CS5785 HW3

3.(a)

k means can also be viewed as a special case of EM algorithm, but it assumes clusters are spherical distributed.

- E step: given computed centroids(means), assign each data point label according to the closest centroid class.
- M step: given the newly assigned label, compute the new centroid(mean) for each class.

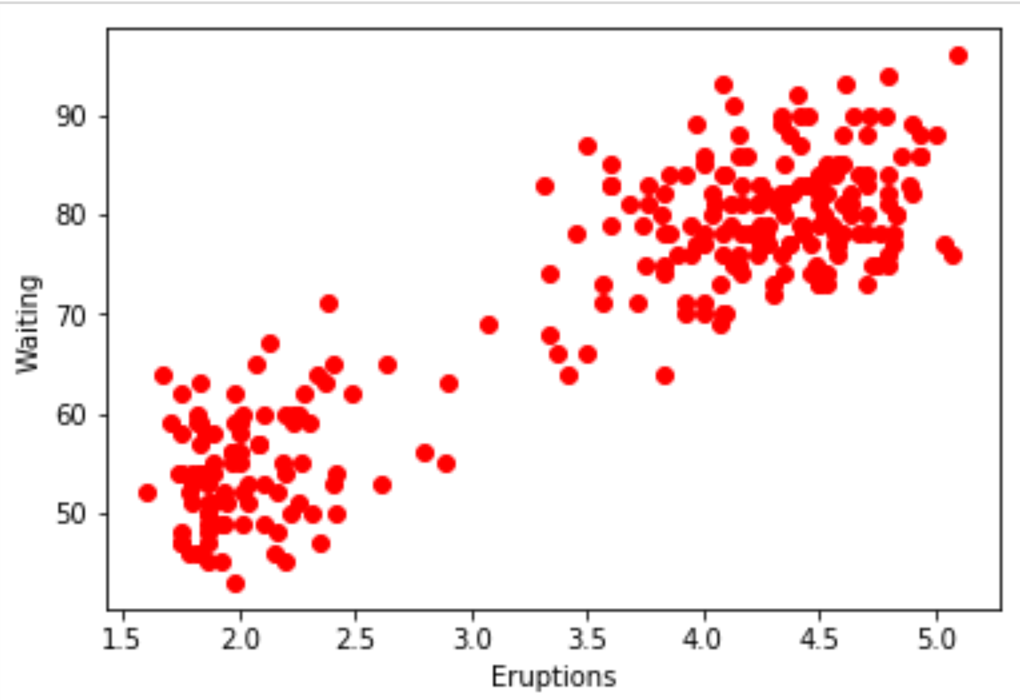
3.(a). let x_i $(x_1, \dots, x_n)$ be data points, μ_j denote cluster centroids, c_i be the label for each datapoint x_i

E: $c_i = \arg\min_j \|x_i - \mu_j\|^2$ (for each datapoint, choose closest centroid and assign its label.)

M: $$\mu_j = \frac{\sum_{i=1}^{N} 1\{c_i = j\} \cdot x_i}{\sum_{i=1}^{N} 1\{c_i = j\}}$$ (calculate mean of all datapoint that were labeled as c_i.)

3.(b)

parse the data and plot all points on 2D plane.

