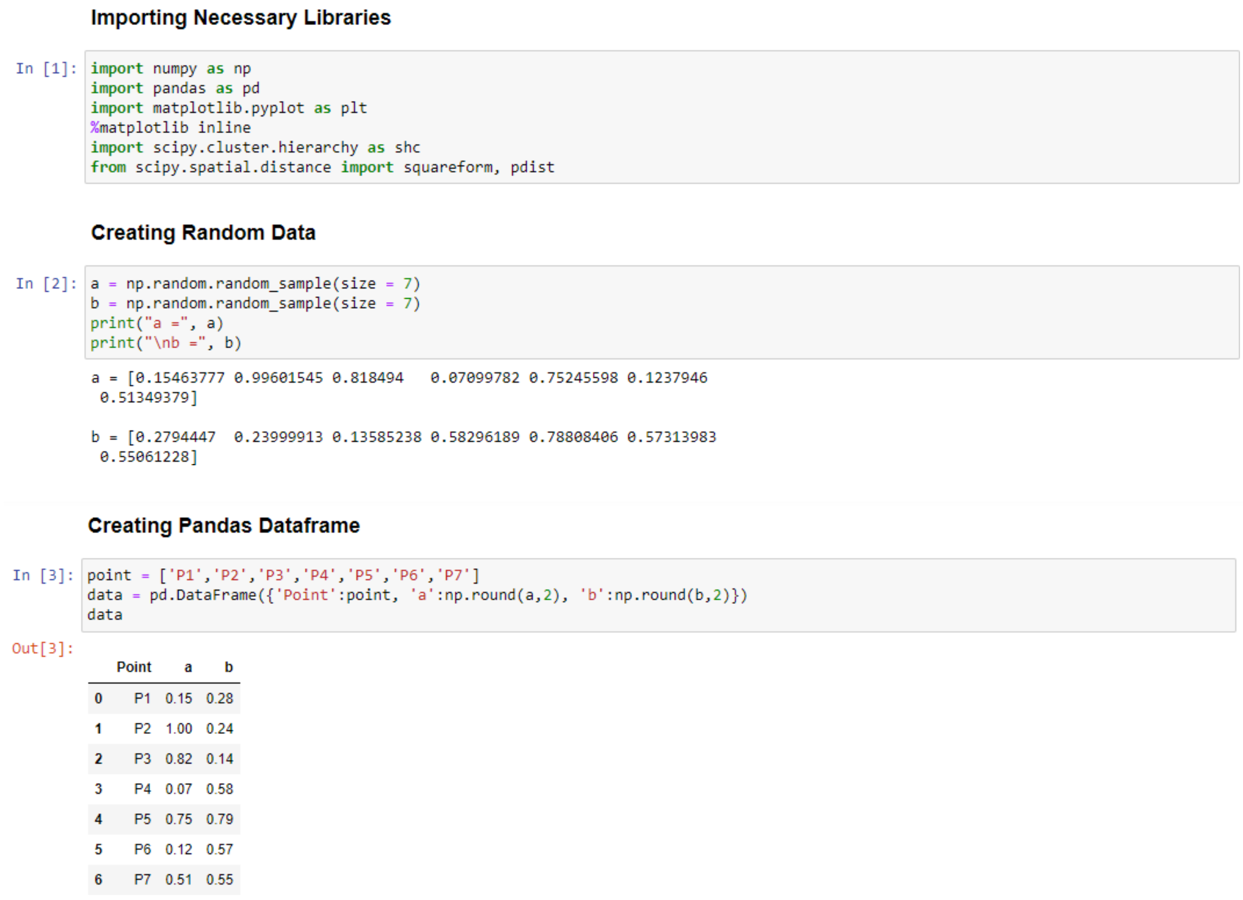
Experiment 8

Aditya Sawant
Subject: DWM
Class: TE4
Roll no: 46
Batch: D

Aim: Implementation of Single Link Agglomerative Hierarchical Clustering method



Importing Necessary Libraries

```
In [1]: 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
```

