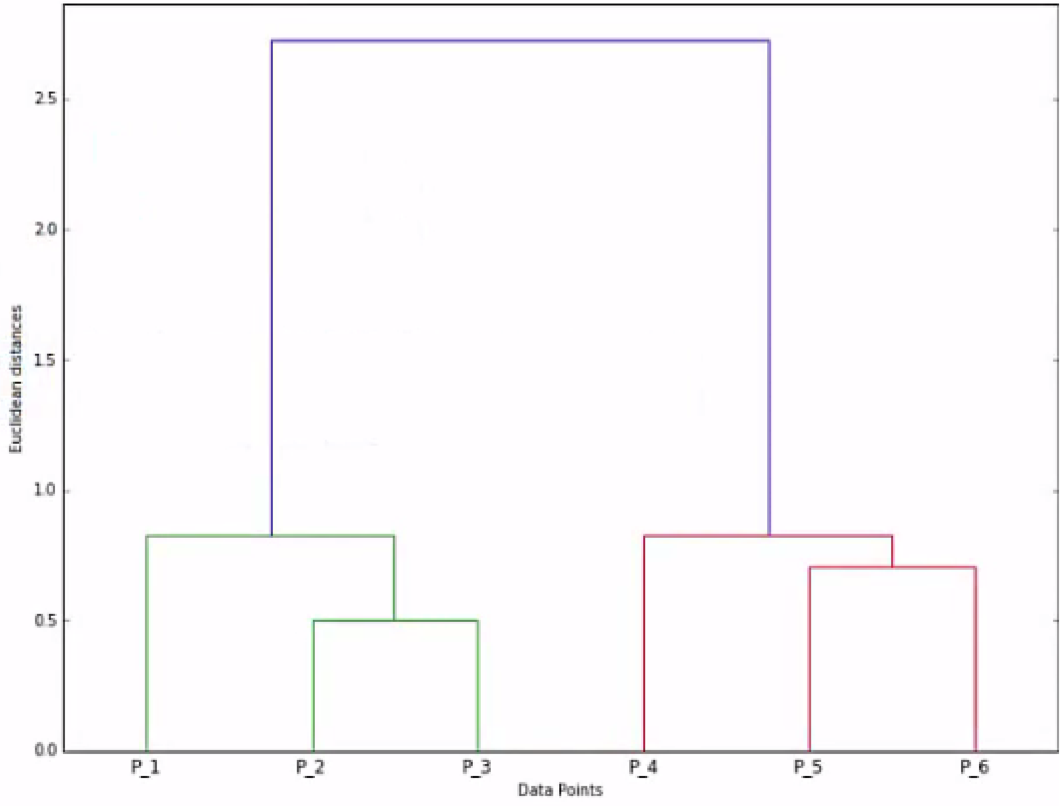




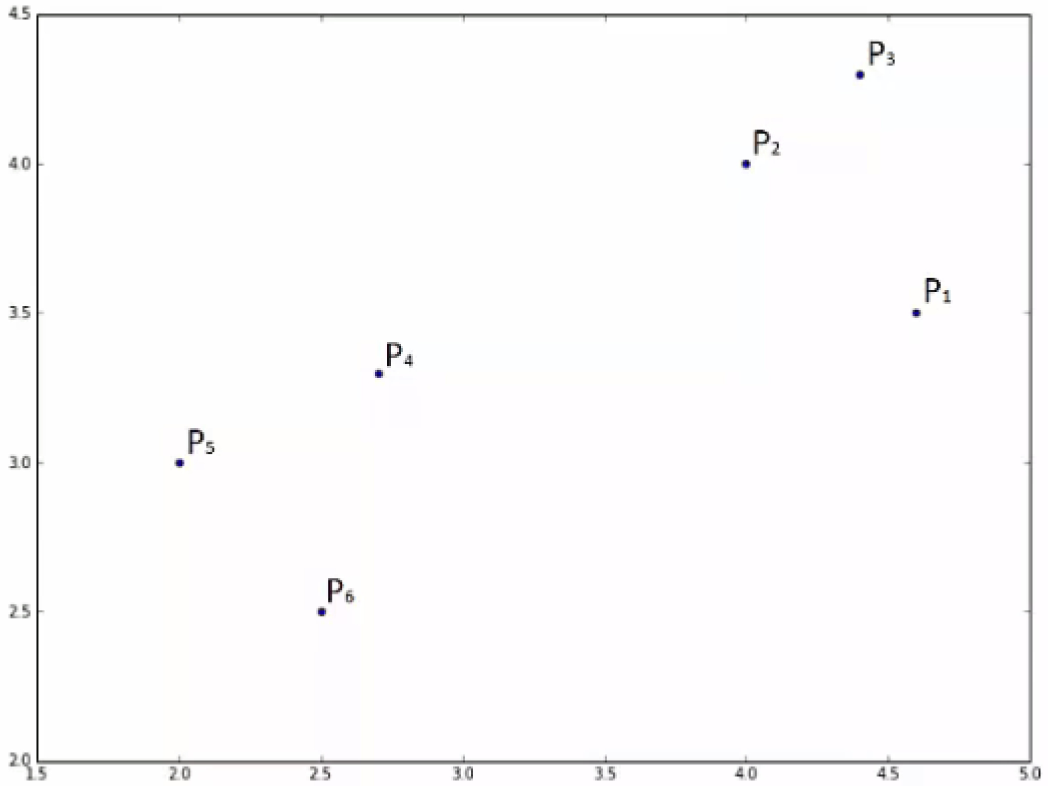
Using Dendrograms

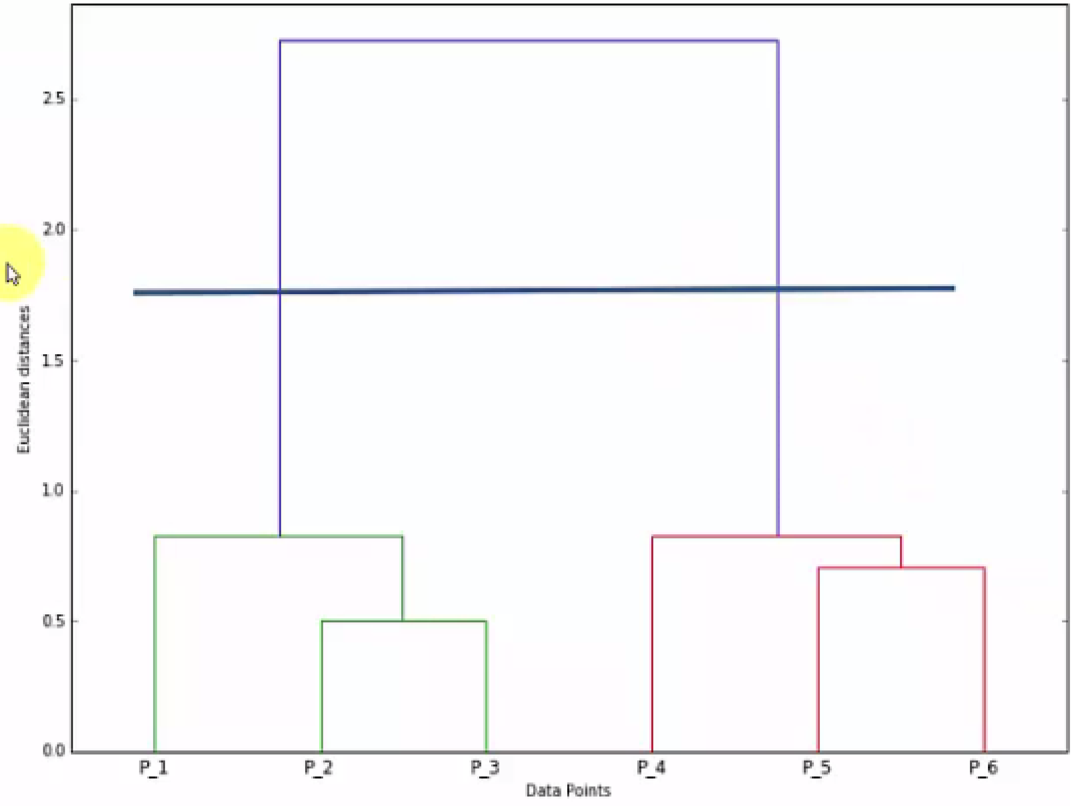
Dendrograms – Two Clusters



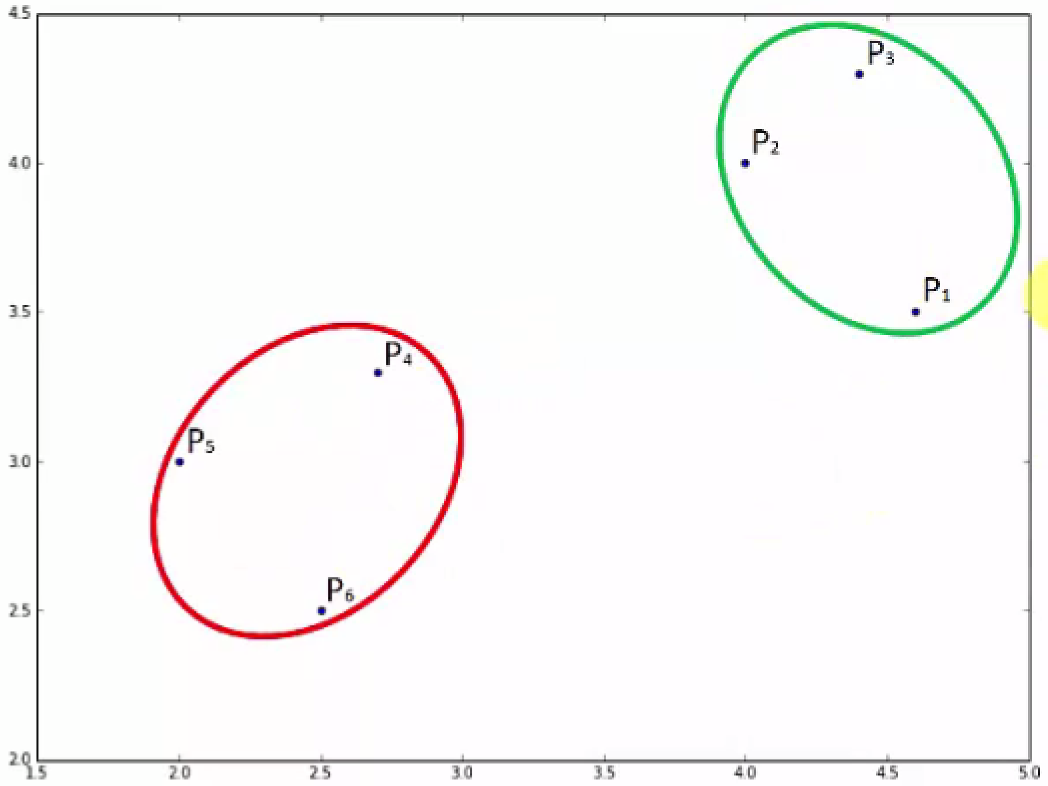