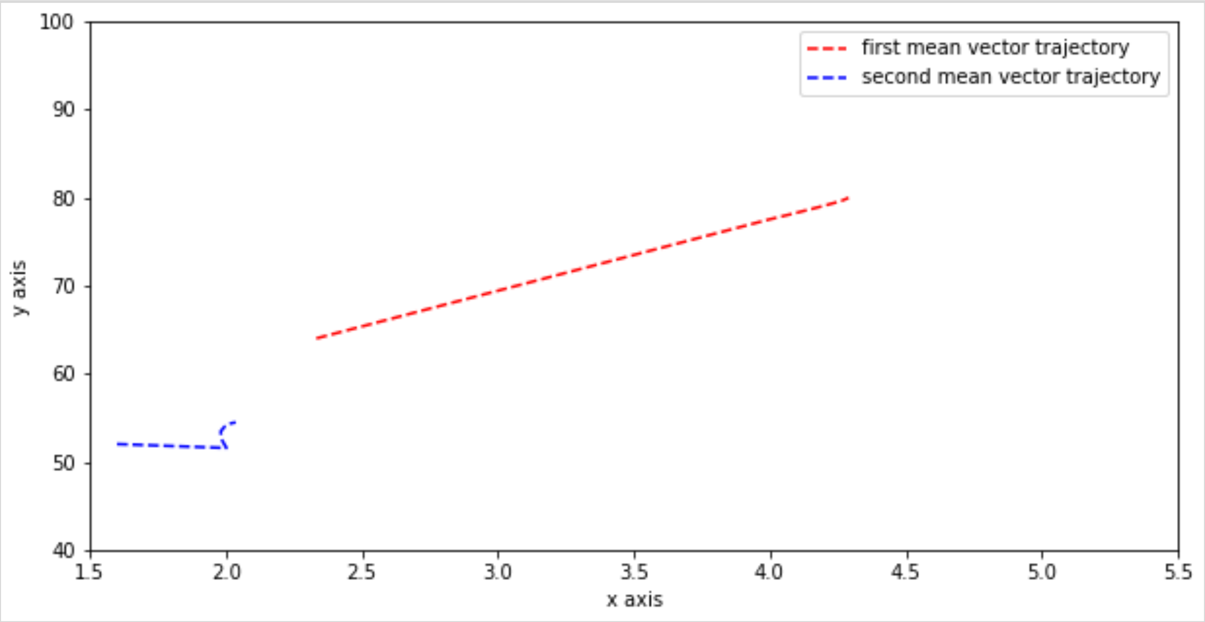


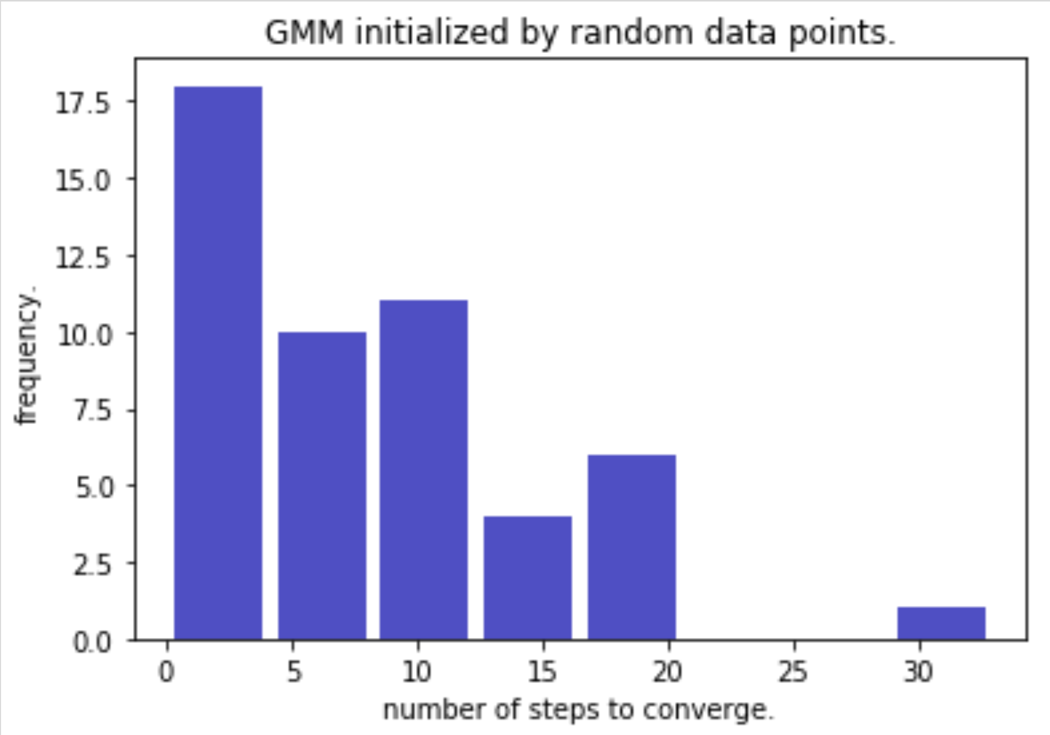
3.(c)

We implement GMM by ourselves. there are several assumptions we make:

- initial mu in Gaussian distribution is initialized by randomly choosing from existing datapoint.
- initial sigmas in Gaussian distribution is initialized by randomly sampling from (1, 6).

The below graph shows the means vector trajectory for a single random initialization.