Creating Random Data

```
In [2]: a = np.random.random_sample(size = 7)
        b = np.random.random_sample(size = 7)
        print("a =", a)
        print("\nb =", b)

        a = [0.15463777 0.99601545 0.818494   0.07099782 0.75245598 0.1237946
         0.51349379]

        b = [0.2794447  0.23999913 0.13585238 0.58296189 0.78808406 0.57313983
         0.55061228]
```

Creating Pandas Dataframe

```
In [3]: point = ['P1','P2','P3','P4','P5','P6','P7']
        data = pd.DataFrame({'Point':point, 'a':np.round(a,2), 'b':np.round(b,2)})
        data
```

Out[3]:

	Point	a	b
0	P1	0.15	0.28
1	P2	1.00	0.24
2	P3	0.82	0.14
3	P4	0.07	0.58
4	P5	0.75	0.79
5	P6	0.12	0.57
6	P7	0.51	0.55

In [4]:

```
data = data.set_index('Point')
data
```

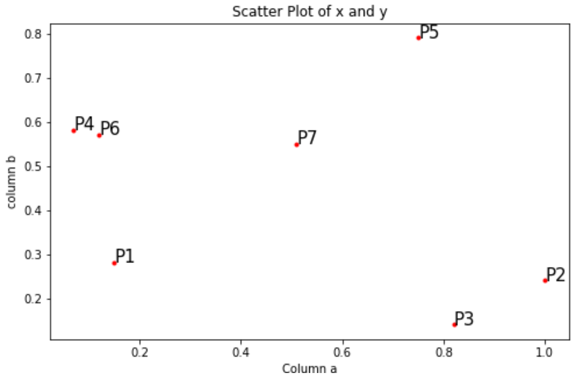
Out[4]:

	a	b
Point		
P1	0.15	0.28
P2	1.00	0.24
P3	0.82	0.14
P4	0.07	0.58
P5	0.75	0.79
P6	0.12	0.57
P7	0.51	0.55

Visualizing the data using a Scatter Plot.

In [5]:

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



Calculating the distance matrix in Euclidean method using pdist

In [6]:

```
dist = pd.DataFrame(squareform(pdist(data[['a','b']]), 'euclidean'), columns=data.index.values, index=data.index.values)
dist
```

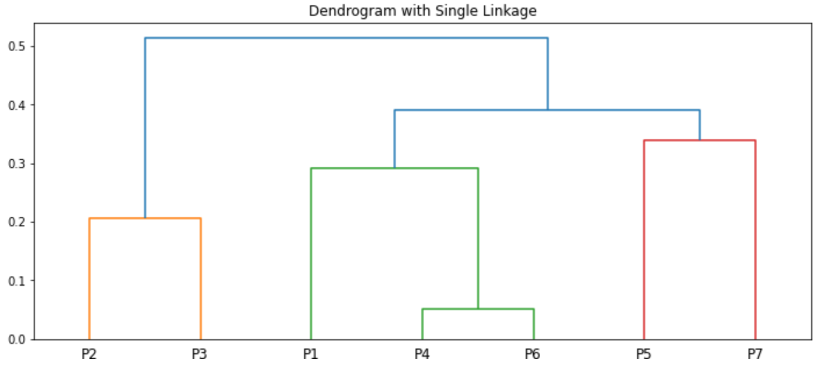
Out[6]:

	P1	P2	P3	P4	P5	P6	P7
P1	0.000000	0.850941	0.684471	0.310483	0.787464	0.291548	0.450000
P2	0.850941	0.000000	0.205913	0.990202	0.604152	0.939840	0.579828
P3	0.684471	0.205913	0.000000	0.869540	0.653758	0.821523	0.514004
P4	0.310483	0.990202	0.869540	0.000000	0.711688	0.050990	0.441022
P5	0.787464	0.604152	0.653758	0.711688	0.000000	0.667308	0.339411
P6	0.291548	0.939840	0.821523	0.050990	0.667308	0.000000	0.390512
P7	0.450000	0.579828	0.514004	0.441022	0.339411	0.390512	0.000000

Visualizing using a Dendrogram

In [7]:

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


Dendrogram with Single Linkage
P2
P3
P1
P4
P6
P5
P7