ML LAB 08 - SINGLE AND COMPLETE LINK

Name : Tulasi Raman R

Register Number : 21MIS1170

Importing Libraries

In [73]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import scipy.cluster.hierarchy as shc
from scipy.spatial.distance import squareform, pdist
```

Generate Dataset

In [85]:
```
a = np.random.random_sample(size = 10)
b = np.random.random_sample(size = 10)
point = ['P1','P2','P3','P4','P5','P6','P7','P8','P9','P10']
df = pd.DataFrame({'Point':point, 'a':np.round(a,2), 'b':np.round(b,2)})
df = df.set_index('Point')
df
```

Out[85]:

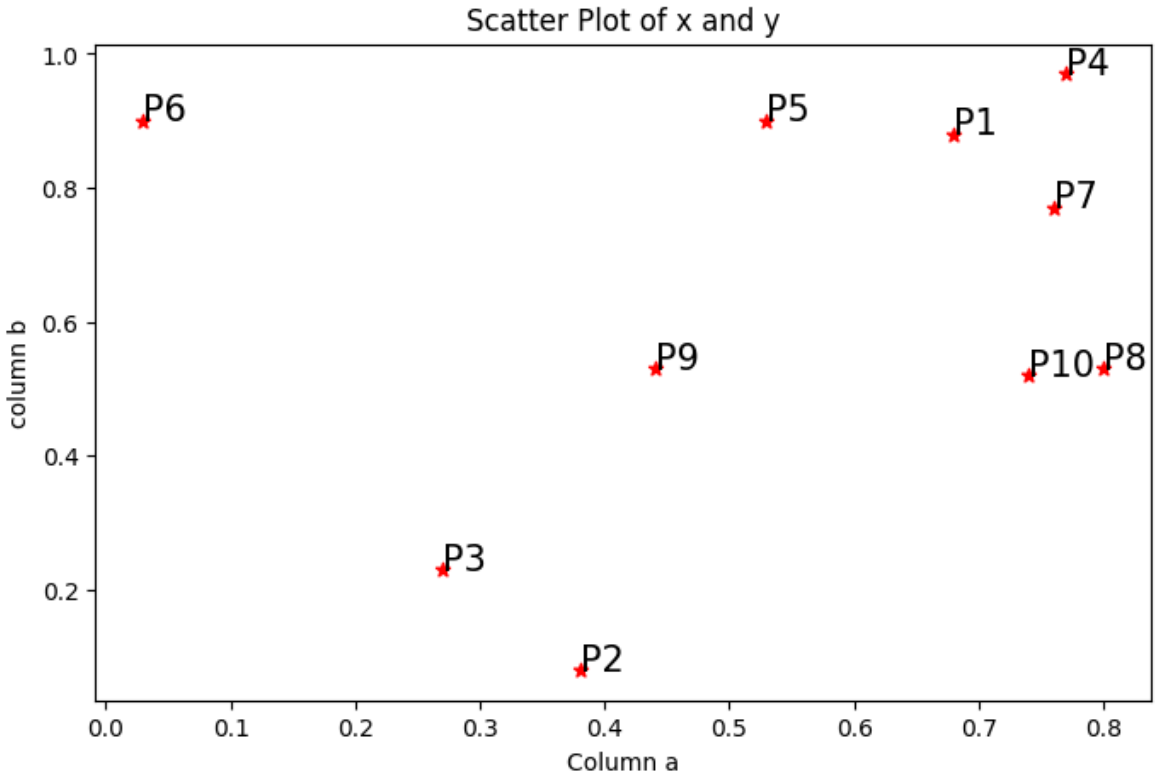
Point	a	b
P1	0.68	0.88
P2	0.38	0.08
P3	0.27	0.23
P4	0.77	0.97
P5	0.53	0.90
P6	0.03	0.90
P7	0.76	0.77
P8	0.80	0.53
P9	0.44	0.53
P10	0.74	0.52

In [98]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, P1 to P10
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   a       10 non-null     float64
 1   b       10 non-null     float64
dtypes: float64(2)
memory usage: 540.0+ bytes
```