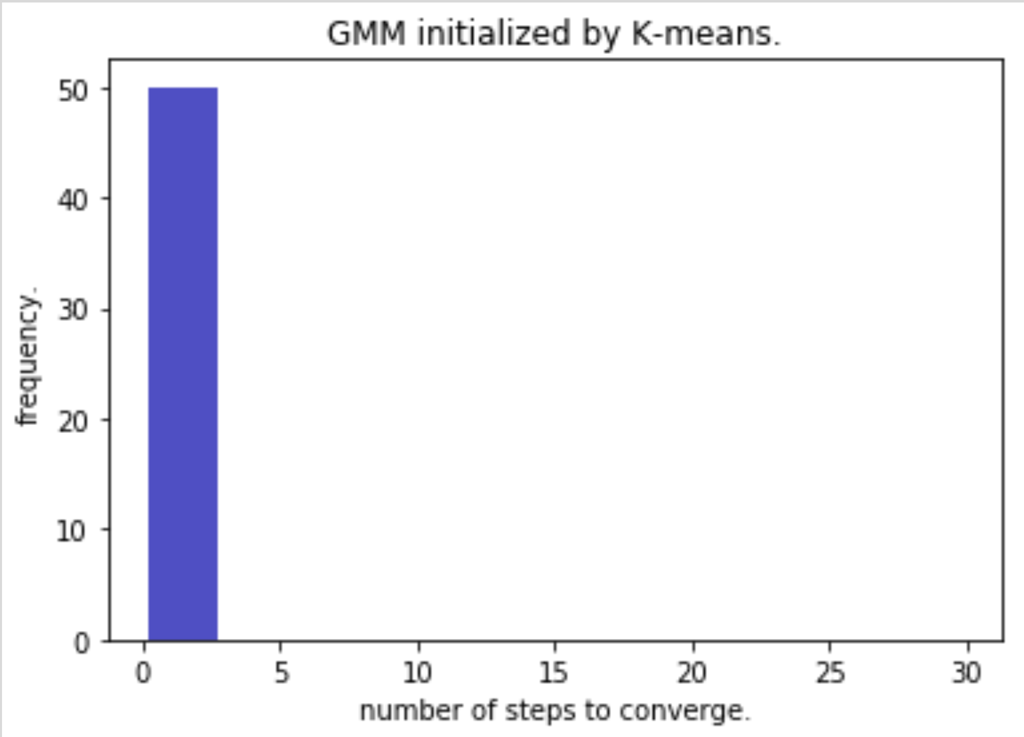
3.(d)

```
# new mean by maxmimum likelihood estimation on K-means.
[array([ 4.29793023, 80.28488372]), array([ 2.09433, 54.75    ])]



# new variance by maxmimum likelihood estimation on K-means.
[array([[ 0.17761717,  0.76310127],
       [ 0.76310127, 31.48279475]]),
 array([[ 0.1542787,  0.9856625],
       [ 0.9856625, 34.4075    ]])]
```

we can see the performance of initialization using k-means is way better than the random initialization as it takes less iterations to converge.



4.(a) i

assumptions:

- We assume that we can use euclidean distances to approximate Nei's distance.
- We assume that m dimensions yielded by MDS are suffice to simulate the original data relations.
- We also assume that the positions(coordinates) of the MDS-transformed points are't related to its original meaning, as it is subject to rotation, translation and reflection and therefore are not unique.

circumstances that it could fail:

When the original data are in a high dimensional space, reducing them into 2-dimensional space with MDS may not be able to correctly represent their original structure.

how could be measure:

We can measure the total information loss by measuring the sum difference between the predicted distances and their original distances.

