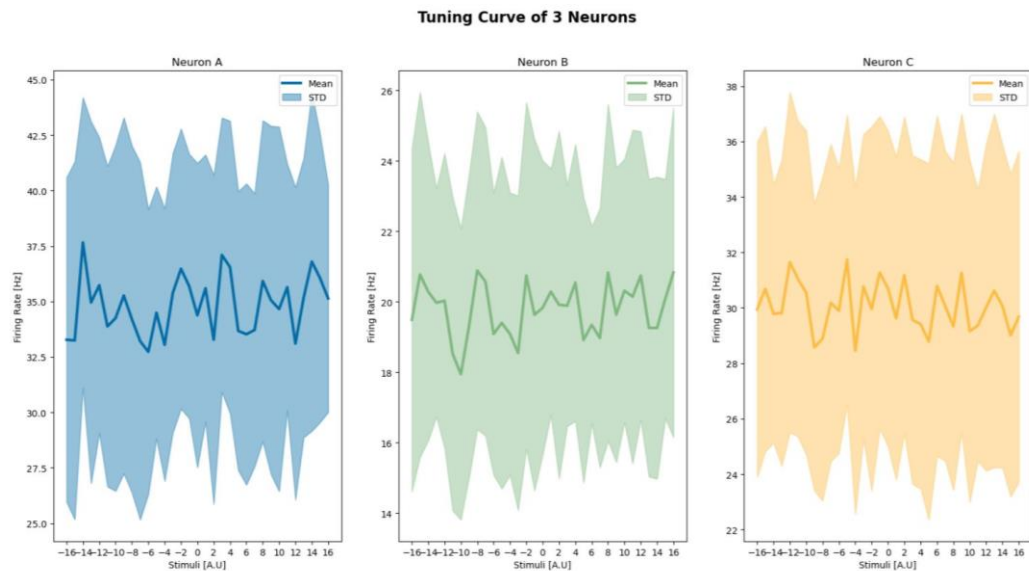


Assignment #5

Our calculations were made in Python, the code is attached in a notebook file.

Question 1

a)



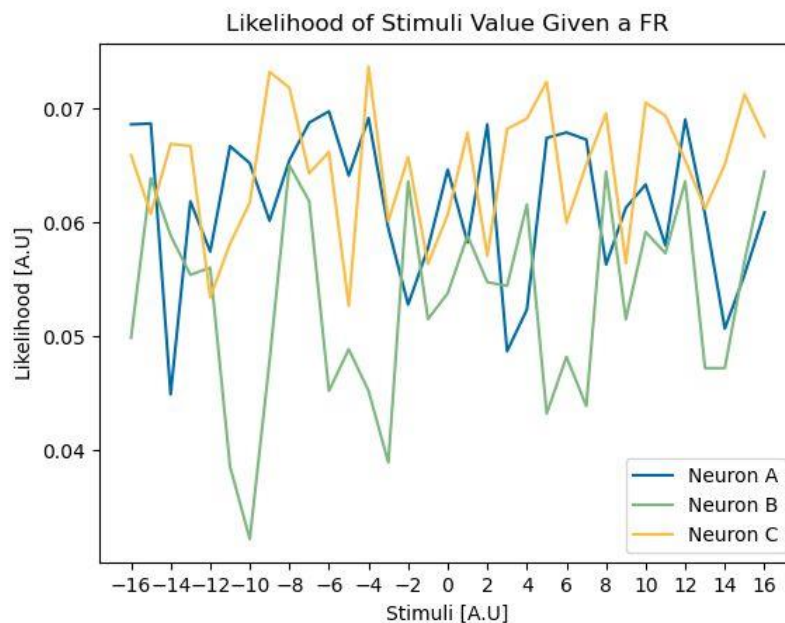
b)

Mean firing rate of each neuron:

```
{'Neuron A': 34.81293054922771, 'Neuron B': 19.78874458874459, 'Neuron C': 30.055312882447506}
```

As can be seen, each neuron's firing rate mean is aligned with the mean of the distribution it was sampled from, confirming that the samples were taken randomly.

c)



d)

The MLEs for the 3 neurons are as follows:

Neuron A: The MLE is: 0.0698 and the most likely stimulus value is: [-6]

Neuron B: The MLE is: 0.065 and the most likely stimulus value is: [-8]