Visualizing the Dataset

In [99]:
```
plt.figure(figsize=(8,5))
plt.scatter(df['a'], df['b'], c='r', marker='*')
plt.xlabel('Column a')
plt.ylabel('column b')
plt.title('Scatter Plot of x and y')
for j in df.itertuples():
    plt.annotate(j.Index, (j.a, j.b), fontsize=15)
```



Calculating Euclidean Distance Between Each Point

In [100…
```
dist = pd.DataFrame(squareform(pdist(df[['a','b']]), 'euclidean'), columns = df.
dist
```

Out[100…

	P1	P2	P3	P4	P5	P6	P7	P8	
P1	0.000000	0.854400	0.768505	0.127279	0.151327	0.650308	0.136015	0.370000	0.4
P2	0.854400	0.000000	0.186011	0.971700	0.833607	0.891572	0.787718	0.615549	0.4
P3	0.768505	0.186011	0.000000	0.893085	0.718679	0.711688	0.729178	0.609016	0.3
P4	0.127279	0.971700	0.893085	0.000000	0.250000	0.743303	0.200250	0.441022	0.5
P5	0.151327	0.833607	0.718679	0.250000	0.000000	0.500000	0.264197	0.458039	0.3
P6	0.650308	0.891572	0.711688	0.743303	0.500000	0.000000	0.741485	0.854283	0.5
P7	0.136015	0.787718	0.729178	0.200250	0.264197	0.741485	0.000000	0.243311	0.4
P8	0.370000	0.615549	0.609016	0.441022	0.458039	0.854283	0.243311	0.000000	0.3
P9	0.424382	0.453982	0.344819	0.550000	0.380789	0.552268	0.400000	0.360000	0.0
P10	0.364966	0.568507	0.552268	0.450999	0.434166	0.805295	0.250799	0.060828	0.3

Single Linkage Clustering

Interpretation from Euclidean Distance for Single Link

- Notice that Point(P8,P10) is having the the least distance of all with 0.060828, so we'll conclude these both clusters as neighbouring clusters. So initially we'll merge those two points together considering the least distance between the cluster pointer which is again 0.060828. Then the next least distance is (P1, P4) with 0.127279, so we'll merge those two neighbouring clusters by calculating the minimum distance between cluster points which is again 0.127279 as we have only one point in each cluster and minimum distance have to be computed.
- Similarly calculating the least distance between each and every clusters and merging them based on the minimum distance between the neighbouring clusters is called Single Linkage Clustering. This iterative process continues until all data points are clustered, revealing patterns in the data.