Dendrograms – Two Clusters

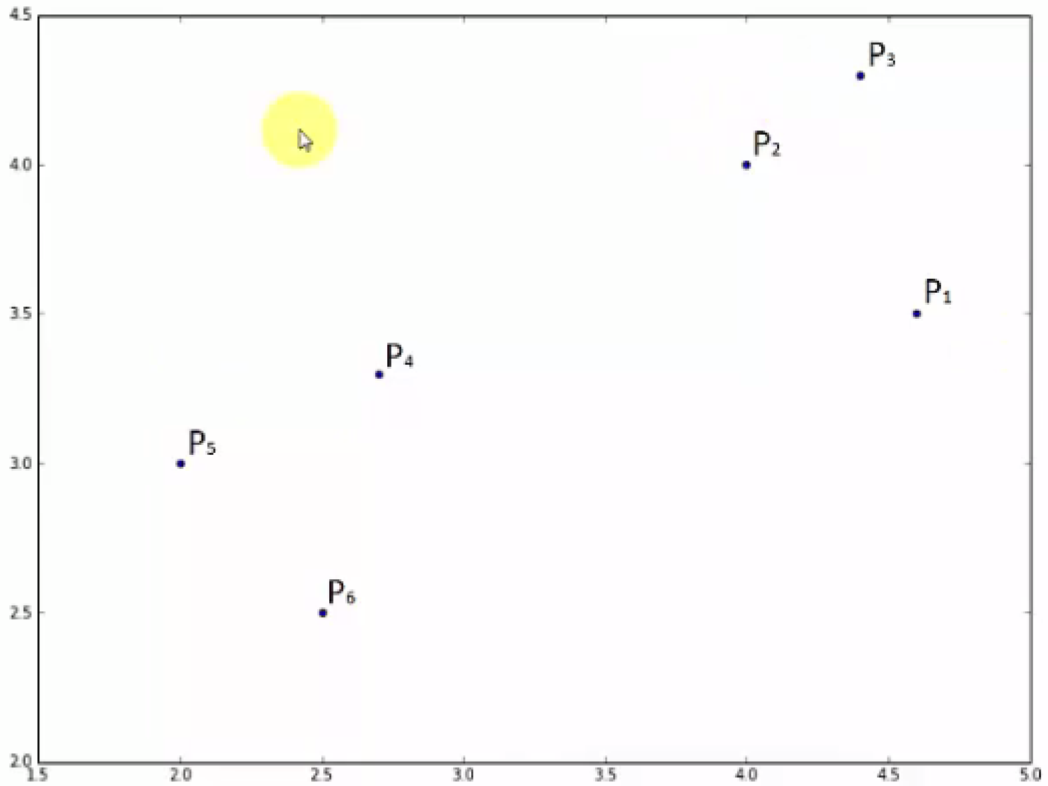


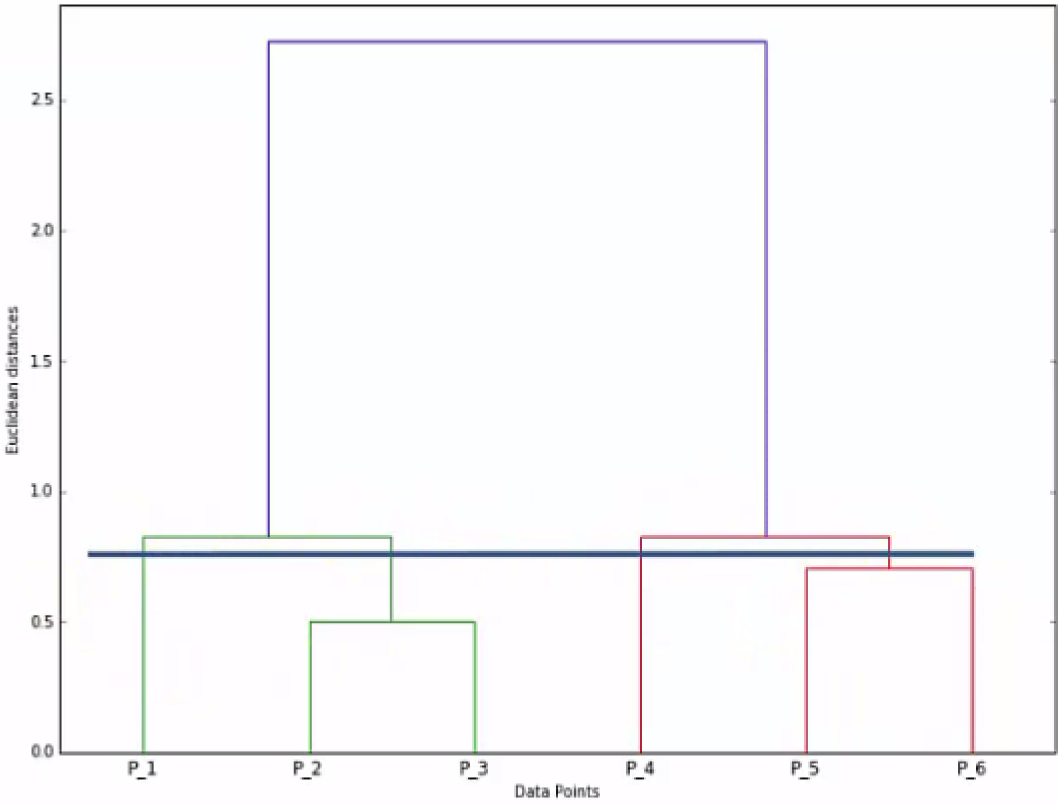


Dendrograms – Two Clusters

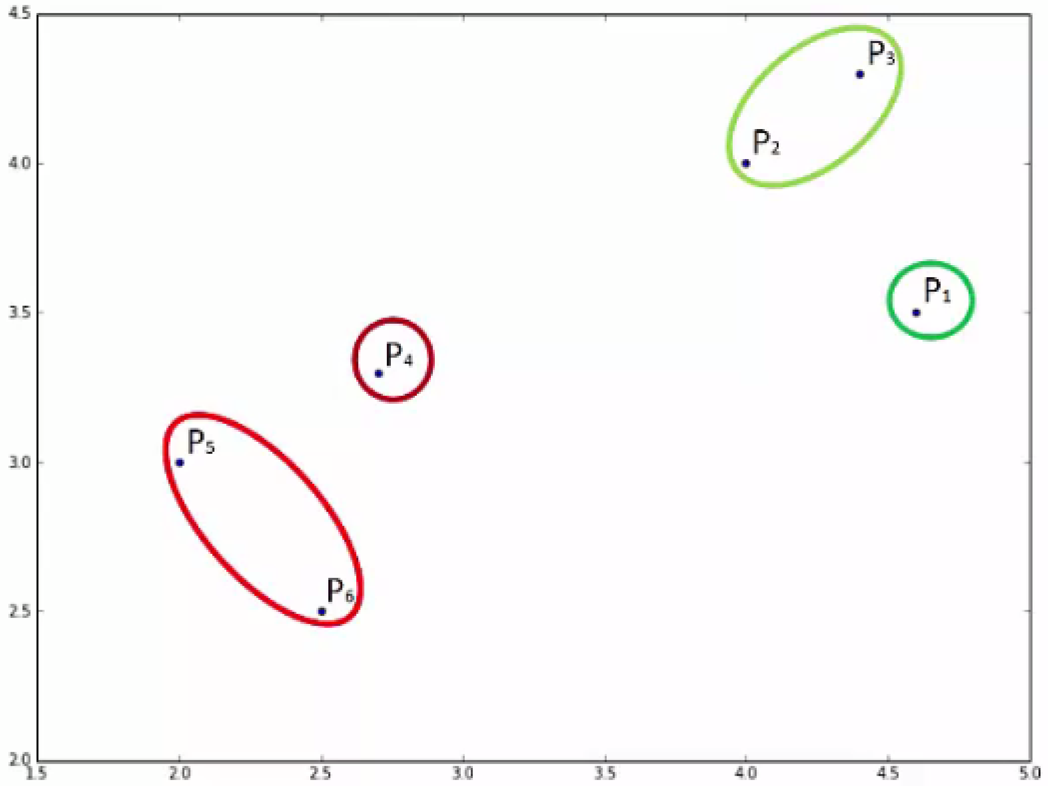


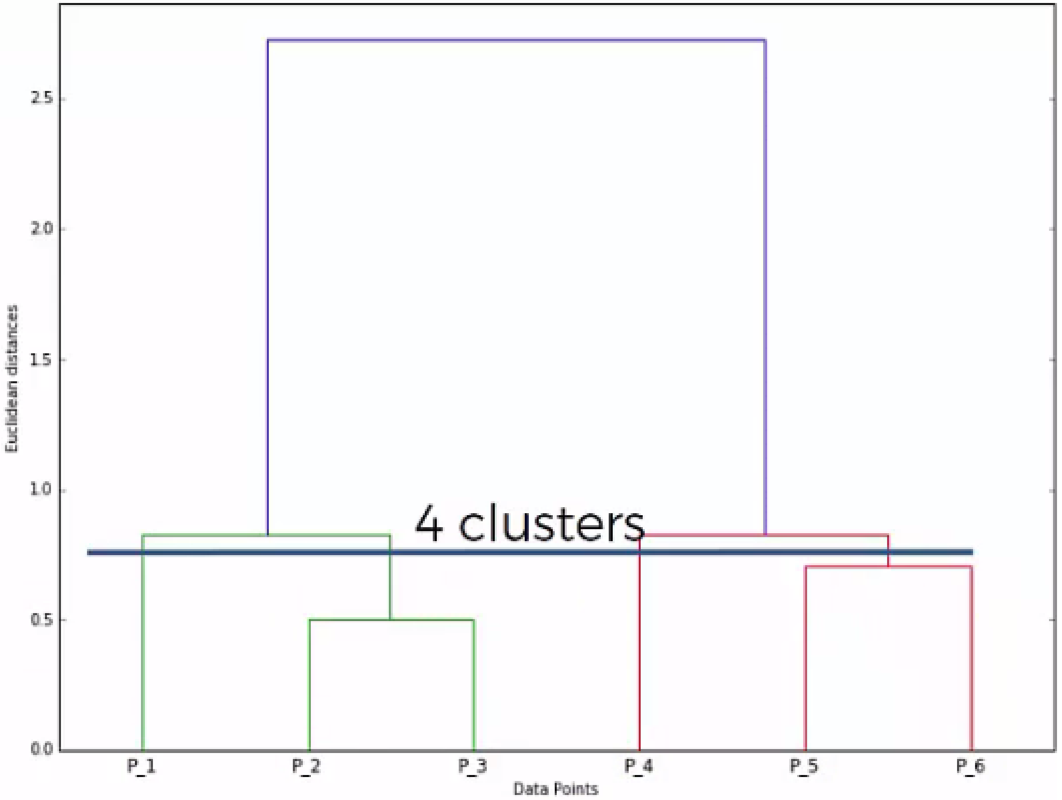
