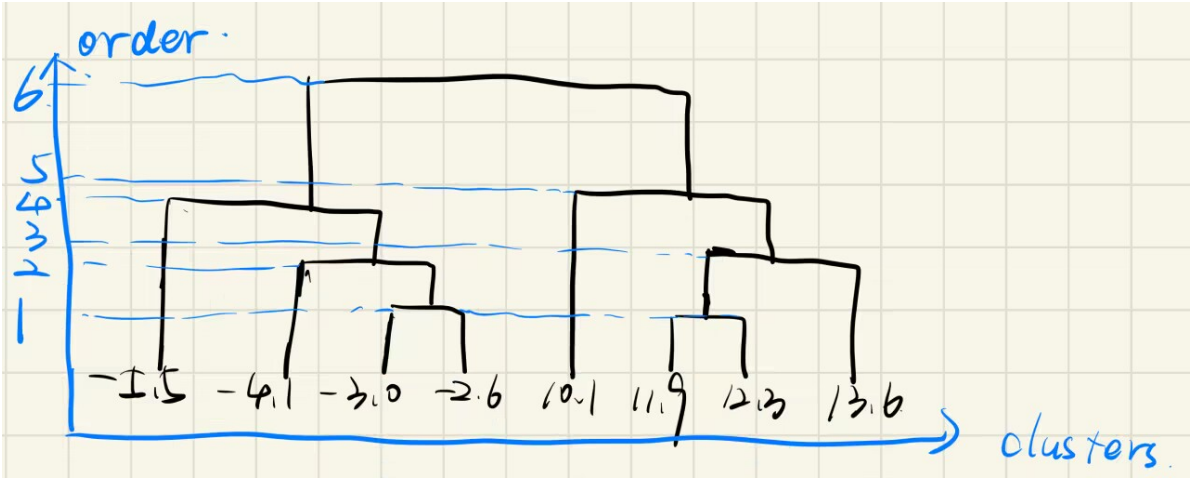
Problem 1

For the distance computation will be dirty if fully written in this file, herein we just plot the cluster tree here.