In [106…

```
def single_linkage(dist_matrix):
    n = len(dist_matrix)
    while n > 1:
        min_val = float('inf')
        min_index = None
        for i in range(n):
            for j in range(i+1, n):
                if dist_matrix.iloc[i, j] < min_val and dist_matrix.index[i] !=
                    min_val = dist_matrix.iloc[i, j]
                    min_index = (i, j)

        if min_val == float('inf'):
            break
```

```
        i, j = min_index
        cluster1, cluster2 = dist_matrix.index[i], dist_matrix.columns[j]
        print(f'Merging clusters {cluster1} and {cluster2} with distance {min_va

        new_cluster = f'({cluster1},{cluster2})'

        dist_matrix[new_cluster] = dist_matrix[[cluster1, cluster2]].min(axis=1)
        dist_matrix.loc[new_cluster] = dist_matrix.loc[[cluster1, cluster2]].min
        dist_matrix = dist_matrix.drop([cluster1, cluster2], axis=0)
        dist_matrix = dist_matrix.drop([cluster1, cluster2], axis=1)
        n -= 1

        print(dist_matrix)
        print("")
print("Single-Link Clustering:")
single_linkage(dist.copy())
```

```
Single-Link Clustering:
Merging clusters P8 and P10 with distance 0.060827625302982254
                P1        P2        P3        P4        P5        P6  \
P1        0.000000  0.854400  0.768505  0.127279  0.151327  0.650308
P2        0.854400  0.000000  0.186011  0.971700  0.833607  0.891572
P3        0.768505  0.186011  0.000000  0.893085  0.718679  0.711688
P4        0.127279  0.971700  0.893085  0.000000  0.250000  0.743303
P5        0.151327  0.833607  0.718679  0.250000  0.000000  0.500000
P6        0.650308  0.891572  0.711688  0.743303  0.500000  0.000000
P7        0.136015  0.787718  0.729178  0.200250  0.264197  0.741485
P9        0.424382  0.453982  0.344819  0.550000  0.380789  0.552268
(P8,P10)  0.364966  0.568507  0.552268  0.441022  0.434166  0.805295

                P7        P9  (P8,P10)
P1        0.136015  0.424382  0.364966
P2        0.787718  0.453982  0.568507
P3        0.729178  0.344819  0.552268
P4        0.200250  0.550000  0.441022
P5        0.264197  0.380789  0.434166
P6        0.741485  0.552268  0.805295
P7        0.000000  0.400000  0.243311
P9        0.400000  0.000000  0.300167
(P8,P10)  0.243311  0.300167  0.000000

Merging clusters P1 and P4 with distance 0.12727922061357852
                P2        P3        P5        P6        P7        P9  \
P2        0.000000  0.186011  0.833607  0.891572  0.787718  0.453982
P3        0.186011  0.000000  0.718679  0.711688  0.729178  0.344819
P5        0.833607  0.718679  0.000000  0.500000  0.264197  0.380789
P6        0.891572  0.711688  0.500000  0.000000  0.741485  0.552268
P7        0.787718  0.729178  0.264197  0.741485  0.000000  0.400000
P9        0.453982  0.344819  0.380789  0.552268  0.400000  0.000000
(P8,P10)  0.568507  0.552268  0.434166  0.805295  0.243311  0.300167
(P1,P4)   0.854400  0.768505  0.151327  0.650308  0.136015  0.424382

          (P8,P10)   (P1,P4)
P2        0.568507  0.854400
P3        0.552268  0.768505
P5        0.434166  0.151327
P6        0.805295  0.650308
P7        0.243311  0.136015
P9        0.300167  0.424382
(P8,P10)  0.000000  0.364966
(P1,P4)   0.364966  0.000000

Merging clusters P7 and (P1,P4) with distance 0.13601470508735442
                    P2        P3        P5        P6        P9  (P8,P10)  \
P2            0.000000  0.186011  0.833607  0.891572  0.453982  0.568507
P3            0.186011  0.000000  0.718679  0.711688  0.344819  0.552268
P5            0.833607  0.718679  0.000000  0.500000  0.380789  0.434166
P6            0.891572  0.711688  0.500000  0.000000  0.552268  0.805295
P9            0.453982  0.344819  0.380789  0.552268  0.000000  0.300167
(P8,P10)      0.568507  0.552268  0.434166  0.805295  0.300167  0.000000
(P7,(P1,P4))  0.787718  0.729178  0.151327  0.650308  0.400000  0.243311

              (P7,(P1,P4))
P2                0.787718
P3                0.729178
P5                0.151327
P6                0.650308
```