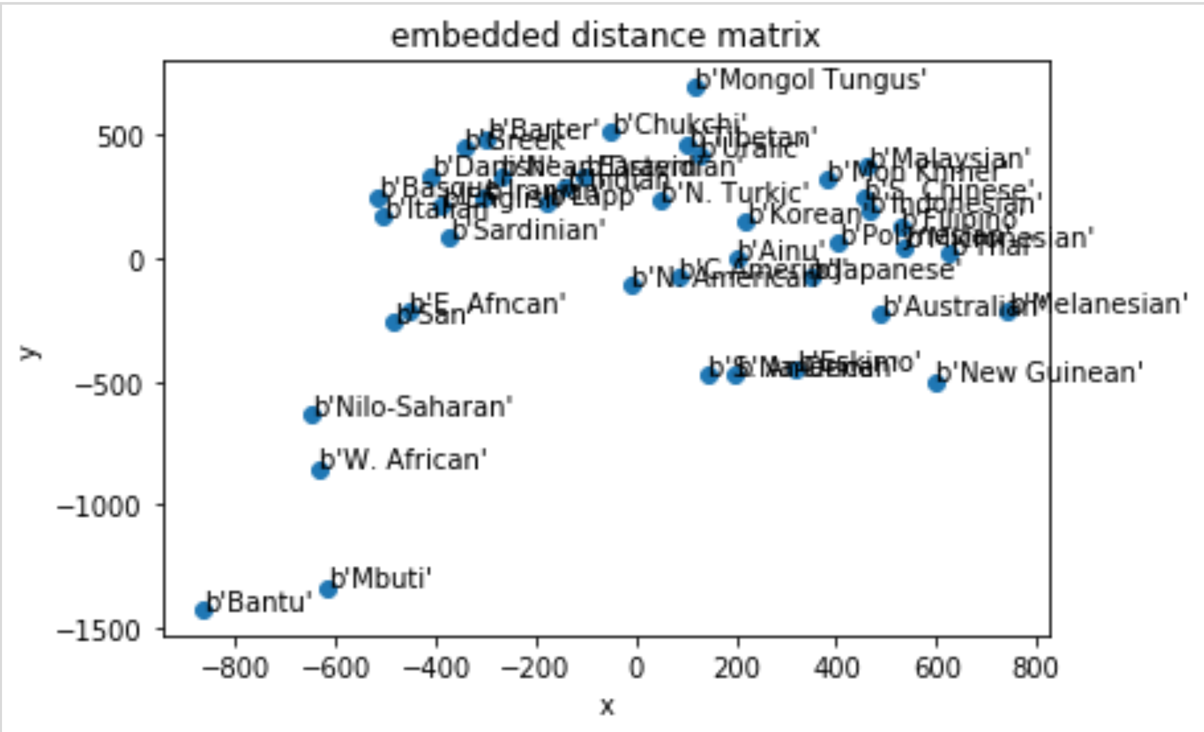
total loss = np.linalg.norm(D - D') / np.linalg.norm(D)

where D represents the original dissimilarity matrix, and D' represents the computed distance matrix, where each entry represents the euclidean distance between two objects after MDS transformation.

4.(a) ii.

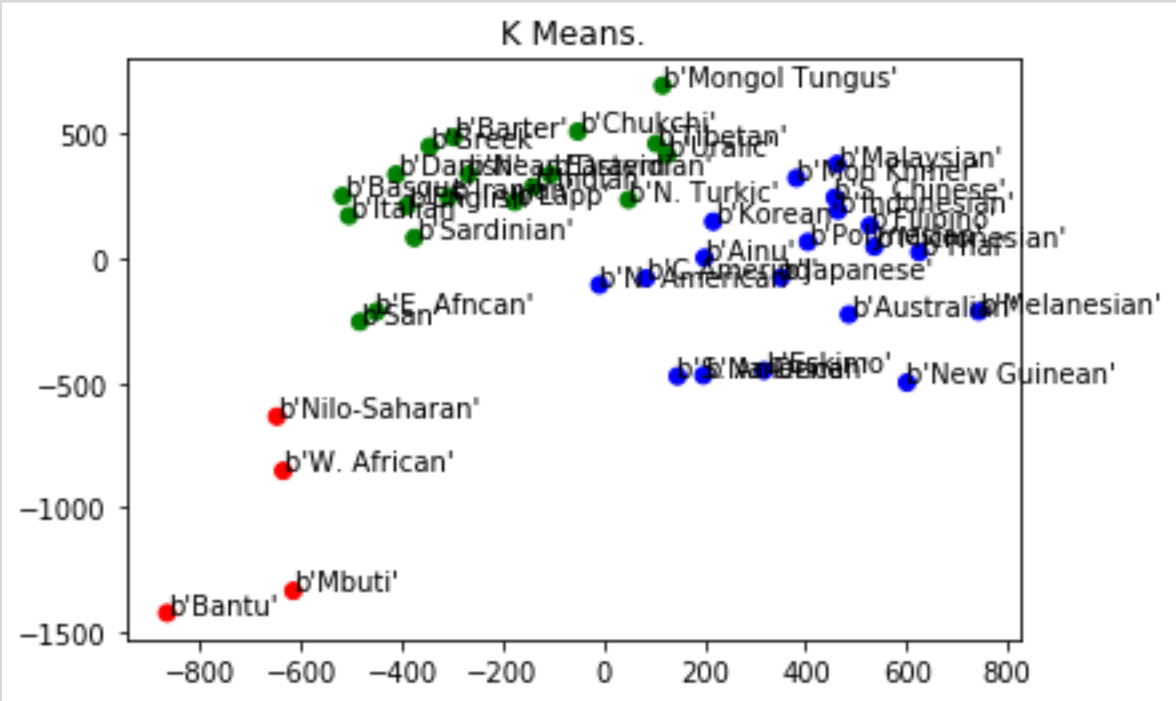
By sorting the singular value of D in descending order, and pick a number of top singular values that would explain the most of the total variance, such as the 95% of the variance. The number of singular values being picked is the desired dimension we want to reduce to.