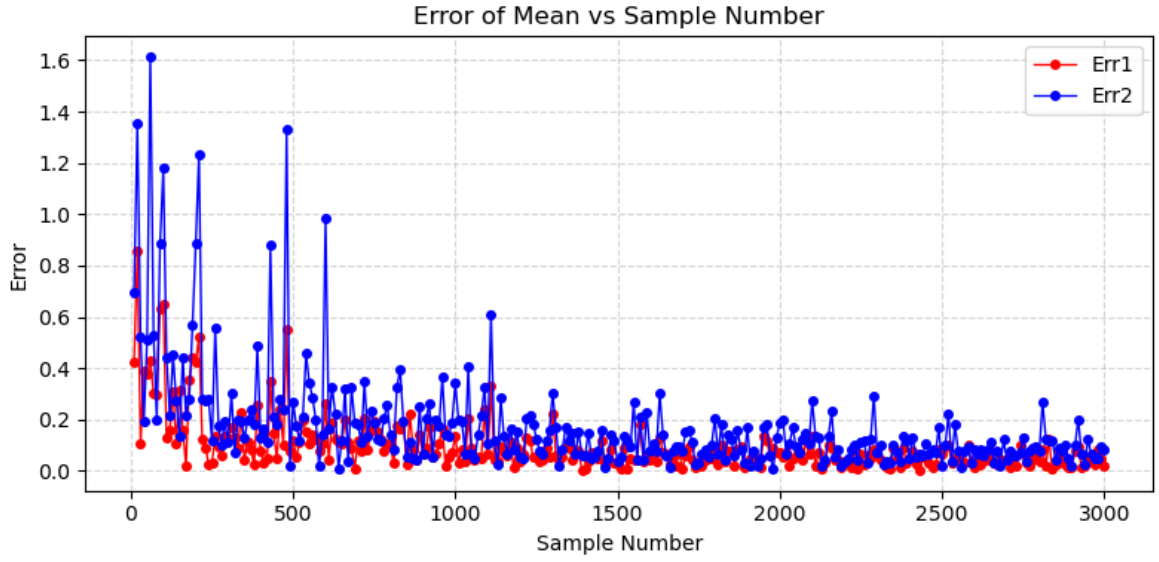


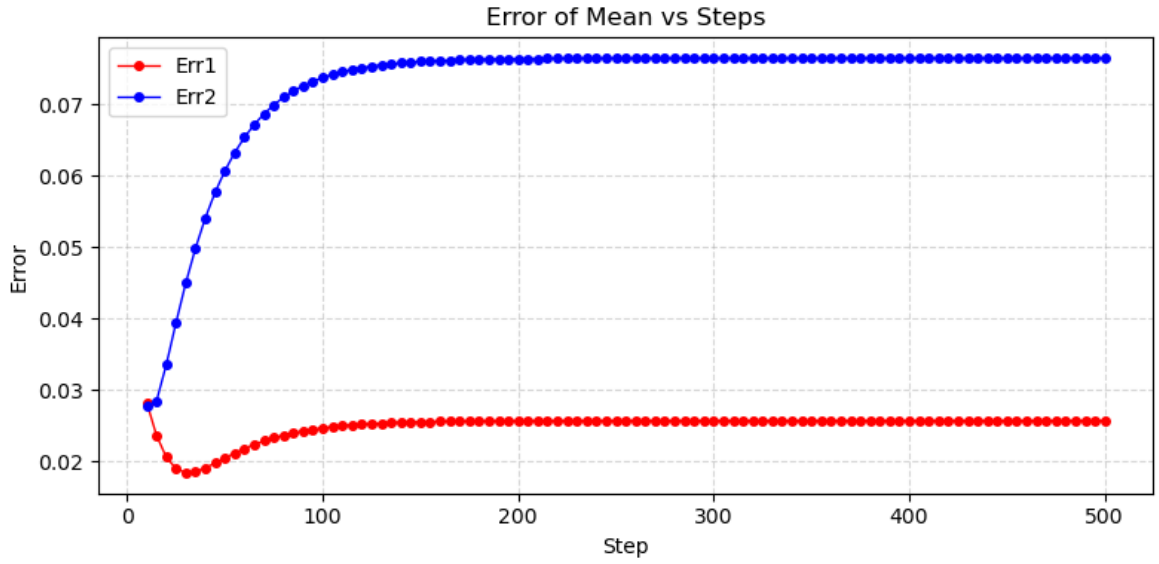
where the value of the horizontal axe is the index of the cluster, and the value of the vertical axe is the order of the option of clustering i.e. the order of the minimum distance between the component clusters. If we just align the $2, 3$ and the $4, 5$ clusters in the vertical direction, the value of the vertical axe can be the order of the cluster i.e. the number of the son clusters.

I tend to cluster them in 2 clusters, for the elements in both the positive and negative clusters are near enough, and the more kind would not bring more meaningful information.

Problem 2

The two diagram needed (Error vs. Sample Number and Error vs. Step) are plotted below





herein `Err1` and `Err2` represents the prediction error of μ_1 and μ_2. The parameters for initialization are listed below:

```
init pi = [0.5 0.5]
init mu = [[-1.  -1. ]
 [ 3.3  3.3]]
init cov = [[[1 0]
  [0 1]]

 [[1 0]
  [0 1]]]
```

and the obtained estimation of this case is that

```
estimated mu = [[-0.00320319 -0.00493657]
 [ 1.96494916  1.91083535]]
estimated cov = [[[ 0.93669309 -0.01668212]
  [-0.01668212  1.02665548]]

 [[ 1.05657667  0.06755868]
  [ 0.06755868  1.00266782]]]
estimated pi = [[0.65453571 0.34546429]]
```

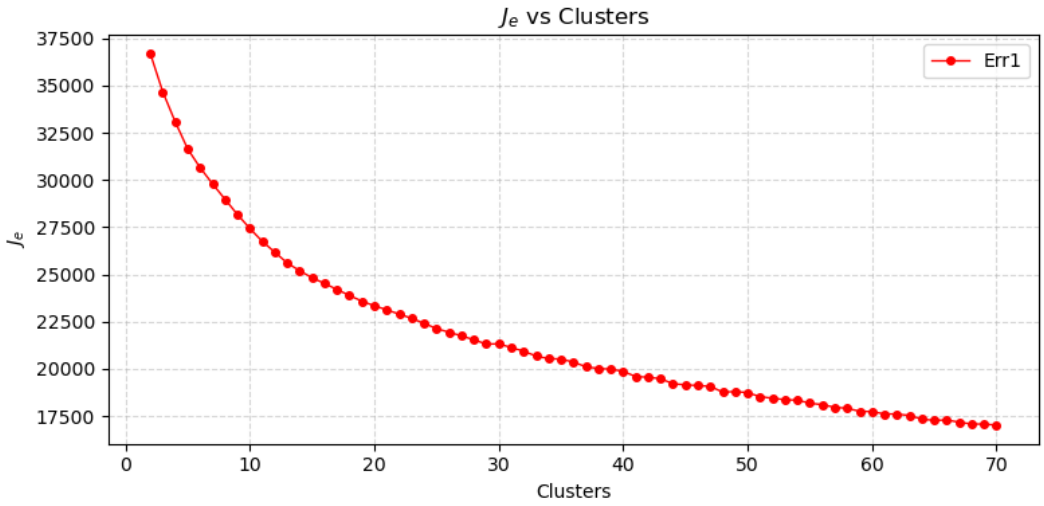
Herein we take the sample number as `N = 2000`.