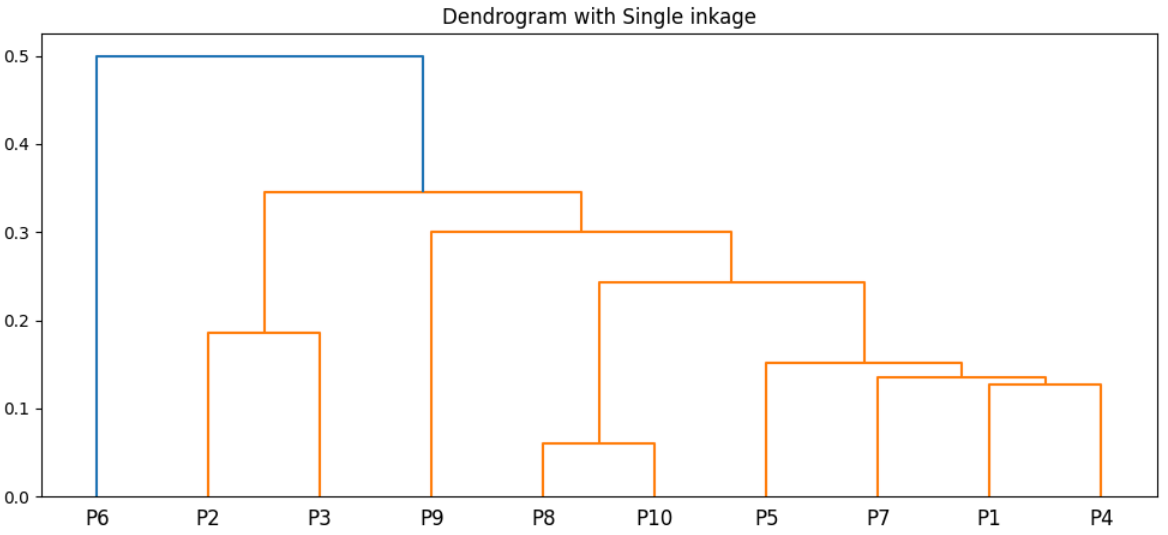
```
P9                0.400000
(P8,P10)          0.243311
(P7,(P1,P4))      0.000000

Merging clusters P5 and (P7,(P1,P4)) with distance 0.15132745950421558
                          P2        P3        P6        P9  (P8,P10)  \
P2                  0.000000  0.186011  0.891572  0.453982  0.568507
P3                  0.186011  0.000000  0.711688  0.344819  0.552268
P6                  0.891572  0.711688  0.000000  0.552268  0.805295
P9                  0.453982  0.344819  0.552268  0.000000  0.300167
(P8,P10)            0.568507  0.552268  0.805295  0.300167  0.000000
(P5,(P7,(P1,P4)))   0.787718  0.718679  0.500000  0.380789  0.243311

                   (P5,(P7,(P1,P4)))
P2                          0.787718
P3                          0.718679
P6                          0.500000
P9                          0.380789
(P8,P10)                    0.243311
(P5,(P7,(P1,P4)))           0.000000

Merging clusters P2 and P3 with distance 0.18601075237738277
                          P6        P9  (P8,P10)  (P5,(P7,(P1,P4)))   (P2,P3)
P6                  0.000000  0.552268  0.805295           0.500000  0.711688
P9                  0.552268  0.000000  0.300167           0.380789  0.344819
(P8,P10)            0.805295  0.300167  0.000000           0.243311  0.552268
(P5,(P7,(P1,P4)))   0.500000  0.380789  0.243311           0.000000  0.718679
(P2,P3)             0.711688  0.344819  0.552268           0.718679  0.000000

Merging clusters (P8,P10) and (P5,(P7,(P1,P4))) with distance 0.2433105012119288
                                    P6        P9   (P2,P3)  \
P6                            0.000000  0.552268  0.711688
P9                            0.552268  0.000000  0.344819
(P2,P3)                       0.711688  0.344819  0.000000
((P8,P10),(P5,(P7,(P1,P4))))  0.500000  0.300167  0.552268

                              ((P8,P10),(P5,(P7,(P1,P4))))
P6                                                0.500000
P9                                                0.300167
(P2,P3)                                           0.552268
((P8,P10),(P5,(P7,(P1,P4))))                      0.000000

Merging clusters P9 and ((P8,P10),(P5,(P7,(P1,P4)))) with distance 0.300166620396
07267
                                         P6   (P2,P3)  \
P6                                 0.000000  0.711688
(P2,P3)                            0.711688  0.000000
(P9,((P8,P10),(P5,(P7,(P1,P4)))))  0.500000  0.344819

                                   (P9,((P8,P10),(P5,(P7,(P1,P4)))))
P6                                                          0.500000
(P2,P3)                                                     0.344819
(P9,((P8,P10),(P5,(P7,(P1,P4)))))                           0.000000

Merging clusters (P2,P3) and (P9,((P8,P10),(P5,(P7,(P1,P4))))) with distance 0.34
48187929913334
                                              P6  \
P6                                           0.0
((P2,P3),(P9,((P8,P10),(P5,(P7,(P1,P4))))))  0.5
```

```
                                                    ((P2,P3),(P9,((P8,P10),(P5,(P7,(P1,P
4))))))
P6
0.5
((P2,P3),(P9,((P8,P10),(P5,(P7,(P1,P4))))))
0.0

Merging clusters P6 and ((P2,P3),(P9,((P8,P10),(P5,(P7,(P1,P4))))))) with distance
0.5
                                                            (P6,((P2,P3),(P9,((P8,P10),(P5,
(P7,(P1,P4)))))))
(P6,((P2,P3),(P9,((P8,P10),(P5,(P7,(P1,P4)))))))
0.0
```