4.(a) iii



4.(b)

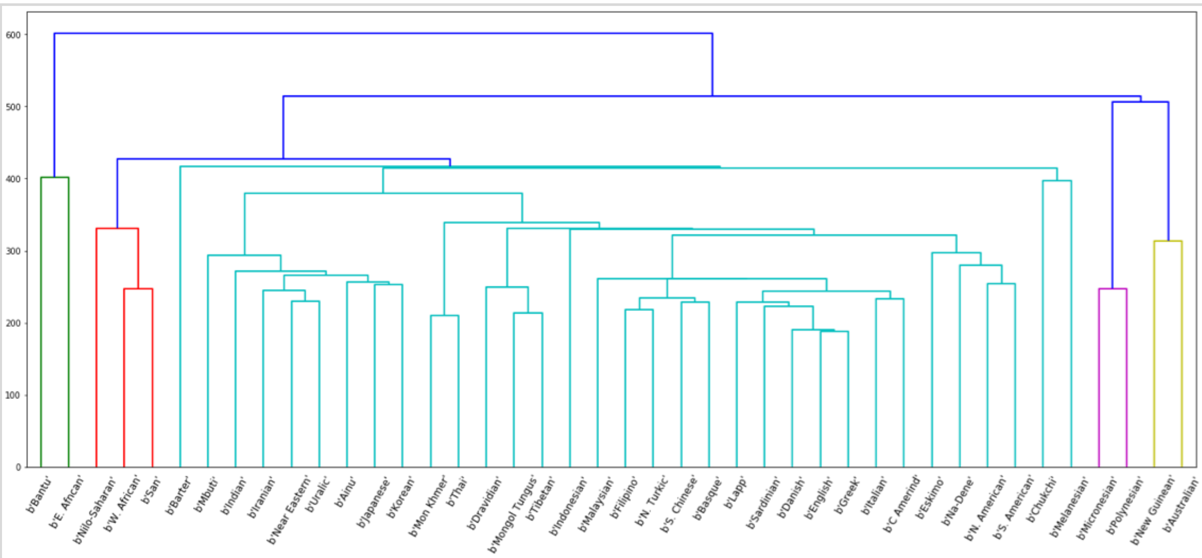
we choose K = 3.

