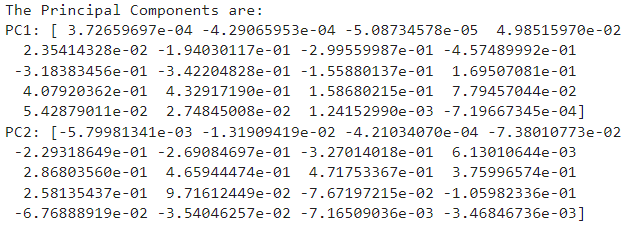
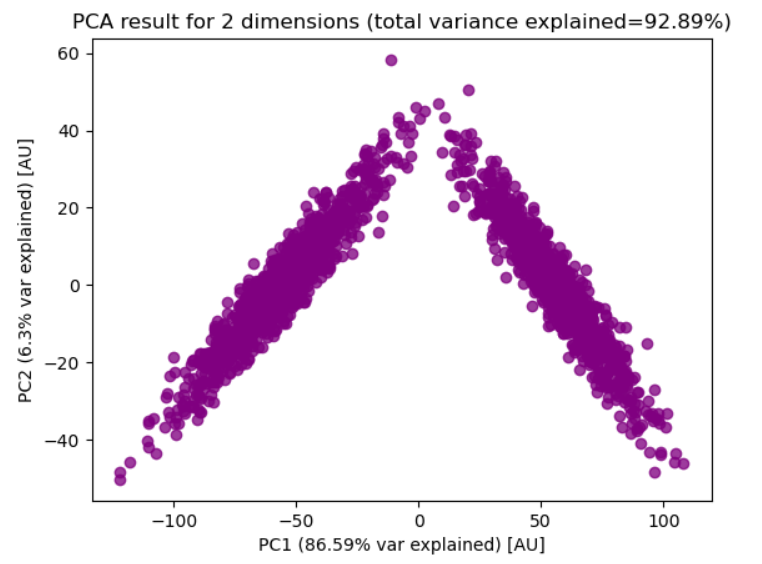
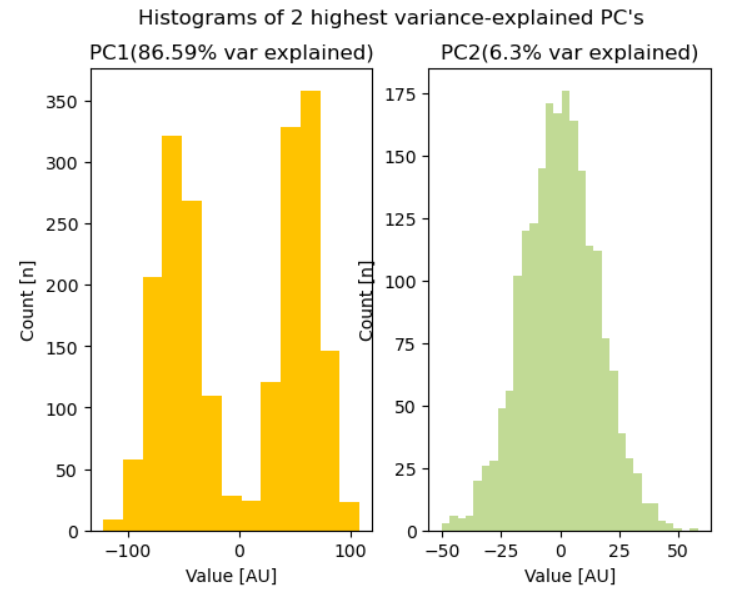
**Assignment #3**

Our calculations were made in Python, the code is attached both in a notebook and HTML files.