Visualizing Single Linkeage

In [102…

```
plt.figure(figsize=(12,5))
plt.title("Dendrogram with Single inkage")
dend = shc.dendrogram(shc.linkage(df[['a', 'b']], method='single'), labels=df.in
```



Complete Linkage Clustering

Interpretation from Euclidean Distance for Complete Link

- Notice that Point(P8,P10) is having the the least distance of all with 0.060828 so we'll conclude these both clusters as neighbouring clusters. Initially we'll merge those two points together considering the maximum distance between the cluster pointer which is again 0.060828. Then the next least distance is (P1, P4) with 0.127279, so we'll merge those two neighbouring clusters by calculating the maximum distance between cluster points which is again 0.127279 as we have only one point in each cluster and minimum distance have to be computed.
- Similarly calculating the least distance between each and every clusters and merging them based on the maximum distance between the neighbouring clusters is called

Single Linkage Clustering. This iterative process continues until all data points are clustered, revealing patterns in the data.

In [103…

```
def complete_linkage(dist_matrix):
    n = len(dist_matrix)
    while n > 1:
        min_val = float('inf')
        min_index = None
        for i in range(n):
            for j in range(i+1, n):
                if dist_matrix.iloc[i, j] < min_val and dist_matrix.index[i] !=
                    min_val = dist_matrix.iloc[i, j]
                    min_index = (i, j)

        if min_val == float('inf'):
            break

        i, j = min_index
        cluster1, cluster2 = dist_matrix.index[i], dist_matrix.columns[j]

        print(f'Merging clusters {cluster1} and {cluster2} with distance {min_va
        new_cluster = f'({cluster1},{cluster2})'

        dist_matrix[new_cluster] = dist_matrix[[cluster1, cluster2]].max(axis=1)
        dist_matrix.loc[new_cluster] = dist_matrix.loc[[cluster1, cluster2]].max
        dist_matrix = dist_matrix.drop([cluster1, cluster2], axis=0)
        dist_matrix = dist_matrix.drop([cluster1, cluster2], axis=1)

        n -= 1

        print(dist_matrix)
        print("")
print("Complete-Link Clustering:")
complete_linkage(dist.copy())
```