Dendrograms – Four Clusters



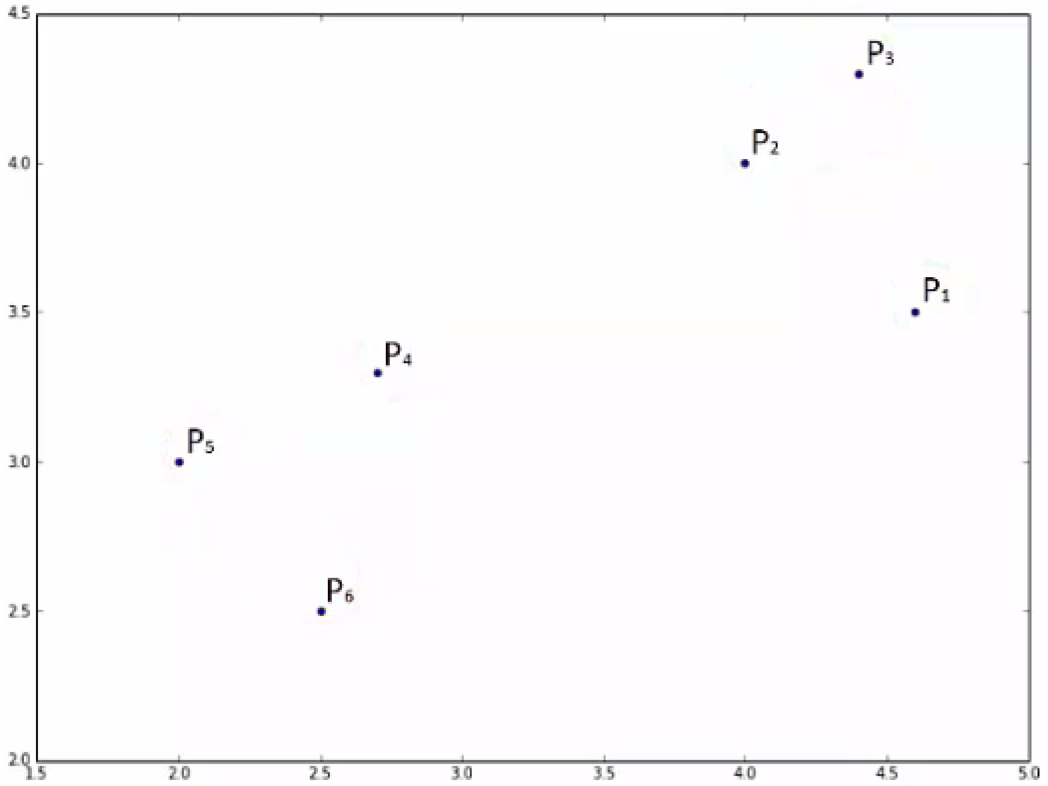


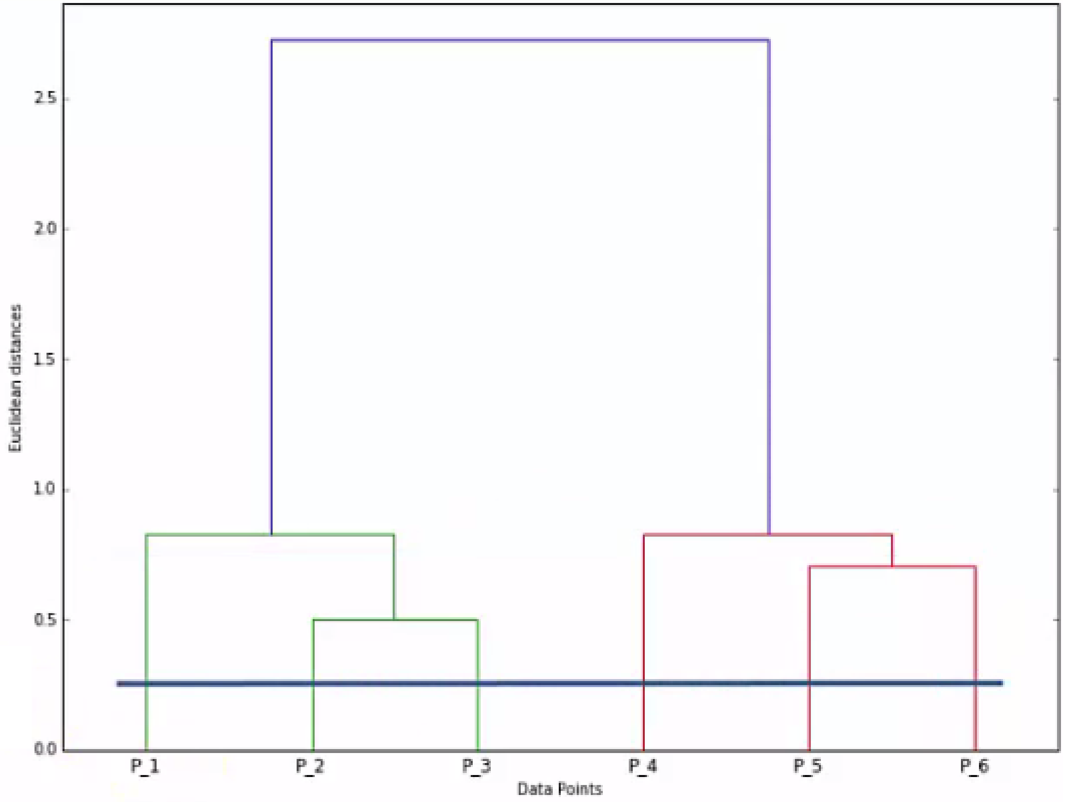
Dendrograms – Four Clusters



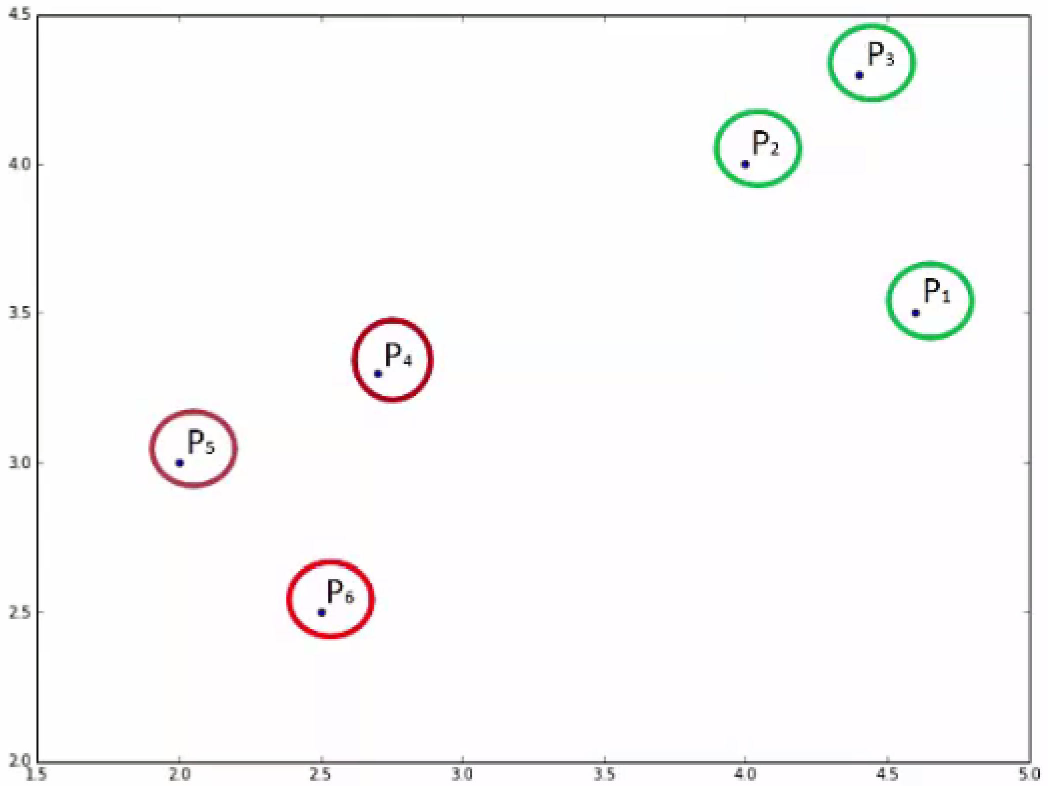


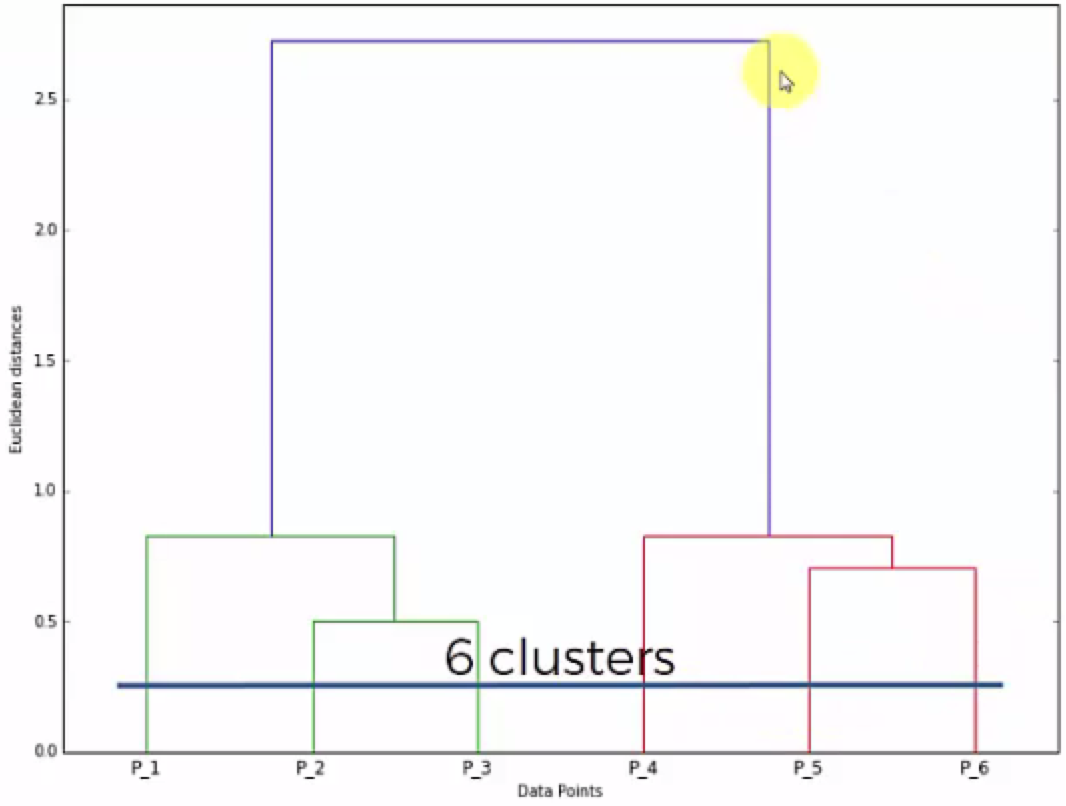