Question 1 – Dimensionality reduction using PCA

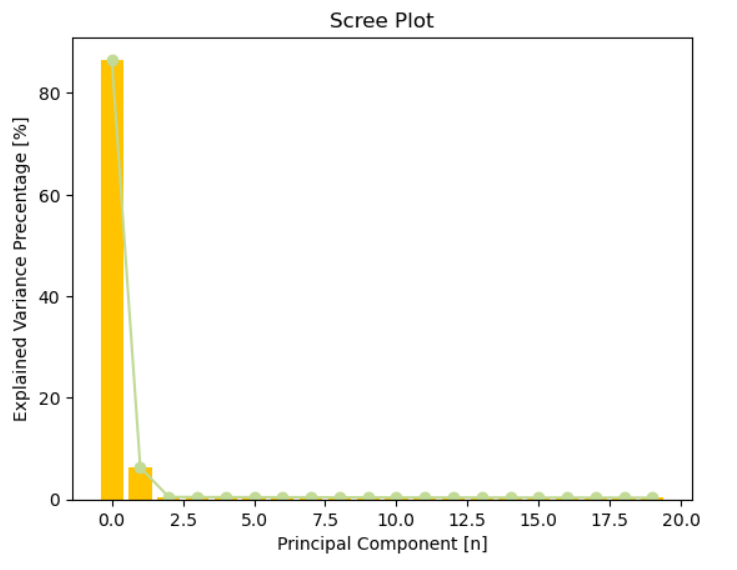
1. Using our own PCA algorithm, we've found the first 2 PCs as the eigenvectors of the covariance matrix with the highest eigenvalues. 
2. We've projected the centered data to the PCs by multiplying it by the feature vector (the vector containing the chosen PCs).



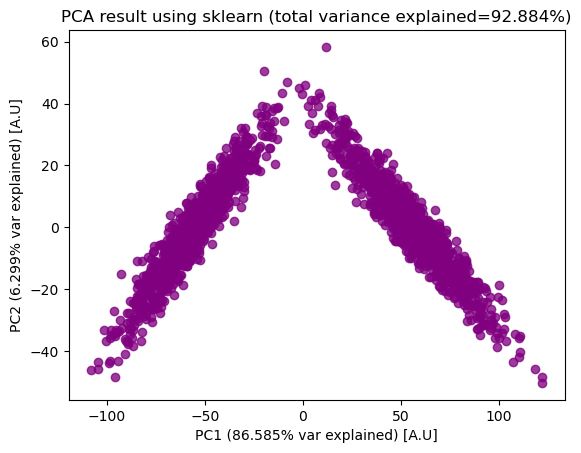


We can infer that PC1, which explains most of the variance, splits the LFP data into two distinct activation patterns – hence the bimodal distribution. In PC2, which explains considerably less of the data, we can't find any meaningful divide in terms of LFP.

1. As shown on the plots from (c), the percentage of variation explained by PC1 is 86.59% and 6.3% for PC2.
2. We certainly can use both PCs to understand this LFP data, as the two of them explain 92.89% of the variance. However, as we discussed in section (c), we can use only PC1 to understand the data, as it both explains most of the variance and splits it in a meaningful manner.
3. We've used the PCA function from the *sklearn* package.   
   (i)