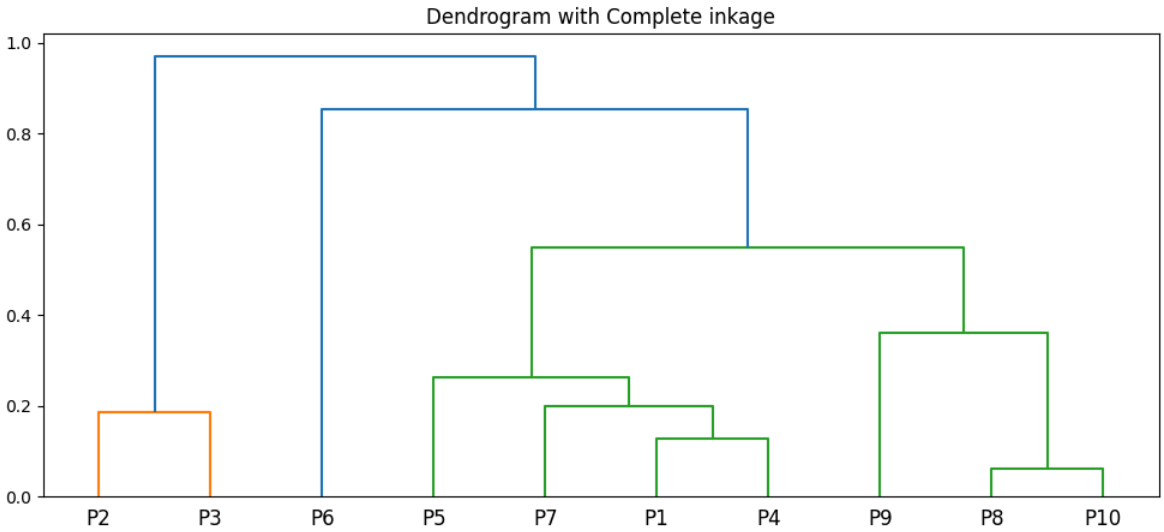
```
Complete-Link Clustering:
Merging clusters P8 and P10 with distance 0.060827625302982254
                P1        P2        P3        P4        P5        P6  \
P1        0.000000  0.854400  0.768505  0.127279  0.151327  0.650308
P2        0.854400  0.000000  0.186011  0.971700  0.833607  0.891572
P3        0.768505  0.186011  0.000000  0.893085  0.718679  0.711688
P4        0.127279  0.971700  0.893085  0.000000  0.250000  0.743303
P5        0.151327  0.833607  0.718679  0.250000  0.000000  0.500000
P6        0.650308  0.891572  0.711688  0.743303  0.500000  0.000000
P7        0.136015  0.787718  0.729178  0.200250  0.264197  0.741485
P9        0.424382  0.453982  0.344819  0.550000  0.380789  0.552268
(P8,P10)  0.370000  0.615549  0.609016  0.450999  0.458039  0.854283

                P7        P9  (P8,P10)
P1        0.136015  0.424382  0.370000
P2        0.787718  0.453982  0.615549
P3        0.729178  0.344819  0.609016
P4        0.200250  0.550000  0.450999
P5        0.264197  0.380789  0.458039
P6        0.741485  0.552268  0.854283
P7        0.000000  0.400000  0.250799
P9        0.400000  0.000000  0.360000
(P8,P10)  0.250799  0.360000  0.060828

Merging clusters P1 and P4 with distance 0.12727922061357852
                P2        P3        P5        P6        P7        P9  \
P2        0.000000  0.186011  0.833607  0.891572  0.787718  0.453982
P3        0.186011  0.000000  0.718679  0.711688  0.729178  0.344819
P5        0.833607  0.718679  0.000000  0.500000  0.264197  0.380789
P6        0.891572  0.711688  0.500000  0.000000  0.741485  0.552268
P7        0.787718  0.729178  0.264197  0.741485  0.000000  0.400000
P9        0.453982  0.344819  0.380789  0.552268  0.400000  0.000000
(P8,P10)  0.615549  0.609016  0.458039  0.854283  0.250799  0.360000
(P1,P4)   0.971700  0.893085  0.250000  0.743303  0.200250  0.550000

          (P8,P10)   (P1,P4)
P2        0.615549  0.971700
P3        0.609016  0.893085
P5        0.458039  0.250000
P6        0.854283  0.743303
P7        0.250799  0.200250
P9        0.360000  0.550000
(P8,P10)  0.060828  0.450999
(P1,P4)   0.450999  0.127279

Merging clusters P2 and P3 with distance 0.18601075237738277
                P5        P6        P7        P9  (P8,P10)   (P1,P4)   (P2,P3)
P5        0.000000  0.500000  0.264197  0.380789  0.458039  0.250000  0.833607
P6        0.500000  0.000000  0.741485  0.552268  0.854283  0.743303  0.891572
P7        0.264197  0.741485  0.000000  0.400000  0.250799  0.200250  0.787718
P9        0.380789  0.552268  0.400000  0.000000  0.360000  0.550000  0.453982
(P8,P10)  0.458039  0.854283  0.250799  0.360000  0.060828  0.450999  0.615549
(P1,P4)   0.250000  0.743303  0.200250  0.550000  0.450999  0.127279  0.971700
(P2,P3)   0.833607  0.891572  0.787718  0.453982  0.615549  0.971700  0.186011

Merging clusters P7 and (P1,P4) with distance 0.2002498439450078
                  P5        P6        P9  (P8,P10)   (P2,P3)  (P7,(P1,P4))
P5          0.000000  0.500000  0.380789  0.458039  0.833607      0.264197
P6          0.500000  0.000000  0.552268  0.854283  0.891572      0.743303
P9          0.380789  0.552268  0.000000  0.360000  0.453982      0.550000
```

```
(P8,P10)      0.458039  0.854283  0.360000  0.060828  0.615549      0.450999
(P2,P3)       0.833607  0.891572  0.453982  0.615549  0.186011      0.971700
(P7,(P1,P4))  0.264197  0.743303  0.550000  0.450999  0.971700      0.200250

