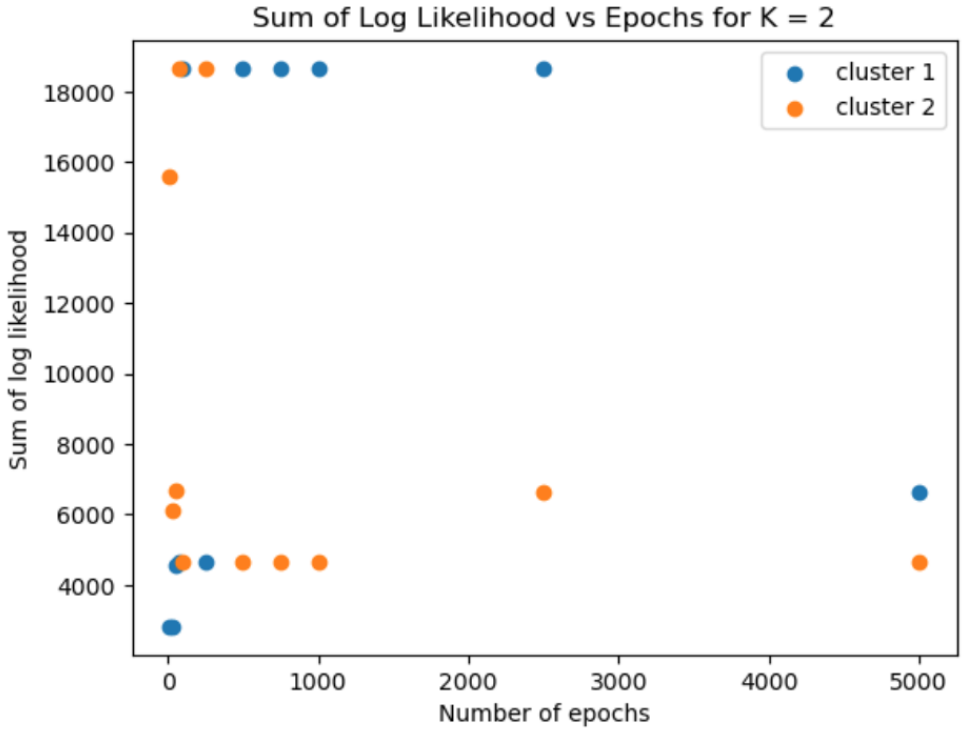
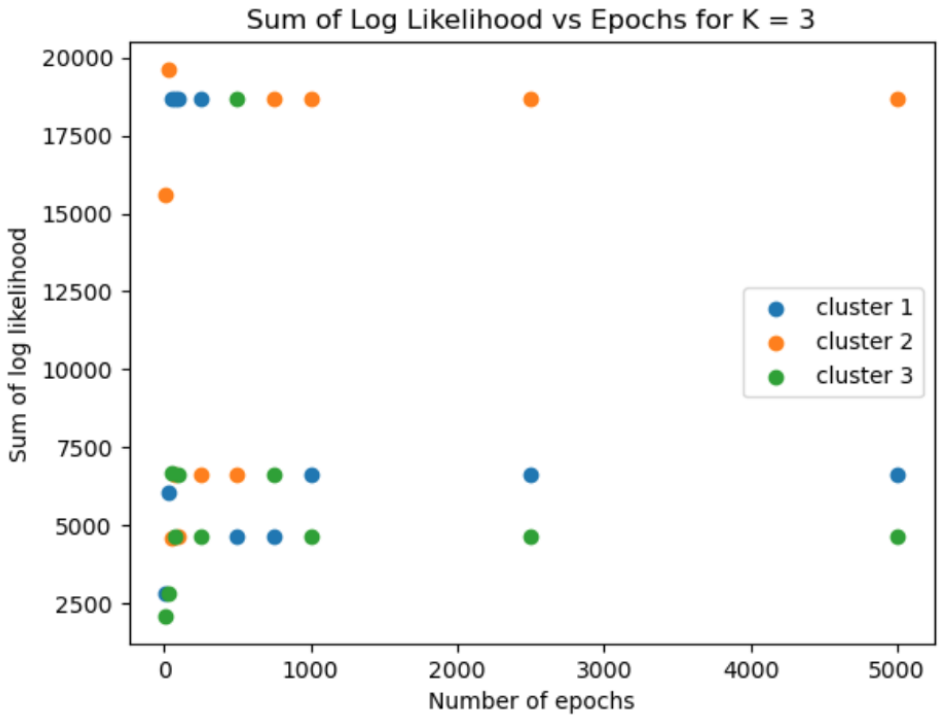
Review the Gaussian Mixture Model (GMM) clustering algorithm using expectation-maximization
For ODD T-number use K = 2 and for even T-number use K = 3 in general

1. For k number of clusters, plot the sum of log likelihoods vs epochs





2. Print the size of likelihood variable, posterior variable, prior variable, and totals variable

```
261     return history, likelihoods, totals, scores
262
263     #return labels, scores
```

(First, I modified the return variables in the fit_predict_gmm function)

```
In [2]:
#modified fit_predict_gmm to return history, likelihoods, totals variables

df = myPack.importData(n_feature =5, scale = True)
n_clusters = 3
n_epochs = 100

history, likelihoods, totals, scores = gmm.fit_predict_gmm(df.values, n_clusters, n_epochs)
```

```
In [15]:
print("size of posterior variable for a single cluster:", scores.size)
print("size of likelihood variable:", likelihoods.size)
print("size of totals variable", totals.size)
print("size of prior variable for a single cluster:", len(history))

size of posterior variable for a single cluster: 1656
size of likelihood variable: 100
size of totals variable 552
size of prior variable for a single cluster: 100
```

3. Fill in the table below

Sample ID	Totals	Cluster 1 score	Cluster 2 score	Cluster 3 score	Decision
1	1.52872790e-04	3.93802235e+01	1.96234397e-01	1.73519197e+00	1
2	8.63562854e-05	4.39375975e+01	6.59835547e-01	7.33544102e-01	1
3	6.77810067e-05	1.30775991e+01	2.24245975e-01	1.69372764e+00	1

4. Fill in the table below

Cluster/Parameters	Mean	Sigma	Importance (Wk)
Cluster 1	-0.63637735	2.15145862e-01	1
Cluster 2	1.36632125e-01	0.36733948	1
Cluster 3	0.6148613	0.10836499	1