It's notable that the converged value is quite sensitive to the initial parameters. By trying I've found that only when the initial $\mu_2 \in (3, 4)$, the estimated value of μ_2 can converge to the neighborhood of $(2, 2)$, otherwise $(1, 1)$ etc.

Problem 3

Part 1

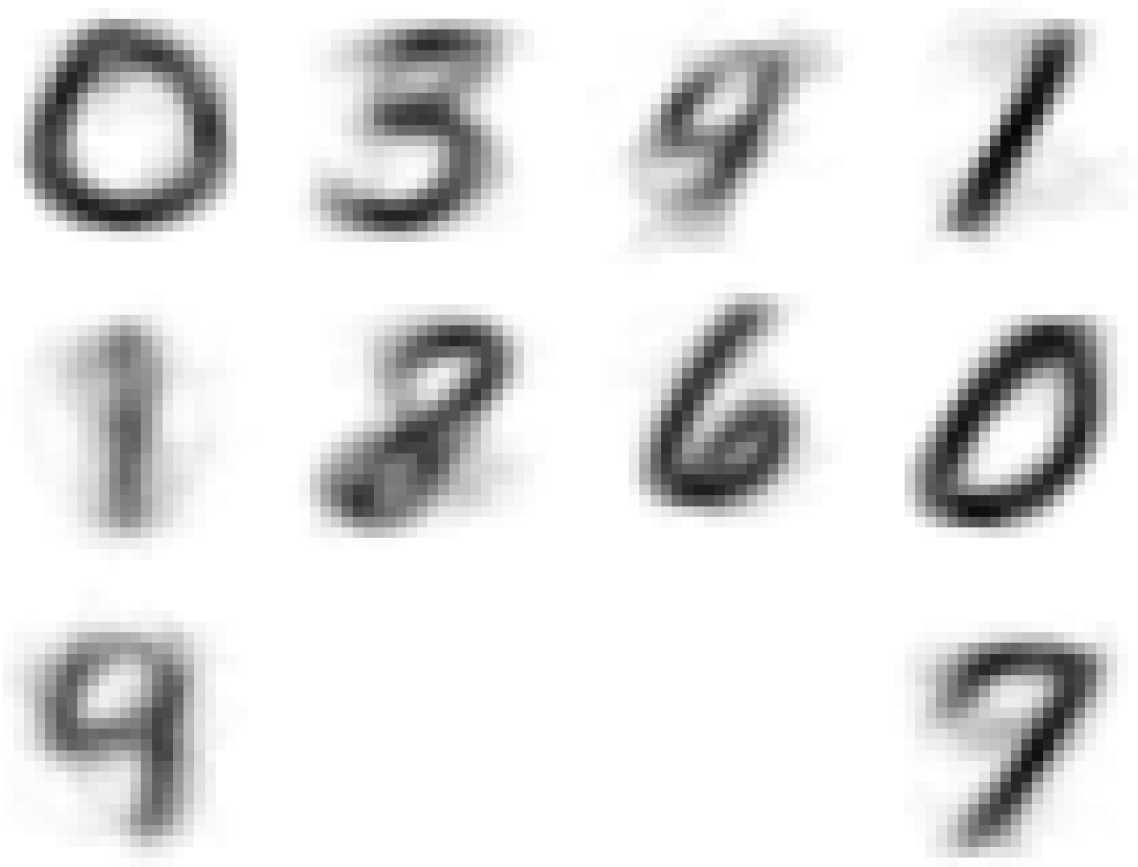
The diagram of J_e w.r.t. number of clusters are plotted below:



here we can see that the elbow point is just located in $[10, 15]$, so an arbitrary value within $[10, 15]$ can be considered as the elbow point. Since the one-hot label has shape `(n_samples, 10)`, we take the elbow point as $10 \in [10, 15]$ in the following parts.