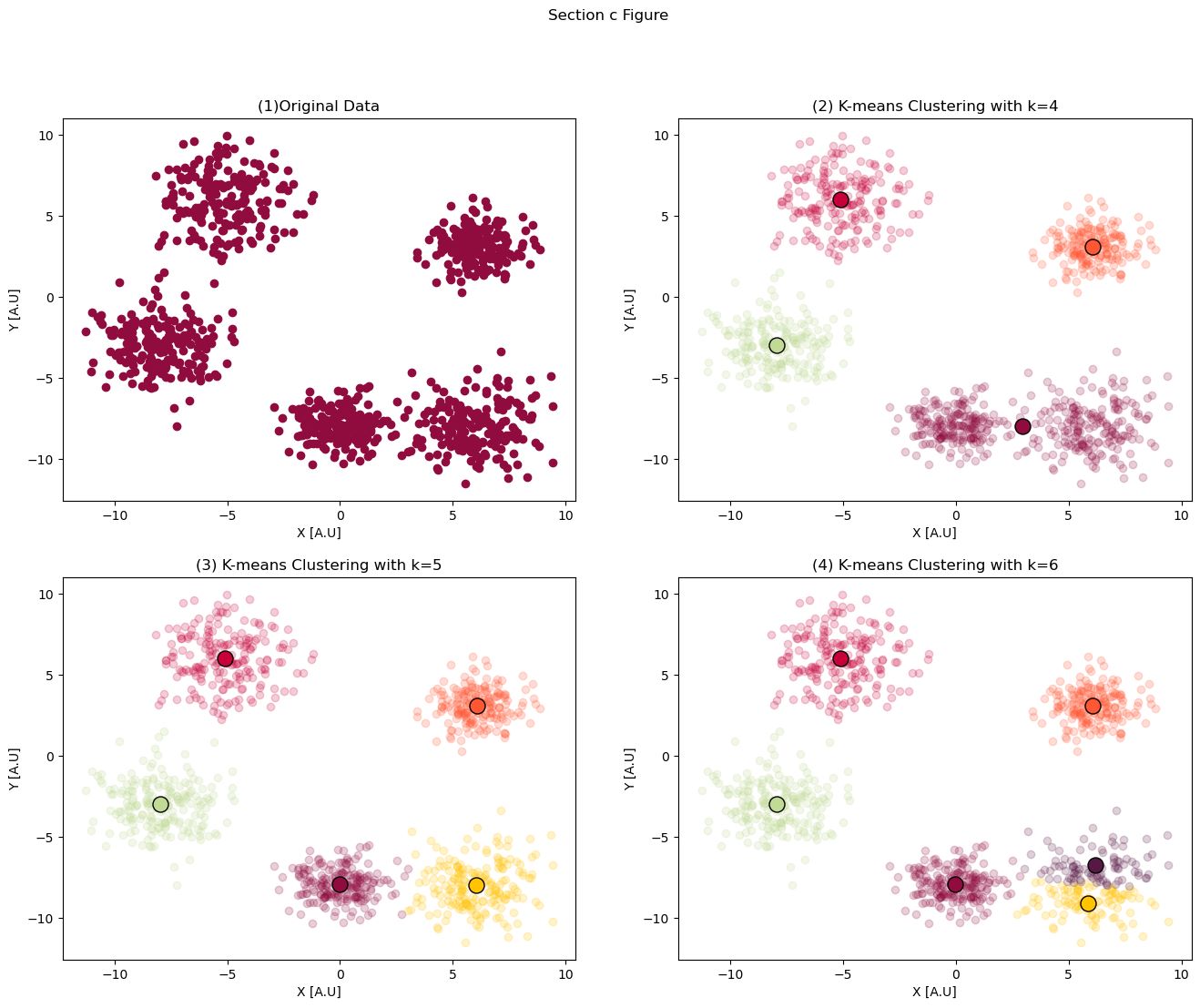
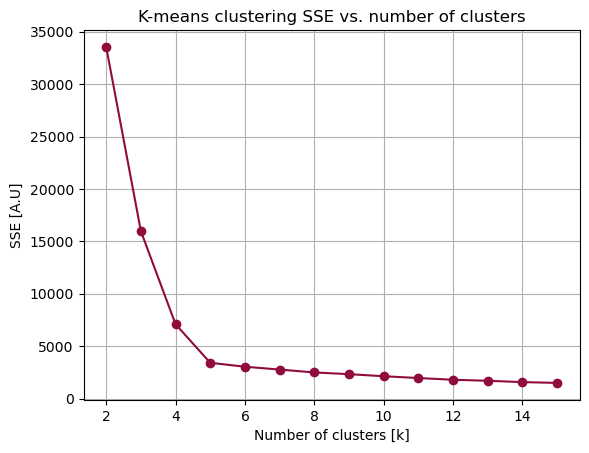


(ii)



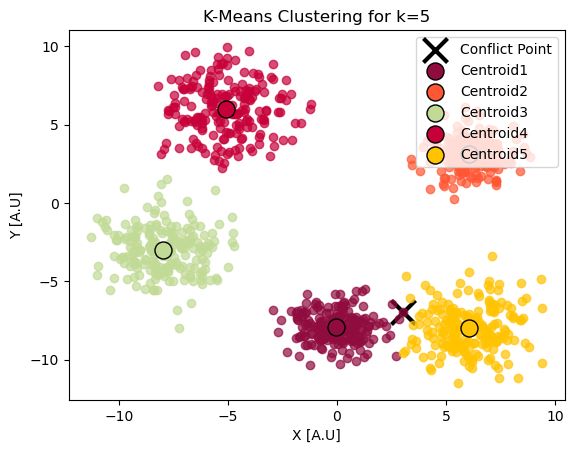
1.   
   As we can see, both methods of calculation result in the **same** features. This is to be expected and can be explained by the fact that PC1, which is comprised of a linear combination of all the features, applies the **largest weights** to the features with the **highest variation**, as it is the PC that explains most of the variation.