Merging clusters P5 and (P7,(P1,P4)) with distance 0.2641968962724581
                          P6        P9  (P8,P10)   (P2,P3)  (P5,(P7,(P1,P4)))
P6                  0.000000  0.552268  0.854283  0.891572           0.743303
P9                  0.552268  0.000000  0.360000  0.453982           0.550000
(P8,P10)            0.854283  0.360000  0.060828  0.615549           0.458039
(P2,P3)             0.891572  0.453982  0.615549  0.186011           0.971700
(P5,(P7,(P1,P4)))   0.743303  0.550000  0.458039  0.971700           0.264197

Merging clusters P9 and (P8,P10) with distance 0.36000000000000004
                          P6   (P2,P3)  (P5,(P7,(P1,P4)))  (P9,(P8,P10))
P6                  0.000000  0.891572           0.743303       0.854283
(P2,P3)             0.891572  0.186011           0.971700       0.615549
(P5,(P7,(P1,P4)))   0.743303  0.971700           0.264197       0.550000
(P9,(P8,P10))       0.854283  0.615549           0.550000       0.360000

Merging clusters (P5,(P7,(P1,P4))) and (P9,(P8,P10)) with distance 0.55
                                       P6   (P2,P3)  \
P6                                 0.000000  0.891572
(P2,P3)                            0.891572  0.186011
((P5,(P7,(P1,P4))),(P9,(P8,P10)))  0.854283  0.971700

                                   ((P5,(P7,(P1,P4))),(P9,(P8,P10)))
P6                                                          0.854283
(P2,P3)                                                     0.971700
((P5,(P7,(P1,P4))),(P9,(P8,P10)))                           0.550000

Merging clusters P6 and ((P5,(P7,(P1,P4))),(P9,(P8,P10))) with distance 0.8542833
253669417
                                        (P2,P3)  \
(P2,P3)                                0.186011
(P6,((P5,(P7,(P1,P4))),(P9,(P8,P10))))  0.971700

                                       (P6,((P5,(P7,(P1,P4))),(P9,(P8,P10))))
(P2,P3)                                                              0.971700
(P6,((P5,(P7,(P1,P4))),(P9,(P8,P10))))                               0.854283

Merging clusters (P2,P3) and (P6,((P5,(P7,(P1,P4))),(P9,(P8,P10)))) with distance
0.9716995420396163
                                                    ((P2,P3),(P6,((P5,(P7,(P1,P
4))),(P9,(P8,P10)))))
((P2,P3),(P6,((P5,(P7,(P1,P4))),(P9,(P8,P10)))))
0.9717
```

Visualizing Complete Linkeage

In [104…

```
plt.figure(figsize=(12,5))
plt.title("Dendrogram with Complete inkage")
dend = shc.dendrogram(shc.linkage(df[['a', 'b']], method='complete'), labels=df.
```



Model Interpretation

Single Linkage:

- Single linkage clustering merges clusters based on the closest pair of data points between them.
- It prioritizes merging clusters with the smallest distances, indicating high similarity among their points.
- This process continues until all data points are in a single cluster, revealing the hierarchical structure of clustering.

Complete Linkage

- Complete linkage clustering merges clusters based on the maximum distance between any pair of data points, one from each cluster.
- It prioritizes merging clusters with the largest distances, indicating lower similarity among their points.
- Complete linkage clustering tends to produce compact, spherical clusters, as it focuses on the maximum dissimilarity between clusters.
- It is less sensitive to noise and outliers compared to single linkage clustering, as it considers the maximum distance rather than just the closest points.