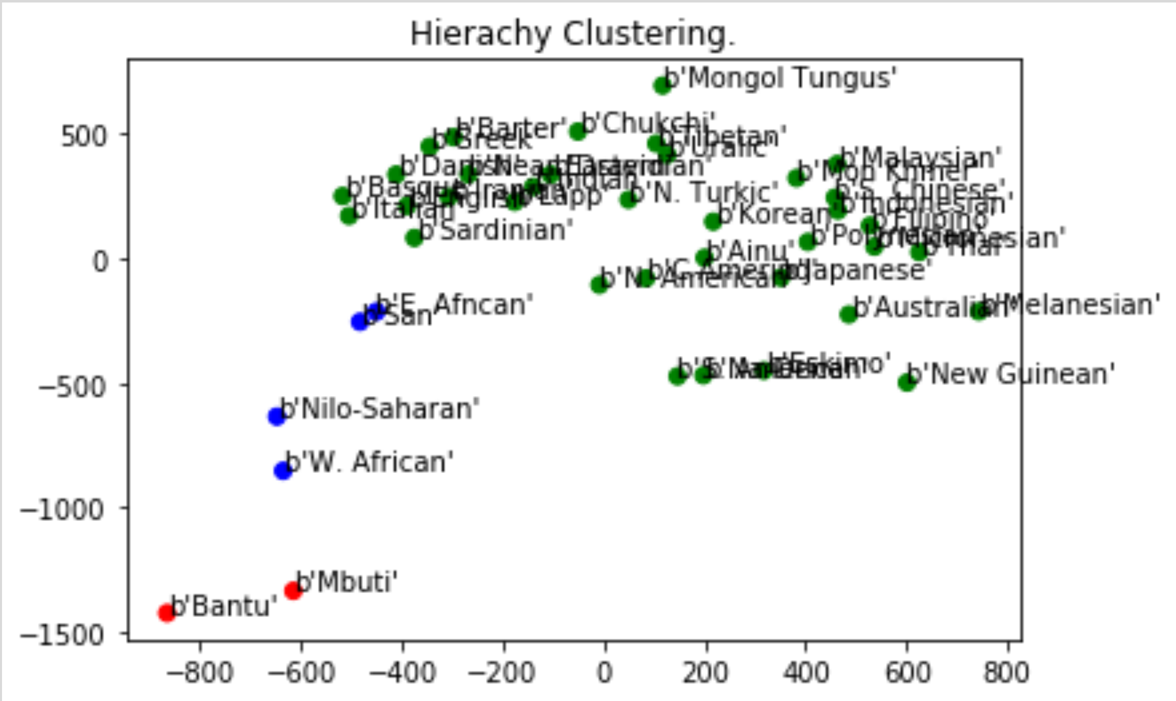


I do not agree with the resulting clustering. Some cultural, political information are lost during the process of MDS into 2 dimensions.

As we can also see from the original disimilarity map where Afncan and Saharan have a low dissimilarity score which means they are more similar than with other points. However, in the resulting clustering, it classifies them into two different groups. As K-means performs on comparing euclidean distances with hard threshold, it provides little explanations on the data points that lie on the edge of decision boundary.

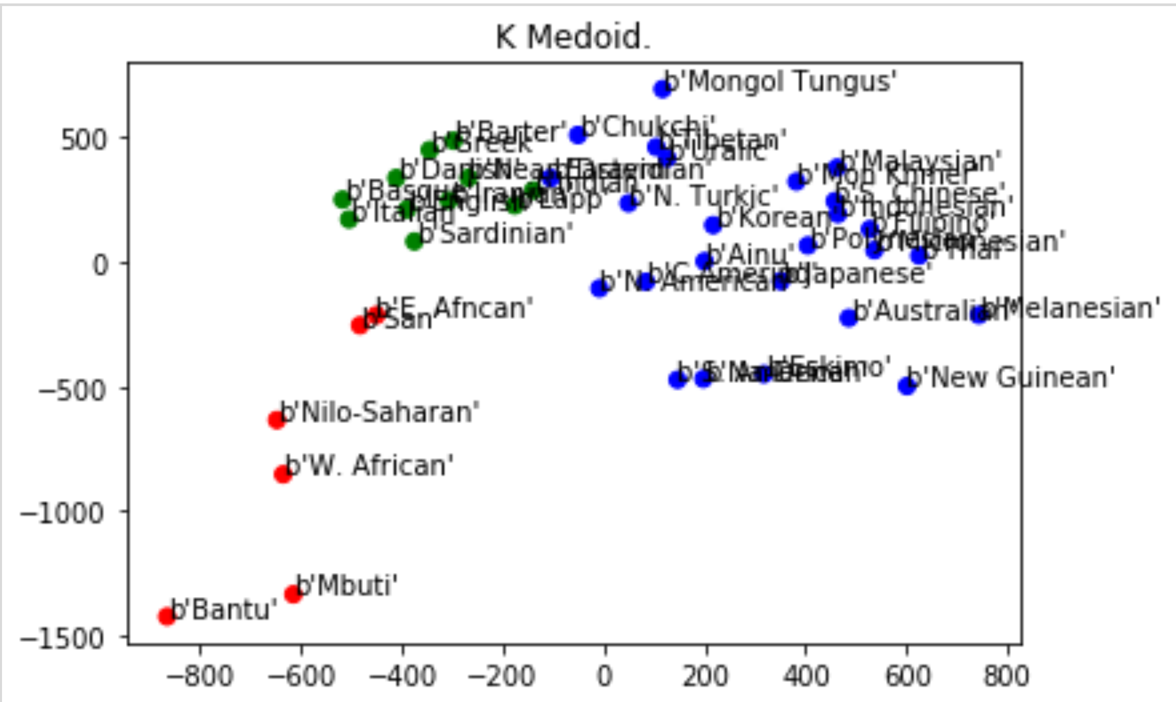
4.(c)





the clustering result I obtain using hierarchy clustering performs worse than the k-means algorithm.

4.(d)



From observation, there is no significant difference between the clusters generated by K-means and K-medoids.