Question 2 – Clustering using K-Means

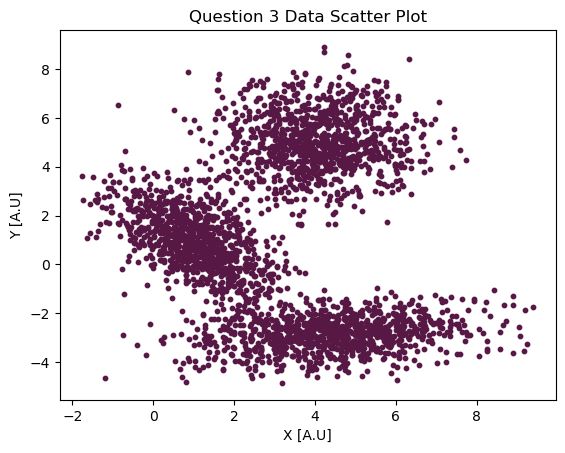
1. Code for our K-Means function is attached  
   
2. By looking at the original data, we've chosen the following points, as they are close to our assumed clusters: 
3.   
   Visually, it makes sense to split the data into **5** clusters. As can be seen, once we've added 6 clusters the split becomes arbitrary.
4. 

To find the optimal number of clusters, we've plotted the SSE (the square distances between every point and its’ cluster) resulting from splitting the data for 2-15 clusters. Using the *elbow method,* we can see where the drop-off becomes shallow, meaning that adding more clusters doesn't contribute meaningfully. This point is **k=5**, confirming our visual assessment.

1. K-means, being a *hard-clustering* algorithm, empirically assigns each point to a cluster based on the shortest distance. Therefore, the only points of conflict would be those that have **equal** distances from two or more clusters. In our case, there was no such point, so instead we found a point (denoted by 'X' in the following figure) that was almost as close to one cluster as to the other, but it’s assigned to cluster1 as it is closer to it.