Dendrograms – Six Clusters



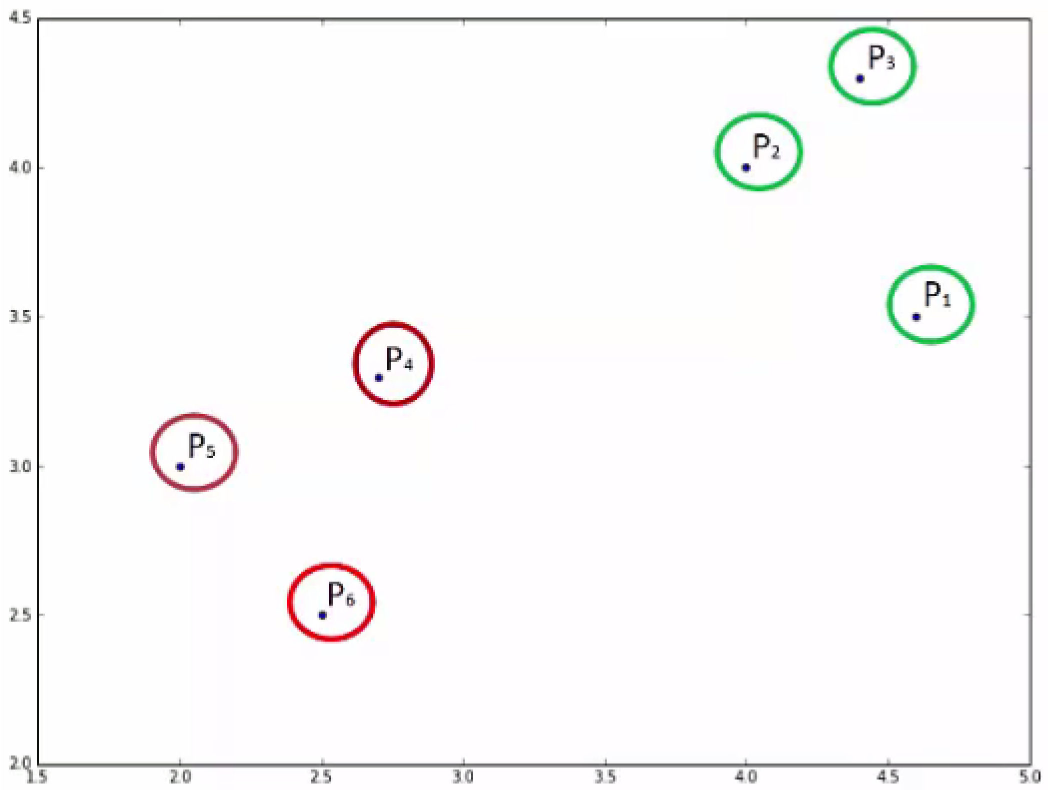


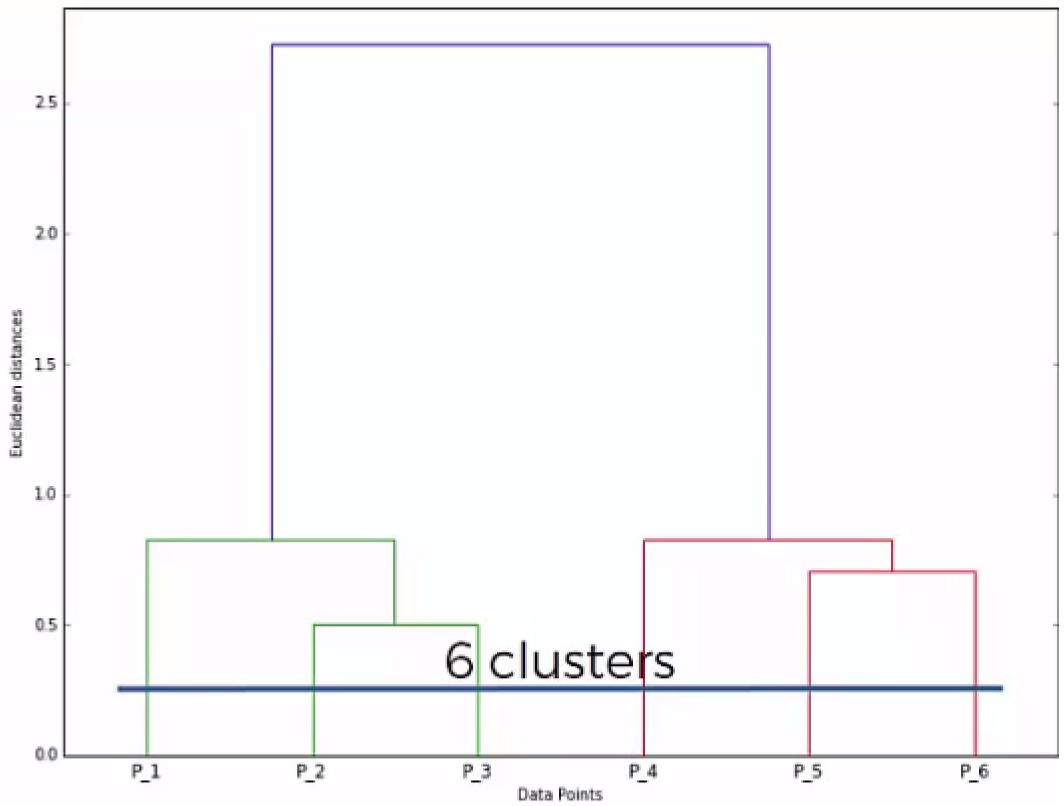
Dendrograms – Six Clusters



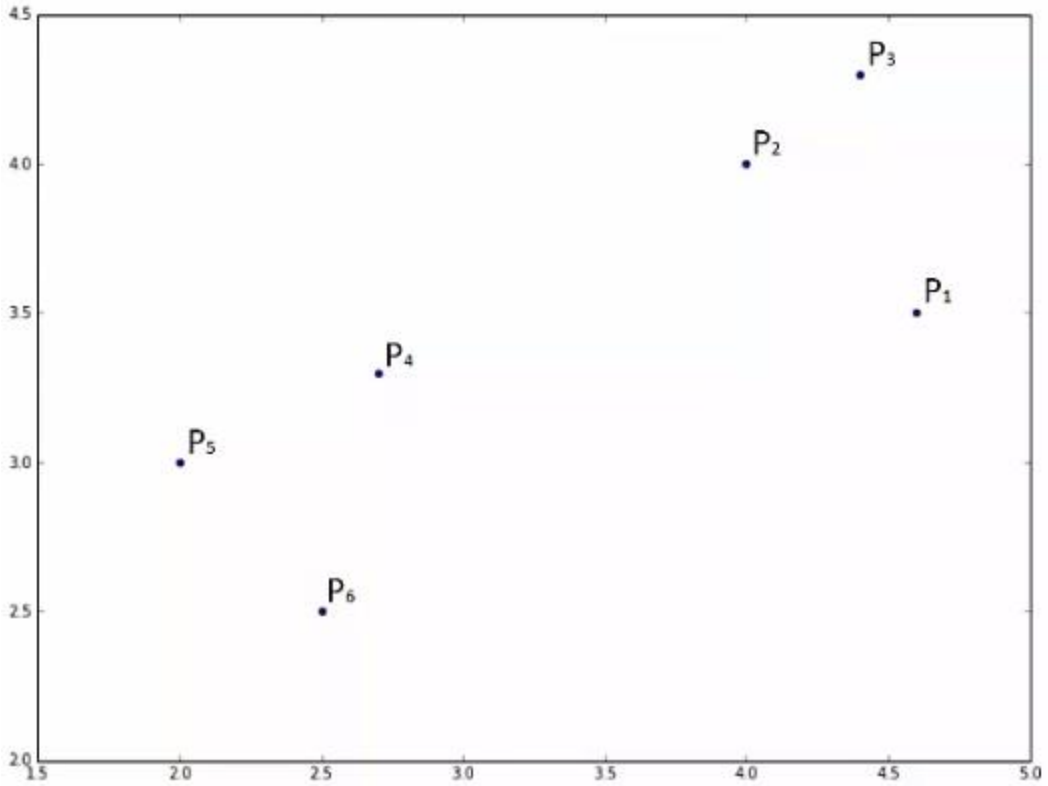


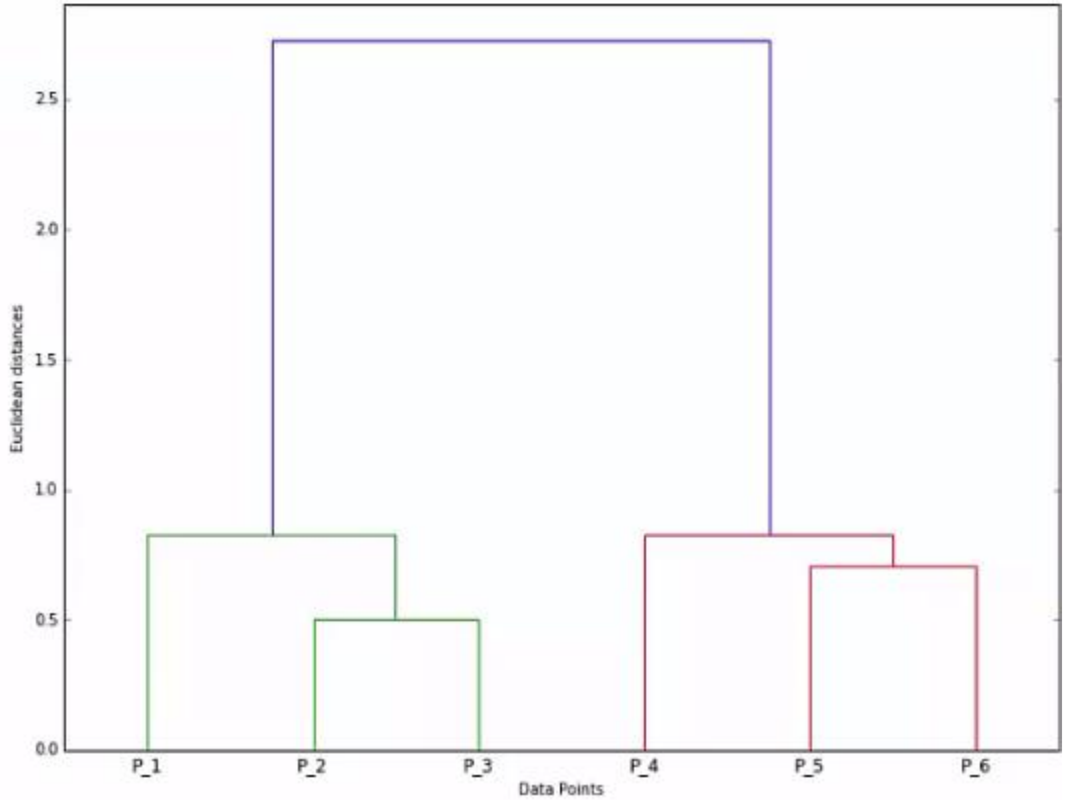