Part 2

The learned means obtained by `KMeans` are plotted below:



They looks just like hand-written numbers. The prediction accuracy on `MNIST` test dataset is

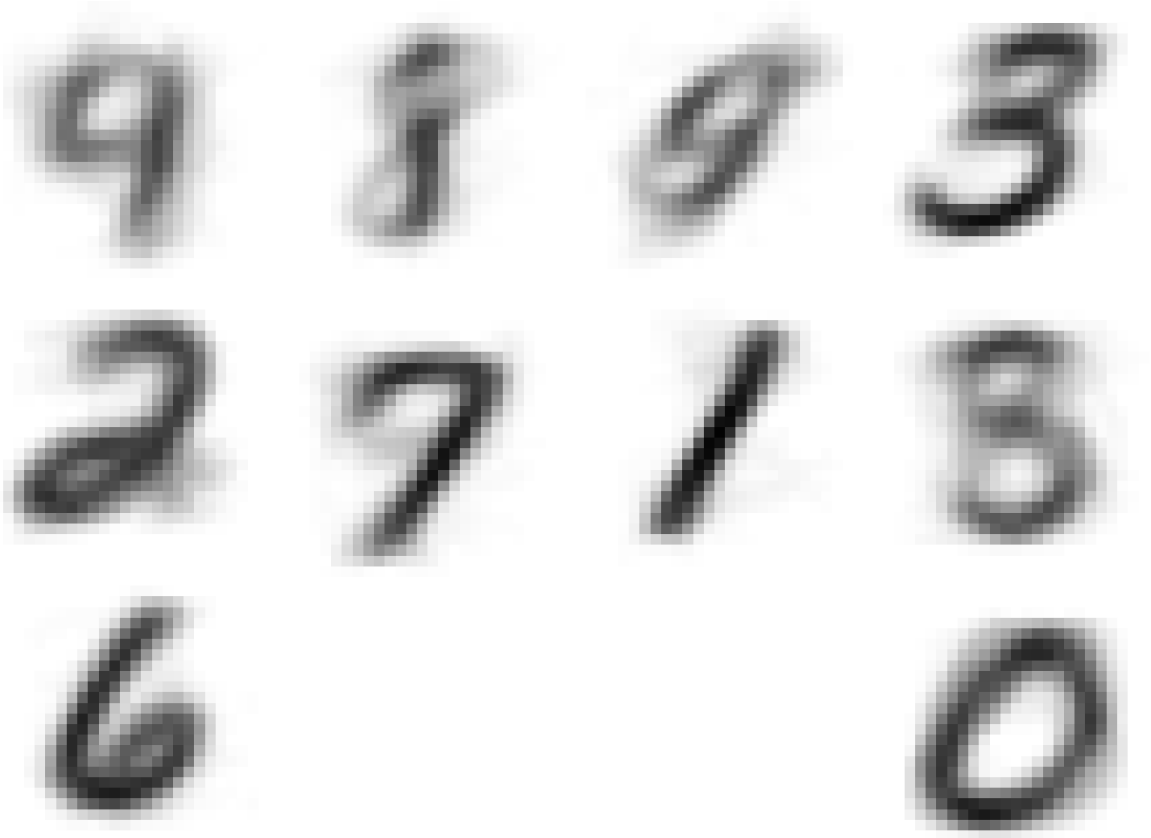
```
accuracy: 0.605
```

under parameters

```
model_kms = kms(
    n_clusters=clusters,
    init='k-means++',
    n_init=50,
    max_iter=600,
    tol=1e-4,
    random_state=None
)
```

Part 3

The means vector gained by GMM are shown below:



and the accuracy is

```
accuracy EM: 0.565
```

somewhat lower than that of `KMeans`.

Whereas, the means vector of `EM` seems better than of `KMeans`. Most the 10 numbers are clearly shown as the means vector - only `4` seems missing. While in the results of `KMeans`, we cannot even find `2`, moreover `4` and `5` becomes the same vector, with 2 `0`s and `1`s existing.