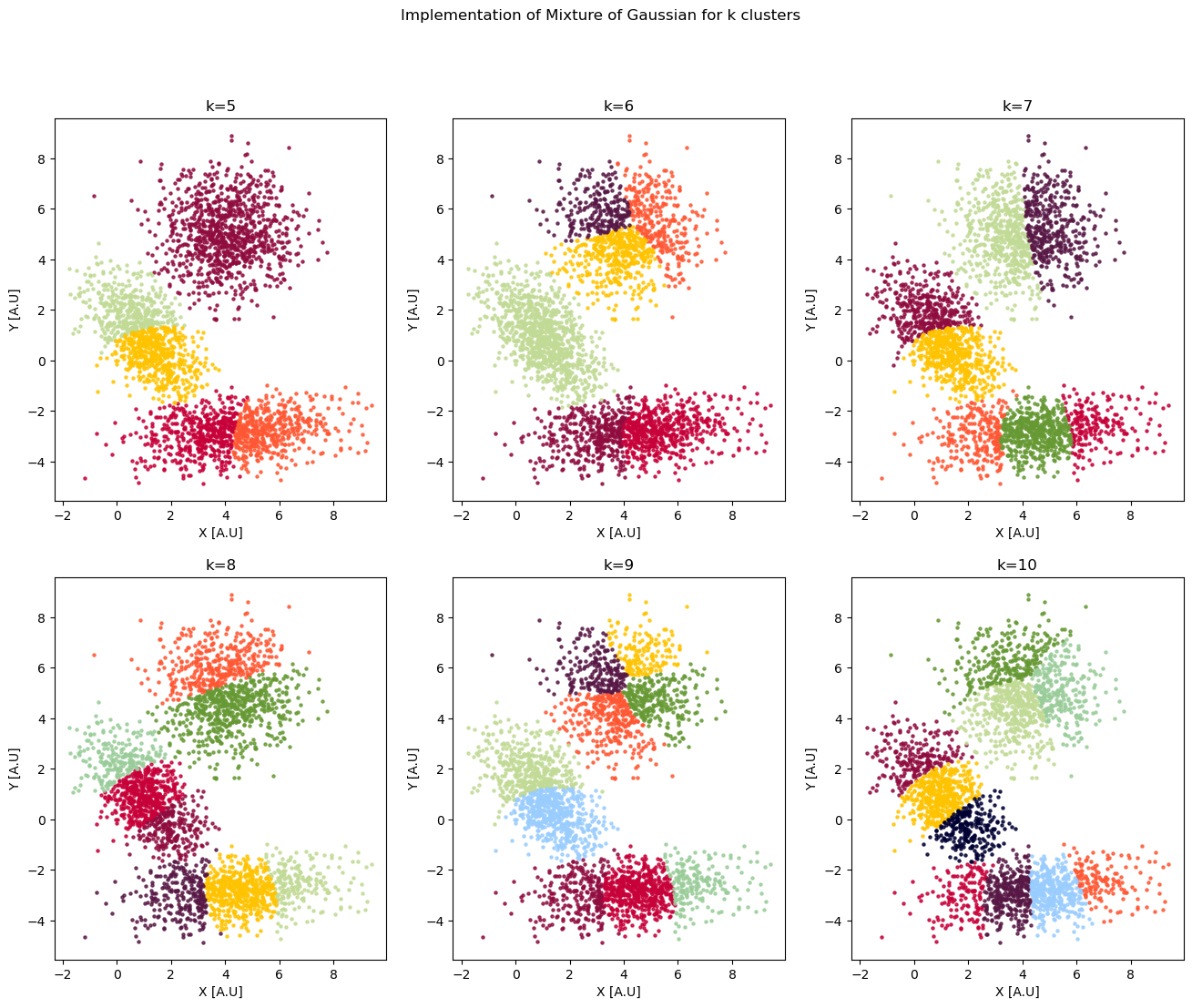
  


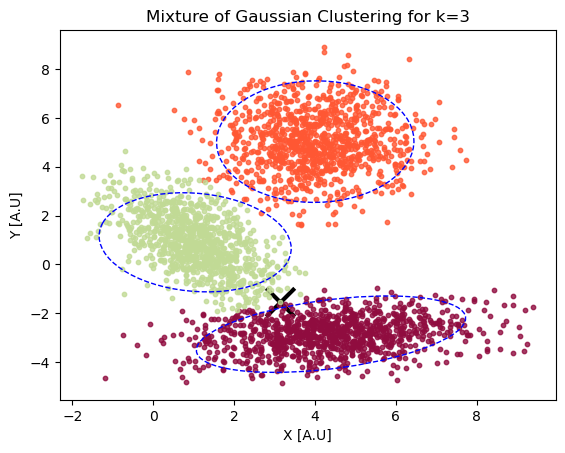
Question 3 – Clustering using Expectation Maximization

* 1. 

Judging by what is seen in the figure, our intuition for k is **k=3**, as there seems to be three distinct groups.

* 1. The built-in function that we've used is also from the *sklearn* package, and the number of clusters tested were 5 to 10. We presented them in the following figure:





תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

* 1. To select a data point with uncertainty regarding its appropriate cluster, we've looked for a point that has around a 50% probability of belonging to one cluster. In a mixture of Gaussians model, the probability of a point belonging to each cluster is given by the posterior probability of the point given the cluster. This probability can be computed using Bayes' theorem. We found point 2887 (denoted by 'X' in the above figure) as such and derived the probability of the point belonging to each cluster with a built-in function



In the end, the point is assigned to the green cluster which has a slightly larger probability.

* 1. Visualization becomes difficult or impossible as the dimensionality of the data increases. Therefore, it may not be possible to use visual inspection of the data to estimate the number of clusters. One method for estimating the number of clusters in high-dimensional data is to use a dimensionality reduction technique, such as PCA, to project the data onto a lower-dimensional space. This can make it easier to visualize the data and identify the number of clusters. Another method is to use an information criterion, such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). These criteria are based on the likelihood of the data given the model and the model with the lowest AIC or BIC score is preferred.