Dendrograms – Six Clusters



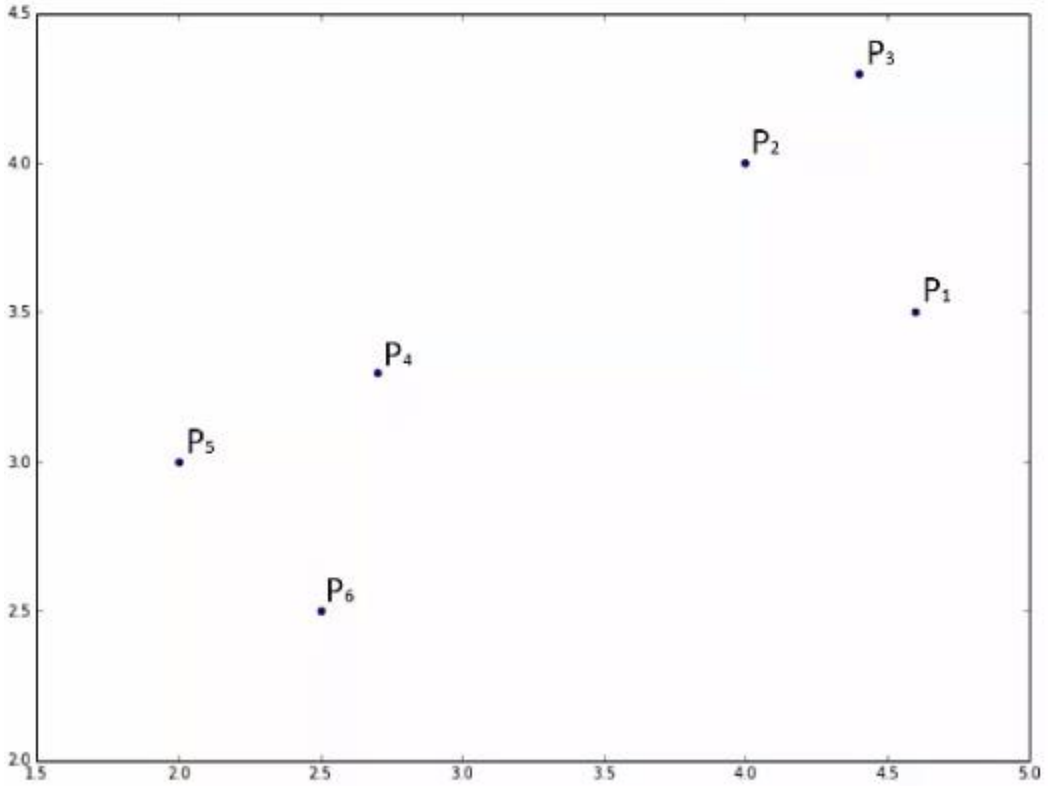


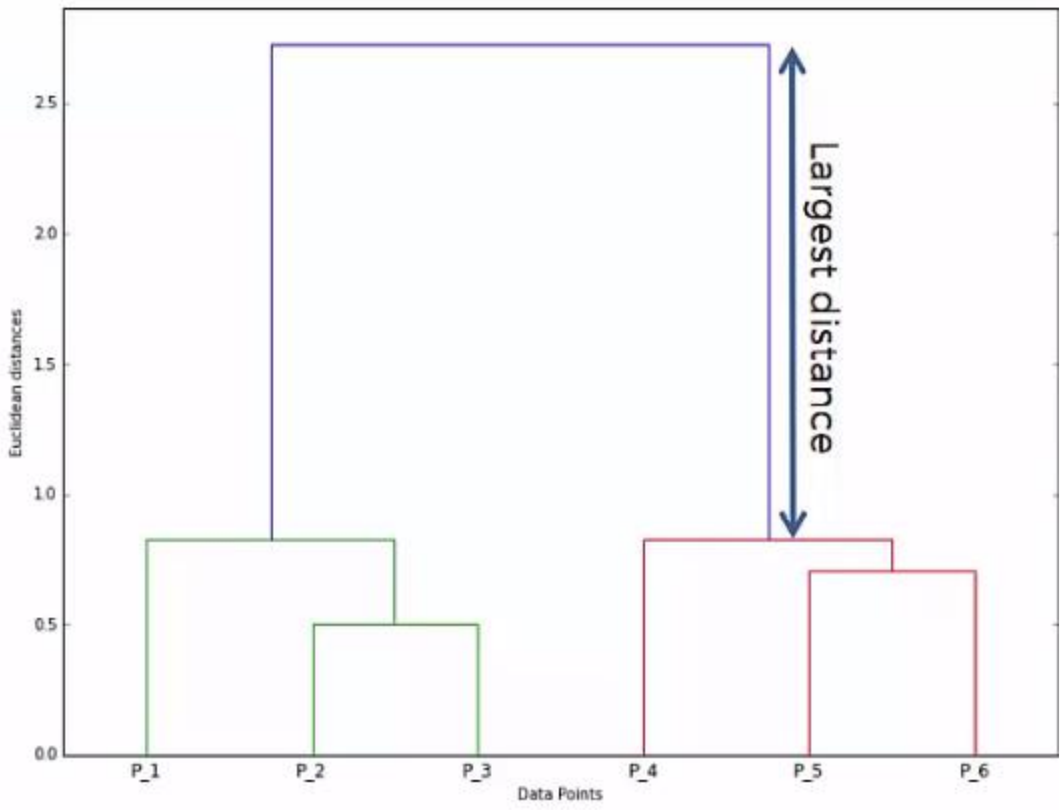
Dendrograms – Optimal # of Clusters



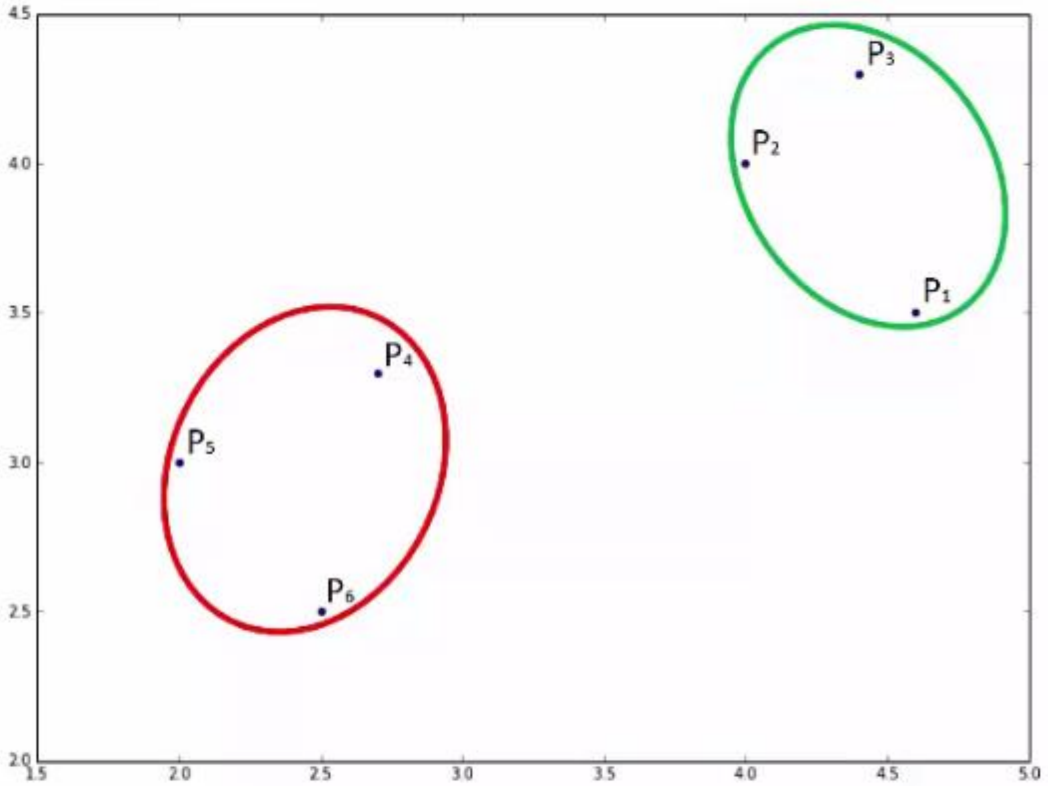


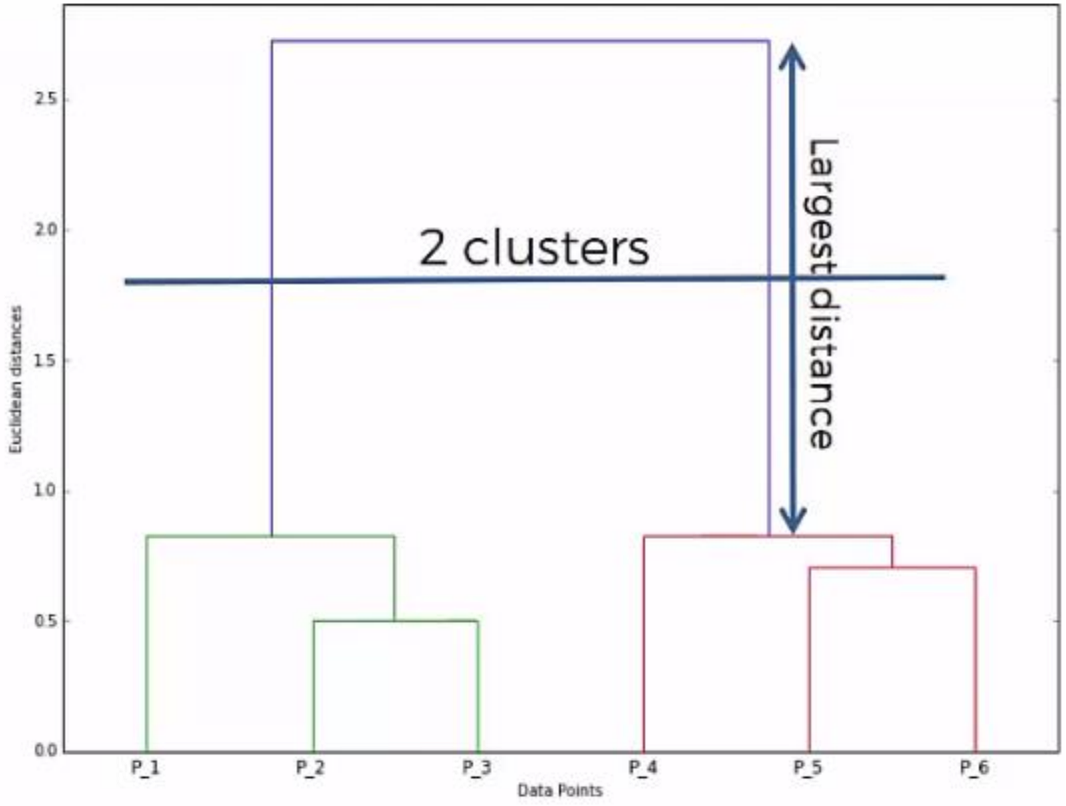
