Homework 9 - Grant Jackson

November 19, 2024

0.0.1 HW9

- Create and display six dendrograms on the Covid data in a 3 by 2 subplots that use single, complete, average, weighted, centroid and ward linkage functions

[2]:
```
import os
os.chdir('C:\\Users\gmoor\Documents\Economic Analytics 1\Data')

import numpy as np
import pandas as pd
import math
```

[5]:
```
raw0 = pd.read_csv('covid.csv')
raw0.head()
```

[5]:

	state	positivelast7per1k	testpositivitylast7
0	AK	4.333295	0.061600
1	AL	2.184091	0.188831
2	AR	2.277153	0.102638
3	AZ	1.470177	0.116040
4	CA	0.853508	0.029735

[6]:
```
raw0.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 3 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   state                50 non-null     object
 1   positivelast7per1k   50 non-null     float64
 2   testpositivitylast7  50 non-null     float64
dtypes: float64(2), object(1)
memory usage: 1.3+ KB
```

[8]:
```
# Defining X as the numeric column
X = raw0[['positivelast7per1k', 'testpositivitylast7']]

X.head()
```

[8]:

	positivelast7per1k	testpositivitylast7
0	4.333295	0.061600
1	2.184091	0.188831
2	2.277153	0.102638
3	1.470177	0.116040
4	0.853508	0.029735

[11]:
```
# Scaling X
from sklearn.preprocessing import StandardScaler

Xn = StandardScaler().fit_transform(X)

Xn
```

[11]:
```
array([[ 0.81186246, -0.42544973],
       [-0.19647633,  0.76081869],
       [-0.15281458, -0.04282679],
       [-0.53142221,  0.08213548],
       [-0.82074369, -0.72254667],
       [ 0.06426255, -0.31338302],
       [-0.54281844, -0.72170883],
       [-0.60494876, -0.63663275],
       [-0.55499381, -0.48720127],
       [-0.67426114, -0.26941822],
       [-1.01739413, -0.83099769],
       [ 0.77931913,  2.32238268],
       [ 0.59950276,  2.24375   ],
       [ 0.57939518, -0.26009217],
       [ 0.28788493, -0.22279074],
       [ 0.22031206,  2.14713988],
       [ 0.16101228,  0.11111782],
       [-0.70476171, -0.57335697],
       [-0.52558166, -0.83564577],
       [-0.70449585, -0.71520524],
       [-1.01944421, -0.89032997],
       [-0.19733134, -0.38700321],
       [ 0.32201637, -0.17946565],
       [ 0.16115094,  0.14792602],
       [-0.3680199 , -0.37008069],
       [ 1.56763524,  0.57255928],
       [-0.39808676, -0.40054704],
       [ 3.80811574,  0.16726301],
       [ 0.77296905, -0.06830302],
       [-0.88441382, -0.78520526],
       [-0.57200946, -0.63939906],
       [ 0.19337375, -0.19493364],
       [-0.20566724, -0.07300756],
```

```
       [-0.83764443, -0.85795011],
       [-0.29023909, -0.42309611],
       [-0.22213233, -0.08664461],
       [-0.81071817, -0.26228808],
       [-0.58665769,  0.24134505],
       [-0.20654754, -0.6991616 ],
       [-0.43004915, -0.45383038],
       [ 3.64891308,  3.6320571 ],
       [-0.02998274, -0.09173507],
       [-0.46419324, -0.02677363],
       [ 0.73878471,  0.56092083],
       [-0.67041741, -0.4108951 ],
       [-1.10614423, -0.95358344],
       [-0.84189954, -0.56228554],
       [ 1.70256167,  0.2832151 ],
       [-0.39104258, -0.58856516],
       [ 1.1442813 ,  3.18970885]])
```

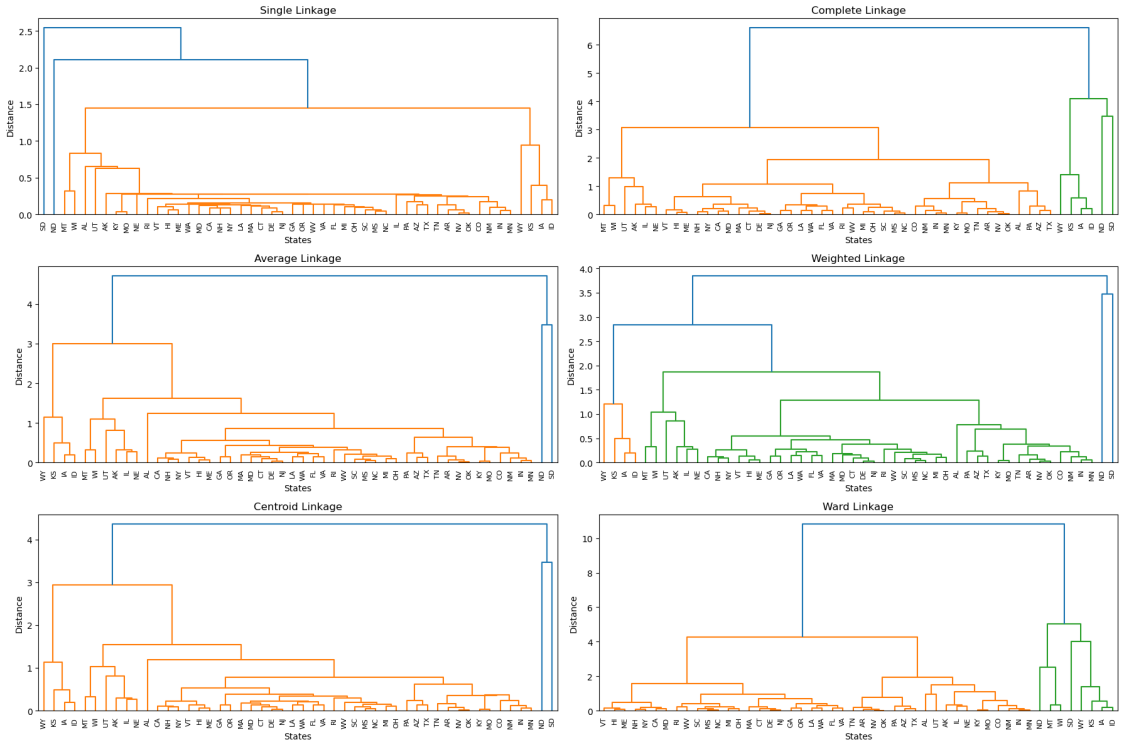
[12]:
```
# Generating and plotting dendrograms
from scipy.cluster import hierarchy
import matplotlib.pyplot as plt

plt.figure(figsize=(18, 12)) # Single figure for all subplots

methods = ['single', 'complete', 'average', 'weighted', 'centroid', 'ward'] #␣
 ↪Linkage methods

for i, method in enumerate(methods, 1):
    plt.subplot(3, 2, i) # 3x2 subplot format
    Z = hierarchy.linkage(Xn, method)
    hierarchy.dendrogram(Z, labels=raw0['state'].tolist(), leaf_rotation=90,␣
 ↪leaf_font_size=8)
    plt.title(f"{method.capitalize()} Linkage") # Dendrogram title
    plt.xlabel("States")
    plt.ylabel("Distance")

plt.tight_layout() # Prevent overlapping
plt.show()
```


Single Linkage
Complete Linkage
Average Linkage
Weighted Linkage
Centroid Linkage
Ward Linkage
Distance
States

[ ]: