Yurim Park hw3

1.

a) 37.75%

a)

$$p(x) = \frac{e^{\beta_0+\beta_1 \times X_1+\beta_2 \times X_2}}{1+e^{\beta_0+\beta_1 \times X_1+\beta_2 \times X_2}}$$

$\beta_0 = -6 \quad \beta_1 = 0.05 \quad \beta_2 = 1 \quad X_1 = 40 \quad X_2 = 3.5$

$$p(x) = \frac{e^{-6+0.05 \times 40+1 \times 3.5}}{1+e^{-6+0.05 \times 40+1 \times 3.5}} \approx 0.3775$$

b) 50hours

b)

$$p(x) = \frac{e^{-6+0.05 \times X_1+1 \times 3.5}}{1+e^{-6+0.05 \times X_1+1 \times 3.3}} = \frac{1}{2}$$

$$e^{-6+0.05 \times X_1+1 \times 3.5} = 1$$

$$-6+0.05 \times X_1+1 \times 3.5 = 0$$

$$X_1 = 50$$

c) Specifically, β0=−6 implies that when both X1(hours studied) and X2 (undergrad GPA) are zero, the log odds of receiving an A is -6.

2.

a)

Mean:

Year 1999.993019

Lag1 0.203541

Lag2 0.203747

Lag3 0.207269

Lag4 0.205614

Lag5 0.206440

Volume 1.011219

Today 0.200951

dtype: float64

Standard Deviation:

Year	3.166690
Lag1	2.289741
Lag2	2.289738
Lag3	2.291947
Lag4	2.292765
Lag5	2.292686
Volume	0.506743
Today	2.290949

dtype: float64

Median:

Year	2000.0000
Lag1	0.3120
Lag2	0.3120
Lag3	0.3120
Lag4	0.3120
Lag5	0.3120
Volume	0.9818
Today	0.3120

dtype: float64

Minimum Values:

Year	1995.000000
Lag1	-11.050000
Lag2	-11.050000
Lag3	-11.050000
Lag4	-11.050000
Lag5	-11.050000
Volume	0.241088

Today -11.050000

dtype: float64

Maximum Values:

Year 2005.00000

Lag1 7.78000

Lag2 7.78000

Lag3 7.78000

Lag4 7.78000

Lag5 7.78000

Volume 2.48811

Today 7.78000

dtype: float64

```
Correlation matrix for numeric variables:
            Year      Lag1      Lag2      Lag3      Lag4      Lag5    Volume
Year    1.000000 -0.072070 -0.071078 -0.073912 -0.072937 -0.072944  0.929832
Lag1   -0.072070  1.000000 -0.073905  0.046598 -0.032020 -0.045843 -0.081740
Lag2   -0.071078 -0.073905  1.000000 -0.073803  0.046549 -0.032035 -0.129009
Lag3   -0.073912  0.046598 -0.073803  1.000000 -0.074918  0.046584 -0.106649
Lag4   -0.072937 -0.032020  0.046549 -0.074918  1.000000 -0.075003 -0.098701
Lag5   -0.072944 -0.045843 -0.032035  0.046584 -0.075003  1.000000 -0.085177
Volume  0.929832 -0.081740 -0.129009 -0.106649 -0.098701 -0.085177  1.000000
Today  -0.076954 -0.073805  0.046626 -0.031792 -0.045541 -0.028186 -0.044533

           Today
Year   -0.076954
Lag1   -0.073805
Lag2    0.046626
Lag3   -0.031792
Lag4   -0.045541
Lag5   -0.028186
Volume -0.044533
Today   1.000000
```

b)

```
Optimization terminated successfully.
         Current function value: 1.788334
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:           Direction_Up   No. Observations:                  573
Model:                          Logit   Df Residuals:                      566
Method:                           MLE   Df Model:                            6
Date:                Fri, 22 Mar 2024   Pseudo R-squ.:                     inf
Time:                        16:44:20   Log-Likelihood:                -1024.7
converged:                       True   LL-Null:                        0.0000
Covariance Type:            nonrobust   LLR p-value:                     1.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.4215      0.199      2.122      0.034       0.032       0.811
Lag1          -0.0138      0.037     -0.369      0.712      -0.087       0.060
Lag2           0.0469      0.038      1.236      0.217      -0.027       0.121
Lag3          -0.0125      0.038     -0.329      0.742      -0.087       0.062
Lag4          -0.0363      0.038     -0.959      0.337      -0.111       0.038
Lag5          -0.0650      0.038     -1.721      0.085      -0.139       0.009
Volume        -0.1582      0.172     -0.921      0.357      -0.495       0.178
==============================================================================
```

None of the lag variables (Lag1, Lag2, Lag3, Lag4, Lag5) or the Volume variable appear to be statistically significant, as their p-values are all greater than 0.05. Therefore, based on the logistic regression results, we cannot conclude that any of these predictors are statistically significant in predicting the direction of the response variable.

c)

Confusion Matrix:

[[27 225]

 [35 286]]

Accuracy: 0.5462

False Positive Rate: 89.2857%

False Negative Rate: 10.9034%

d)

Confusion Matrix with Lag2 Predictor:

[[134   8]

 [108 11]]

Overall, Fraction of Correct Predictions with Lag2 Predictor: 0.5555555555555556

Percent of False Positives with Lag2 Predictor: 42.10526315789473

Percent of False Negatives with Lag2 Predictor: 44.62809917355372

e)

Confusion Matrix:

[[134   8]

 [108 11]]

Overall, Fraction of Correct Predictions: 0.5556

Percent of False Positives: 41.38%

Percent of False Negatives: 3.07%

f)

Confusion Matrix:

[[83 59]

 [68 51]]

Overall, Fraction of Correct Predictions: 0.5134

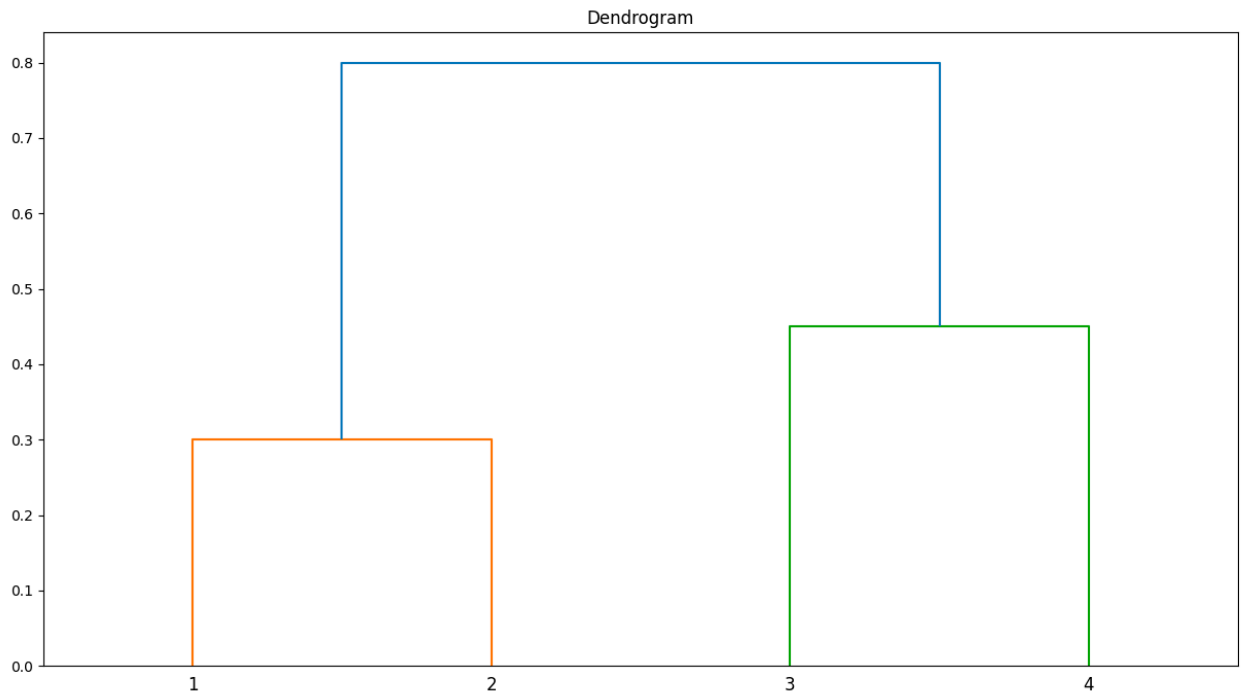
Percent of False Positives: 26.05%

Percent of False Negatives: 22.61%

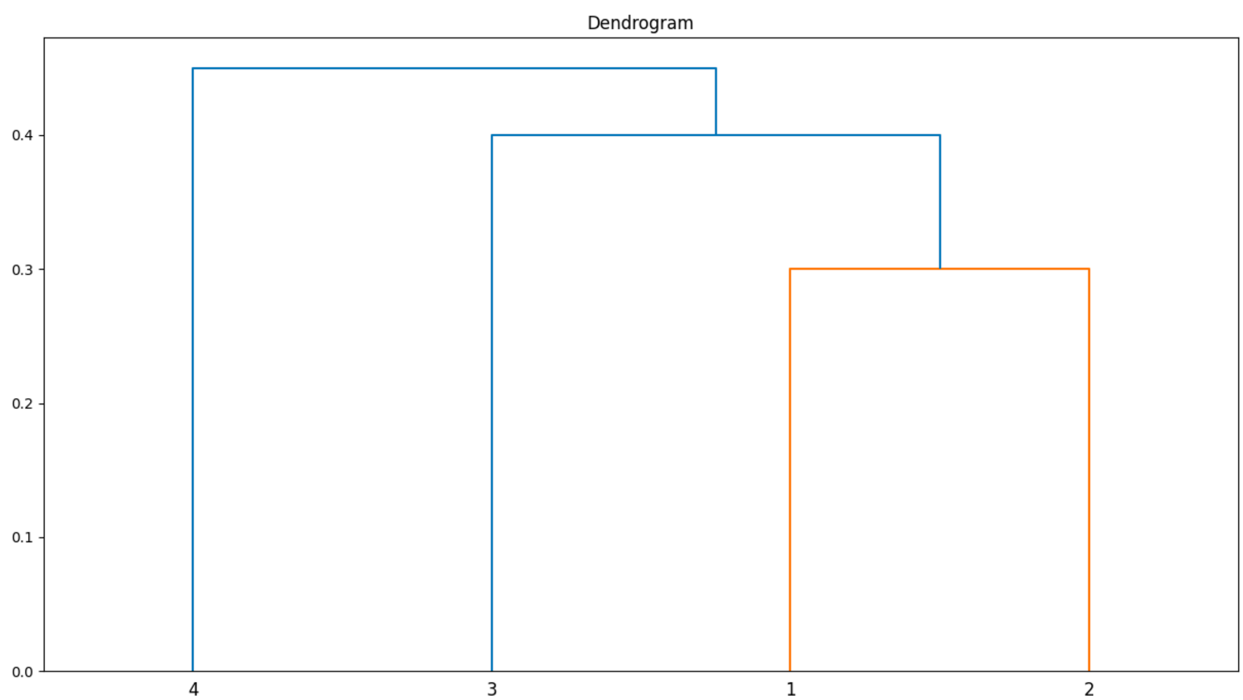
g)

3.

a)


Dendrogram
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4

b)


Dendrogram
0.4
0.3
0.2
0.1
0.0
4
3
1
2

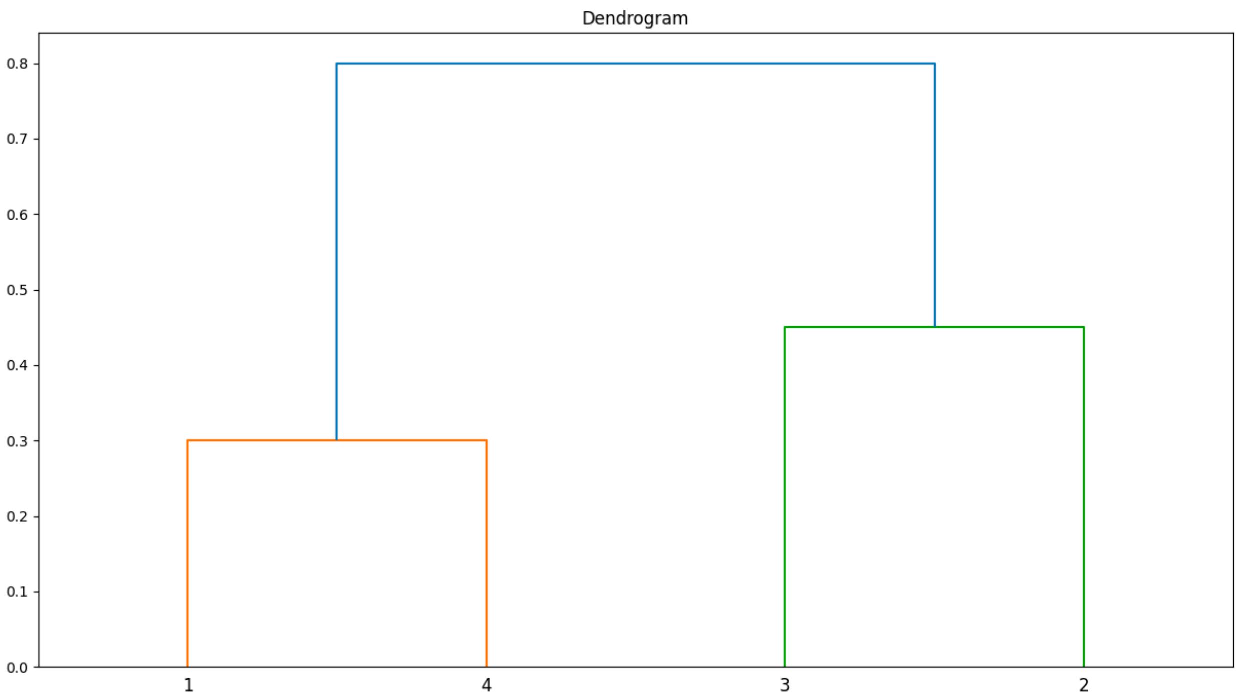
c)

Observations 1 and 2 are in Cluster A and 3 and 4 in Cluster B.

d)

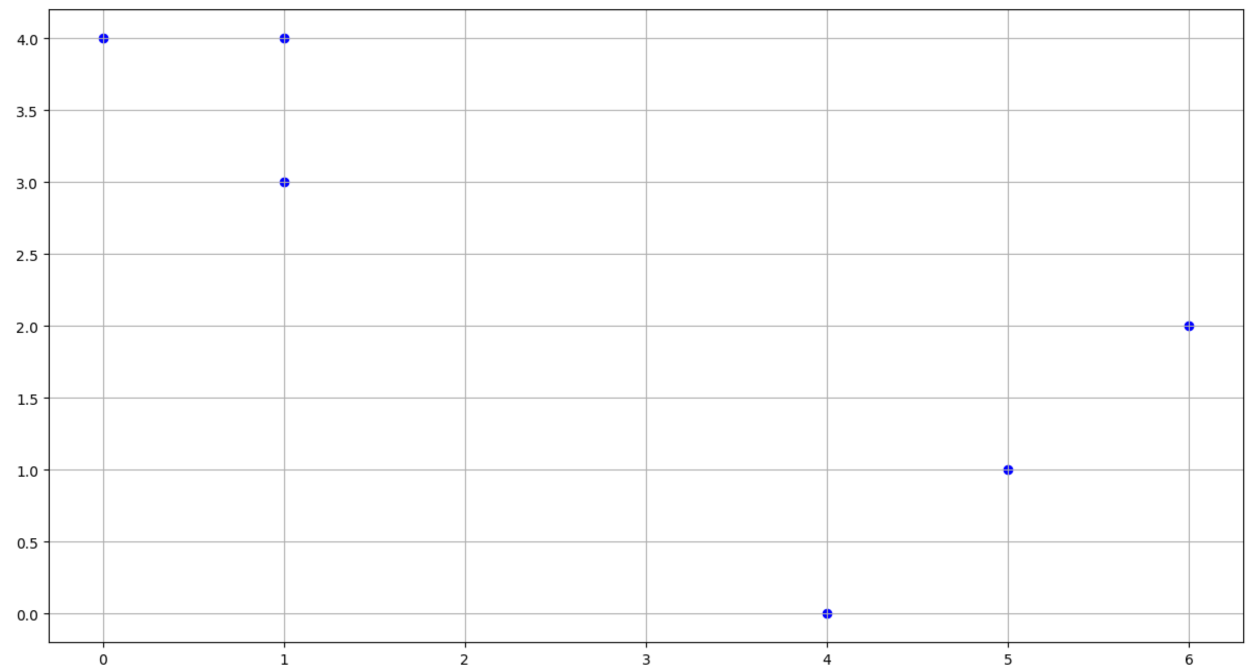
Observations 1, 2 and 3 are in Cluster A and 4 in Cluster B.

e)

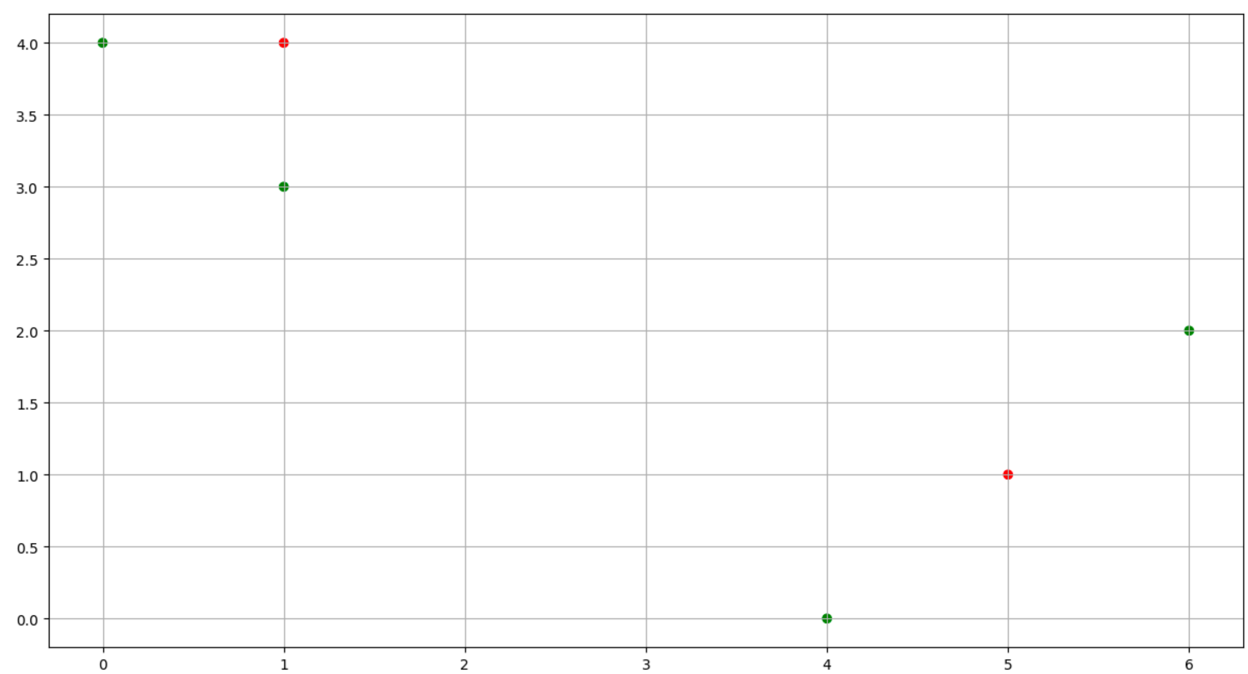


4.

a)



b)

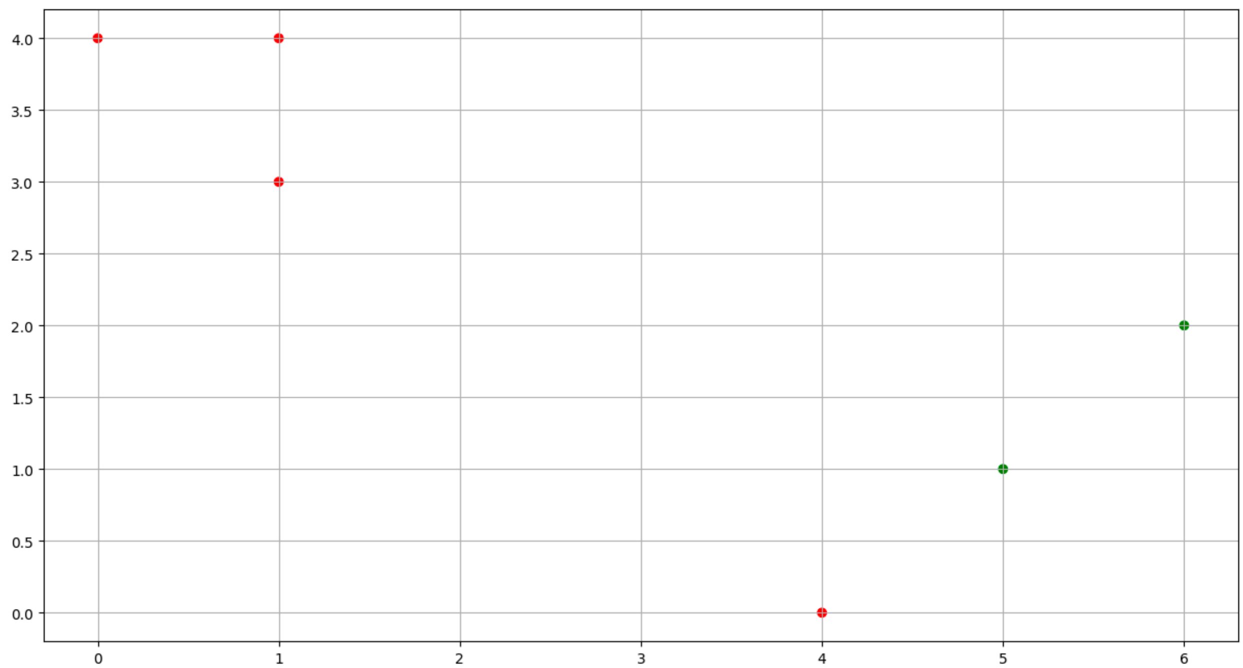


c)

Centriod for Clutser 0 is: 3.0, 2.5

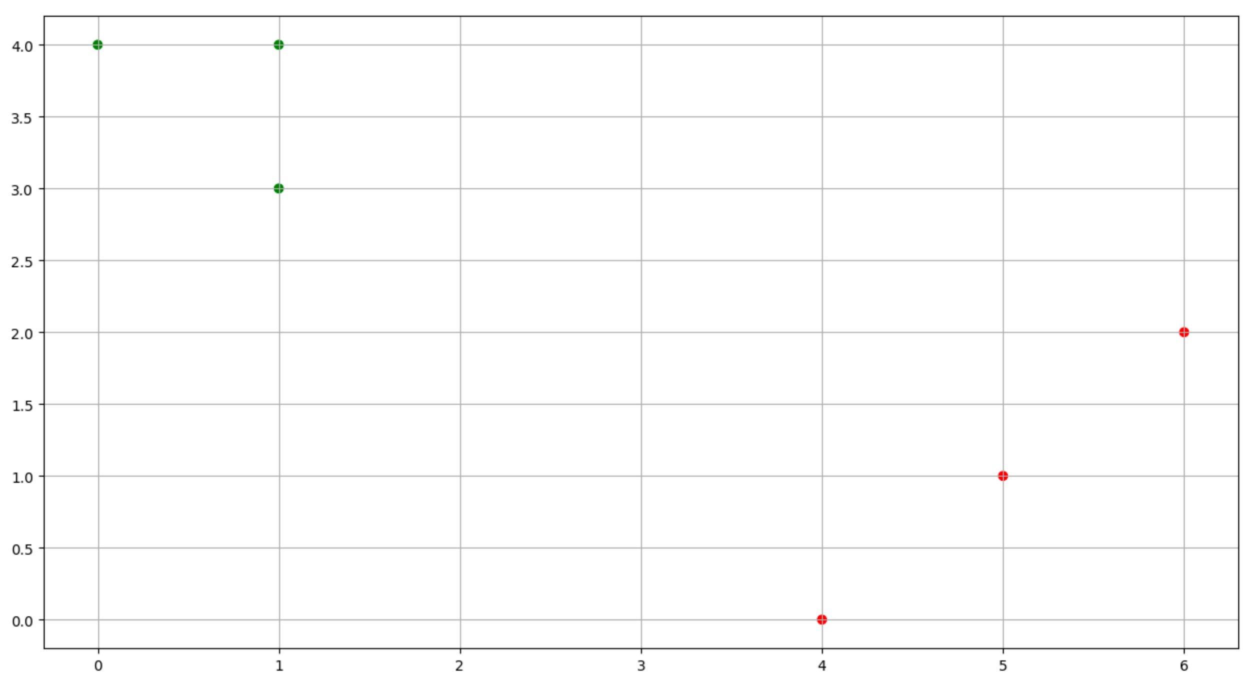
Centriod for Clutser 1 is: 2.75, 2.25

d)



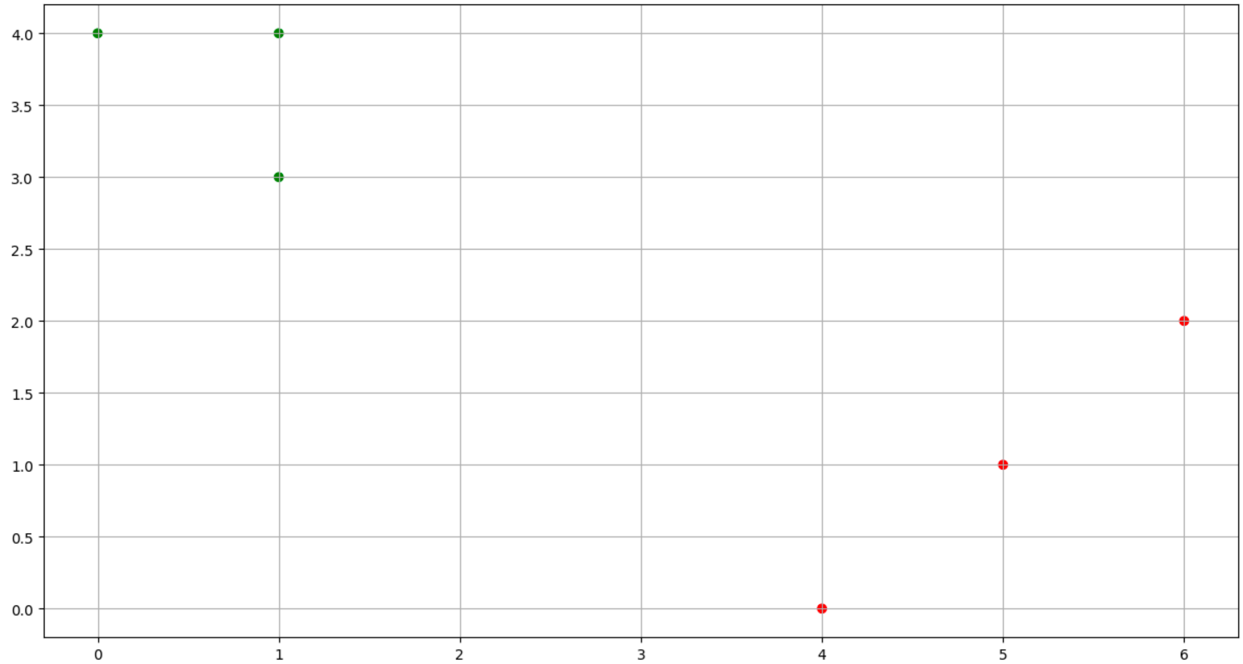
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6

e)



4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6

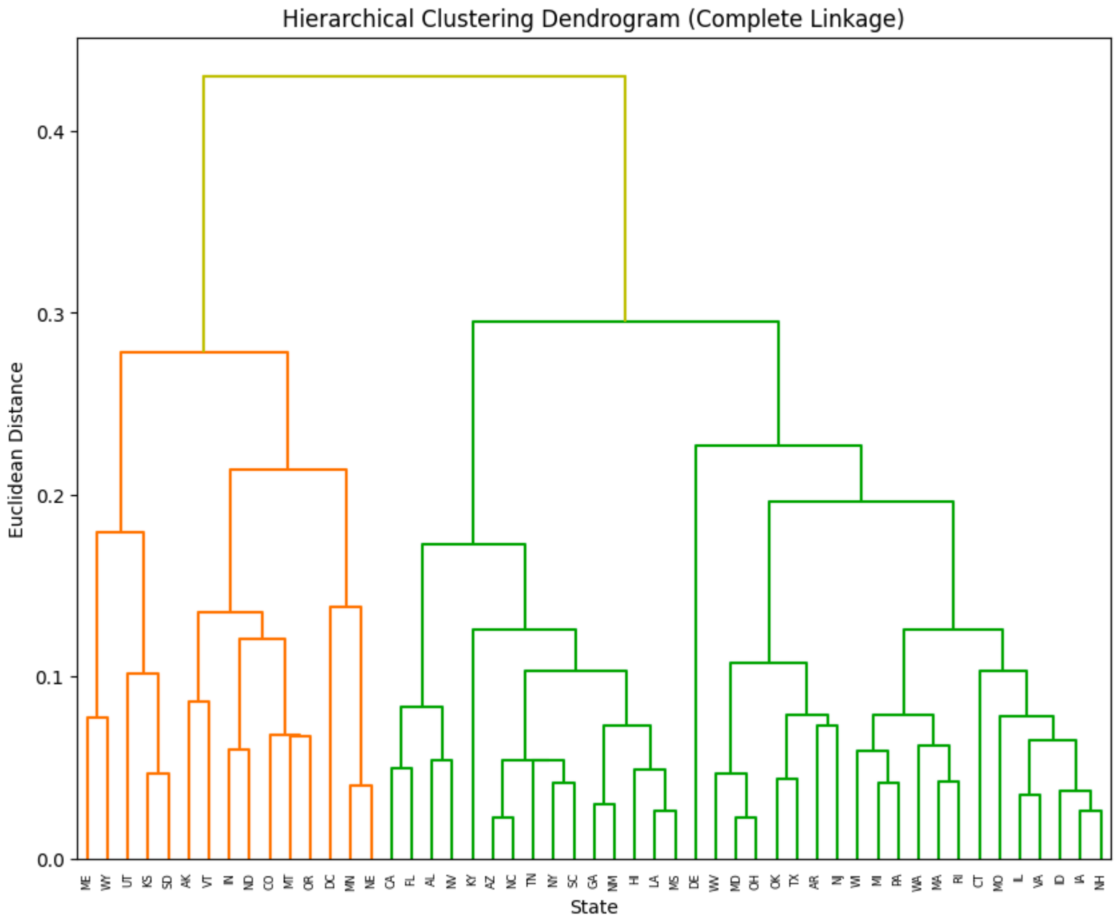
f)



5.

a)

```
import pandas as pd
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

df = pd.read_csv('CEV2021(1).csv')
states = df['State'].
data = df.drop(['State'], axis=1)

Z = linkage(data, method='complete', metric='euclidean')

plt.figure(figsize=(10, 8))
dendrogram(Z, labels=states, above_threshold_color='y', orientation='top')
plt.title('Hierarchical Clustering Dendrogram (Complete Linkage)')
plt.xlabel('State')
plt.ylabel('Euclidean Distance')
plt.xticks(rotation=90)
plt.show()
```



b)

```
Cluster: 1 AK, CO, DC, IN, KS, ME, MN, MT, NE, ND, OR, SD, UT, VT, WY
Cluster: 2 AL, AZ, CA, FL, GA, HI, KY, LA, MS, NV, NM, NY, NC, SC, TN
Cluster: 3 AR, CT, DE, ID, IL, IA, MD, MA, MI, MO, NH, NJ, OH, OK, PA, RI, TX, VA, WA, WV, WI
```

c)

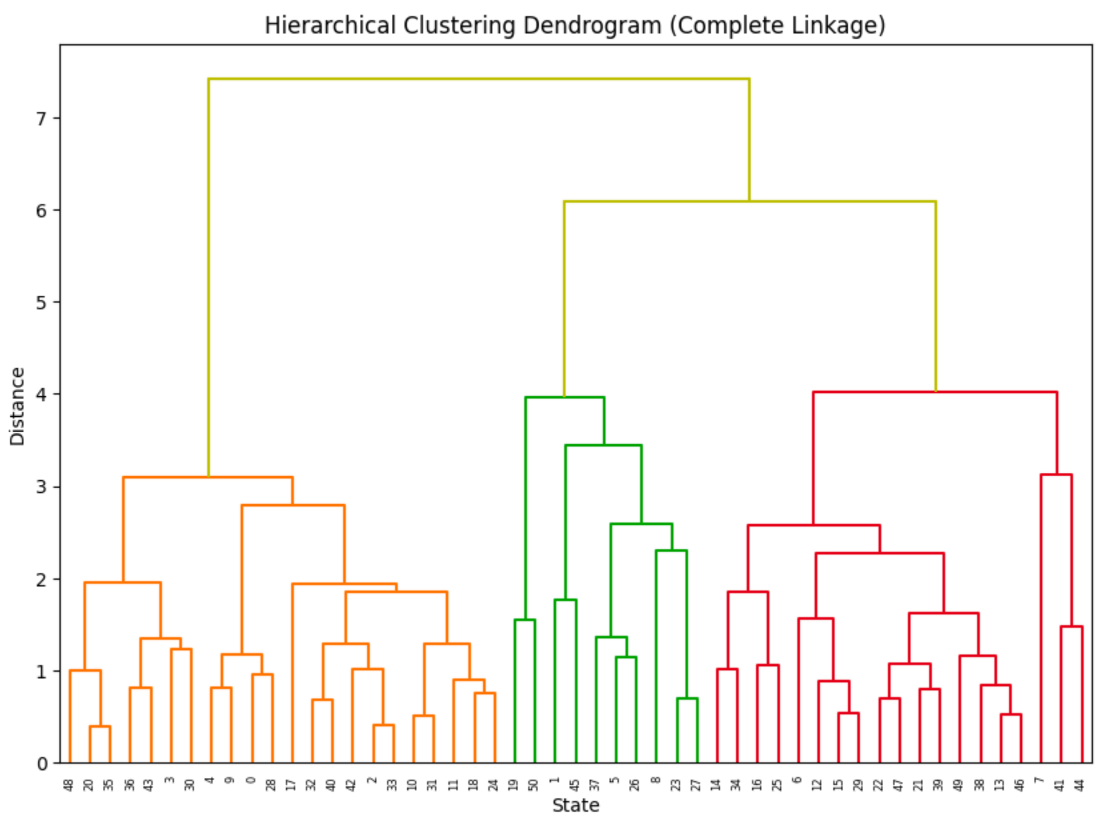
```
Cluster: 1 TX, WV
Cluster: 2 CT, IA, MI, PA
Cluster: 3 AR, DE, IL, MO, NH, NJ, VA, WA
Cluster: 4 ID, MD, MA, OH, OK, RI, WI
```

d)

```
Cluster: 0 WI
Cluster: 1 ID, MA, OH, OK
Cluster: 2 MD, RI
```

e)

```python
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

Z = linkage(scaled_data, method='complete', metric='euclidean')

# Plot the dendrogram
plt.figure(figsize=(10, 7))  # Adjust the size as needed
dendrogram(Z, above_threshold_color='y', orientation='top')
plt.title('Hierarchical Clustering Dendrogram (Complete Linkage)')
plt.xlabel('State')
plt.ylabel('Distance')
plt.xticks(rotation=90)  # Rotate state names for better readability
plt.show()
```



f)

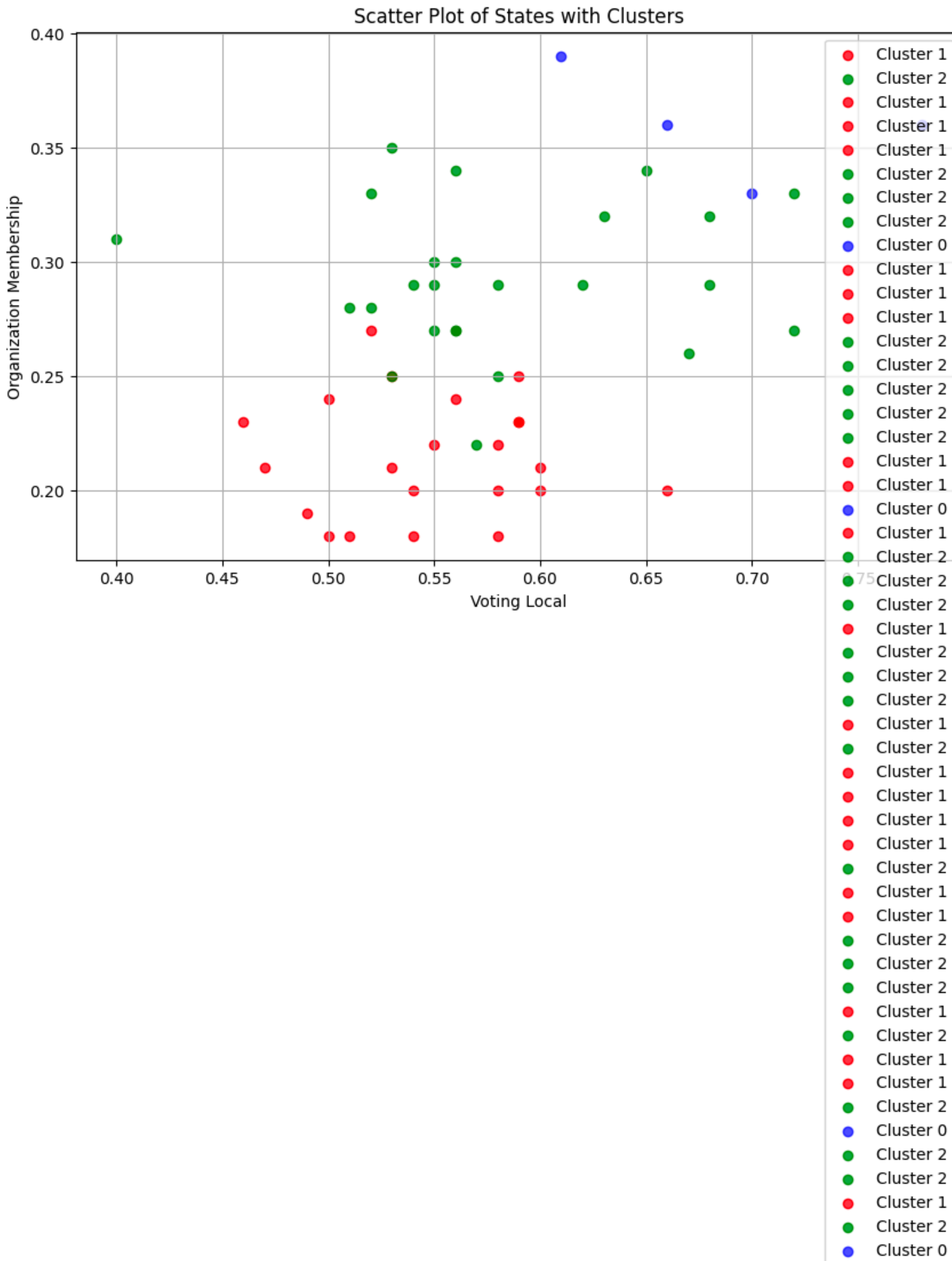
```
Cluster 1:  ['AL', 'AZ', 'AR', 'CA', 'FL', 'GA', 'HI', 'KY', 'LA', 'MD', 'MS', 'NV', 'NJ', 'NM', 'NY', 'NC', 'OH', 'OK', 'SC', 'TN', 'TX', 'WV']
Cluster 2:  ['AK', 'CO', 'DC', 'ME', 'MN', 'MT', 'NE', 'OR', 'VT', 'WY']
Cluster 3:  ['CT', 'DE', 'ID', 'IL', 'IN', 'IA', 'KS', 'MA', 'MI', 'MO', 'NH', 'ND', 'PA', 'RI', 'SD', 'UT', 'VA', 'WA', 'WI']
```

g)

```python
from sklearn.cluster import KMeans

# Perform K-means clustering with K=3 on the scaled data
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(scaled_data)

# Get cluster labels for each state
cluster_labels = kmeans.labels_

# Mapping clusters to states
cluster_assignment = {state: cluster for state, cluster in zip(states,
cluster_labels)}
# Initialize dictionaries to hold lists of states for each cluster
clusters_states = {i: [] for i in range(3)}

# Populate the dictionaries with states grouped by their cluster
for state, cluster in cluster_assignment.items():
    clusters_states[cluster].append(state)

# Print the states in each cluster
for cluster, states in clusters_states.items():
    print(f"Cluster {cluster}: {', '.join(states)}")
```

```
Cluster 0: DC, ME, VT, WY
Cluster 1: AL, AZ, AR, CA, FL, GA, HI, KY, LA, MD, MS, NV, NJ, NM, NY, NC, OH, OK, SC, TN, TX, WV
Cluster 2: AK, CO, CT, DE, ID, IL, IN, IA, KS, MA, MI, MN, MO, MT, NE, NH, ND, OR, PA, RI, SD, UT, VA, WA, WI
```

h)



Scatter Plot of States with Clusters
Organization Membership
Voting Local
0.40
0.35
0.30
0.25
0.20
0.40
0.45
0.50
0.55
0.60
0.65
0.70
Cluster 1
Cluster 2
Cluster 1
Cluster 1
Cluster 1
Cluster 2
Cluster 2
Cluster 2
Cluster 0
Cluster 1
Cluster 1
Cluster 1
Cluster 2
Cluster 2
Cluster 2
Cluster 2
Cluster 2
Cluster 1
Cluster 1
Cluster 0
Cluster 1
Cluster 2
Cluster 2
Cluster 2
Cluster 1
Cluster 2
Cluster 2
Cluster 2
Cluster 1
Cluster 2
Cluster 1
Cluster 1
Cluster 1
Cluster 1
Cluster 2
Cluster 1
Cluster 1
Cluster 2
Cluster 2
Cluster 2
Cluster 1
Cluster 2
Cluster 1
Cluster 1
Cluster 2
Cluster 0
Cluster 2
Cluster 2
Cluster 1
Cluster 2
Cluster 0