Dendrograms – Optimal # of Clusters





Dendrograms – Optimal # of Clusters





Example: Distance metrics used in hierarchical clustering

- **Single Linkage**, **Complete Linkage**, and **Average Linkage**.

Point	x	y
A	1	2
B	2	3
C	3	1
D	5	4
E	6	5

Step 1: Calculate Pairwise Distances

- Euclidean distance between each pair of points using the formula:

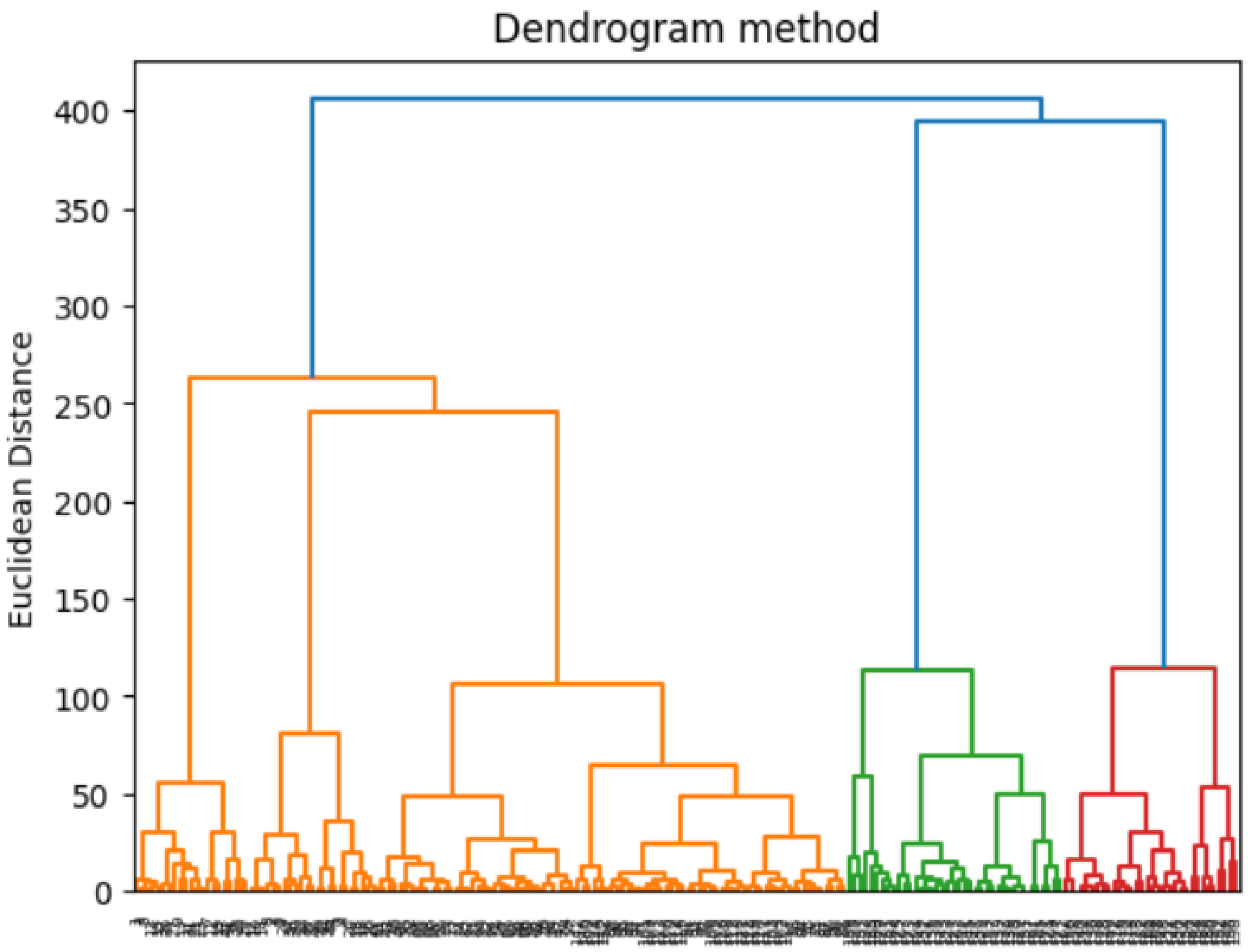
$$d(P_i, P_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

Hierarchical Clustering

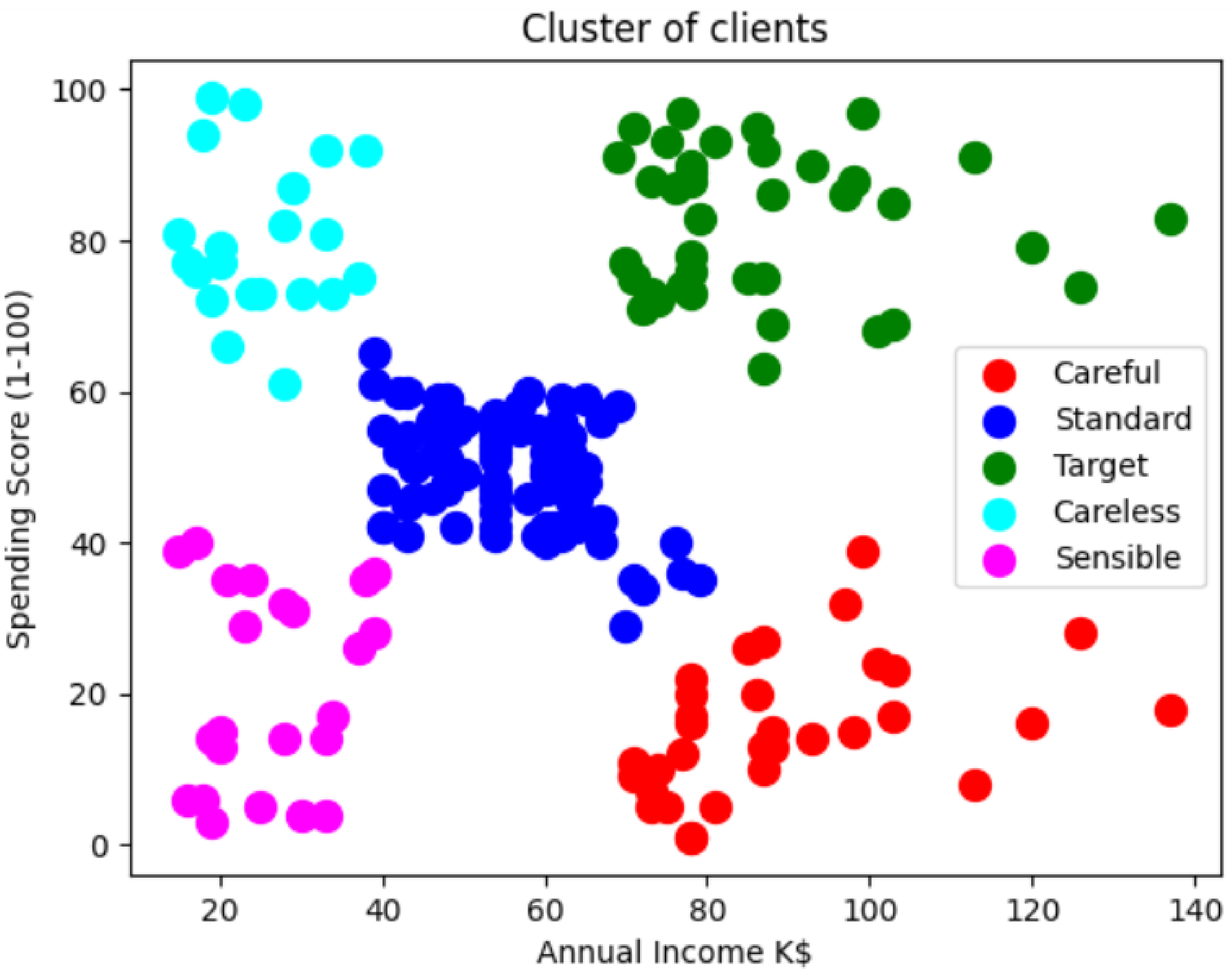
Implementation in Python

```python
#Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from google.colab import drive
drive.mount('/content/drive’)

# Importing the mall dataset with pandas
import pandas as pd

data = pd.read_csv("drive/My Drive/Colab
Notebooks/DataSets/mall.csv")

dataset = data

X = dataset.iloc[:, [3, 4]]. values
X
```

```
#Using dendrogram method to find the optimal
number of clusters

import scipy.cluster.hierarchy as sch

dendrogram = sch.dendrogram(sch.linkage (X,
method ='ward'))

plt.title('Dendrogram method’)
plt.xlabel('Customers’)
plt.ylabel('Euclidean Distance’)
plt.show()
```

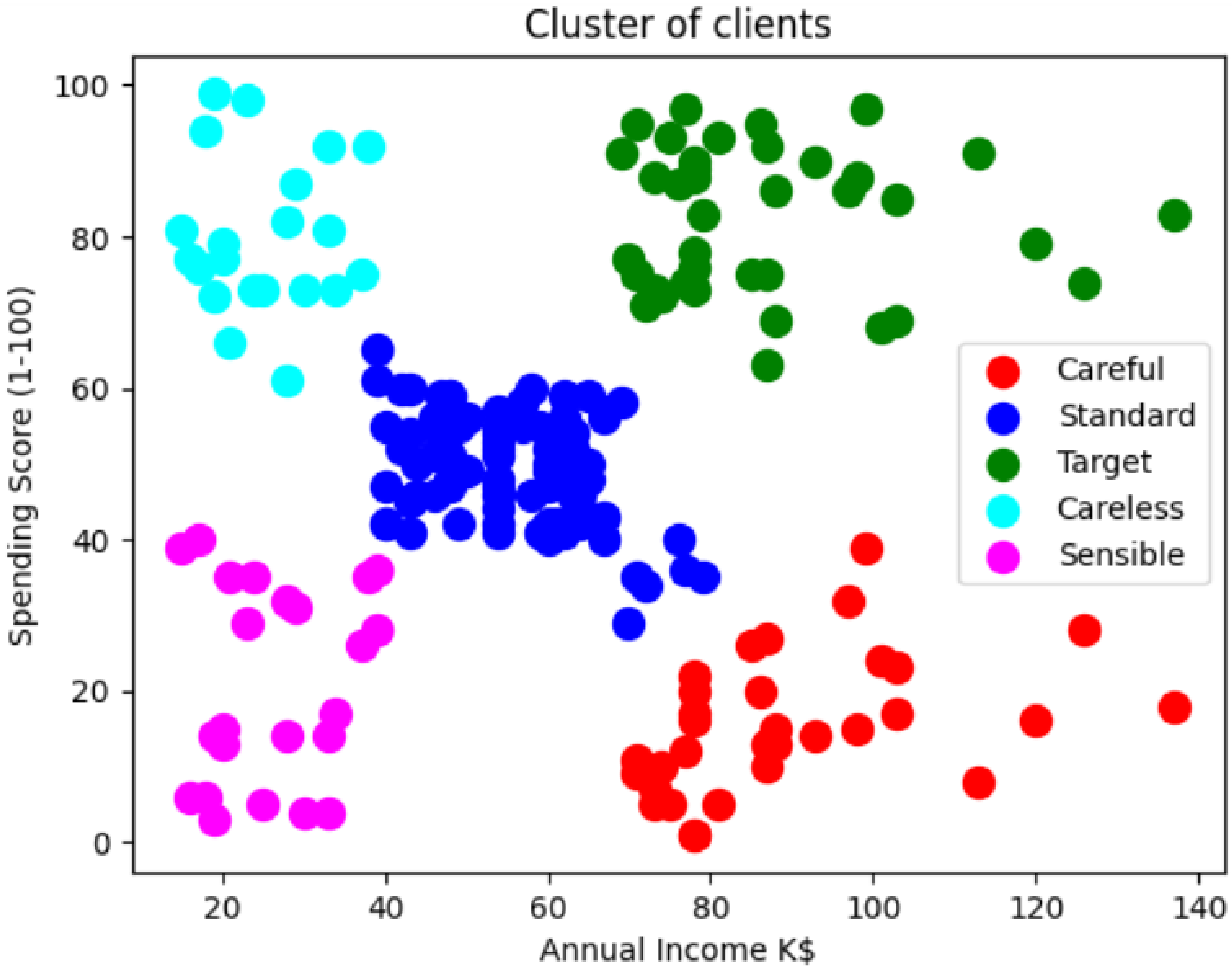

Dendrogram method
Euclidean Distance
400
350
300
250
200
150
100
50
0

```
#fitting HC to the mall dataset
#Agglomerative HC
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters =5, metric
='euclidean', linkage='ward')

#each customer belongs to which cluster
y_hc = hc.fit_predict(X)

print(y_hc)
```


Cluster of clients
Spending Score (1-100)
Annual Income K$
0
20
40
60
80
100
20
40
60
80
100
120
140
Careful
Standard
Target
Careless
Sensible

```
plt.scatter(X[y_hc==0, 0], X[y_hc==0, 1], s=100, c='red’,
label = 'Careful’)
plt.scatter(X[y_hc==1, 0], X[y_hc==1, 1], s=100, c='blue’,
label = 'Standard’)
plt.scatter(X[y_hc==2, 0], X[y_hc==2, 1], s=100,
c='green', label = 'Target’)
plt.scatter(X[y_hc==3, 0], X[y_hc==3, 1], s=100,
c='cyan', label = 'Careless’)
plt.scatter(X[y_hc==4, 0], X[y_hc==4, 1], s=100,
c='magenta', label = 'Sensible')
```

```
plt.title('Cluster of clients’)
plt.xlabel('Annual Income K$’)
plt.ylabel('Spending Score (1-100)’)
plt.legend()
plt.show()
```


Cluster of clients
Spending Score (1-100)
Annual Income K$
0
20
40
60
80
100
20
40
60
80
100
120
140
Careful
Standard
Target
Careless
Sensible

Thank you