Neuron C: The MLE is: 0.0737 and the most likely stimulus value is: [-4]

e)

According to the population's maximum log likelihood, the most likely stimulus value is: -8

The population's log likelihoods for each stimuli value are:

[-8.396, -8.229, -8.64, -8.383, -8.667, -8.806, -8.95, -8.463, -8.092, -8.203, -8.474, -8.708, -8.375, -8.879, -8.417, -8.693, -8.463, -8.365, -8.447, -8.617, -8.408, -8.464, -8.535, -8.557, -8.283, -8.632, -8.238, -8.375, -8.154, -8.647, -8.766, -8.404, -8.235]

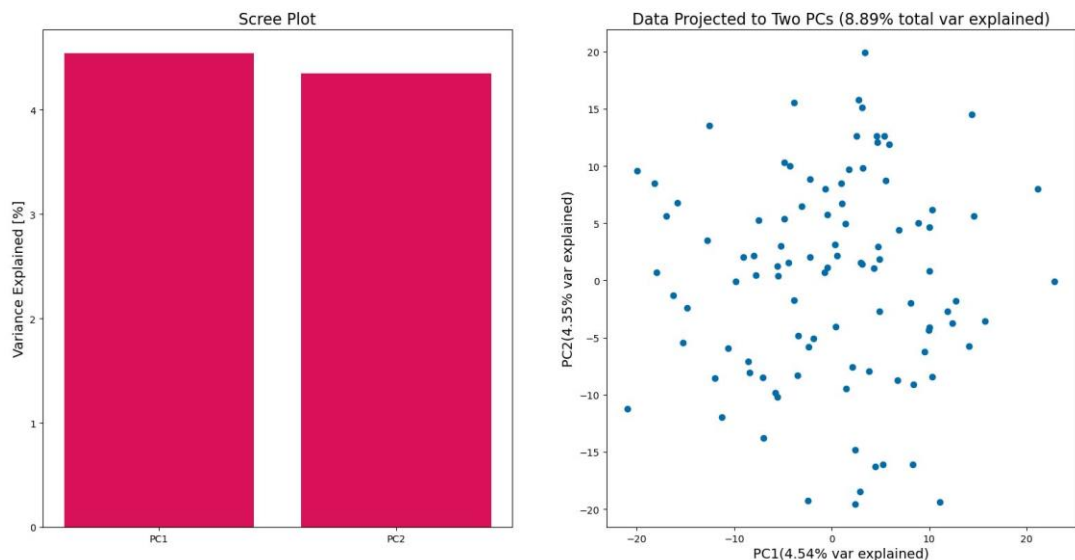
Yes, the answer has changed: now we have only one MLE (of the population) vs. three different MLE's for each neuron. In this case, the stimulus value which maximizes the population's log-likelihood is the same as one of the neurons (B), but as the data is randomly simulated no inference can be made from it.

When we have a large dataset, using the log likelihood function involves taking the natural logarithm of the likelihood function, which can make the calculations more efficient. Furthermore, it is a monotonically increasing function of the likelihood, meaning that maximizing the log likelihood is equivalent to maximizing the likelihood itself.

f) It wouldn't change the answer because in the previous answer we've implemented a ML decoder, which does not take the prior into consideration. But were we to use a MAP decoder or any other posterior-based decoder the prior would be used in the decoding process.

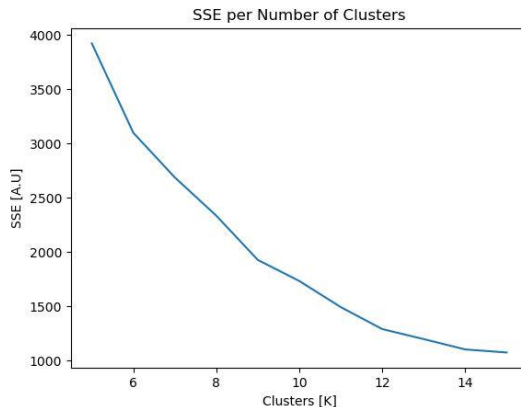
Question 2

- a) We have simulated the EEG recordings of 100 subjects, so that for each electrode we've simulated an average firing rate measured by that electrode (in accordance with the instructions), as can be seen in the code.
- b)

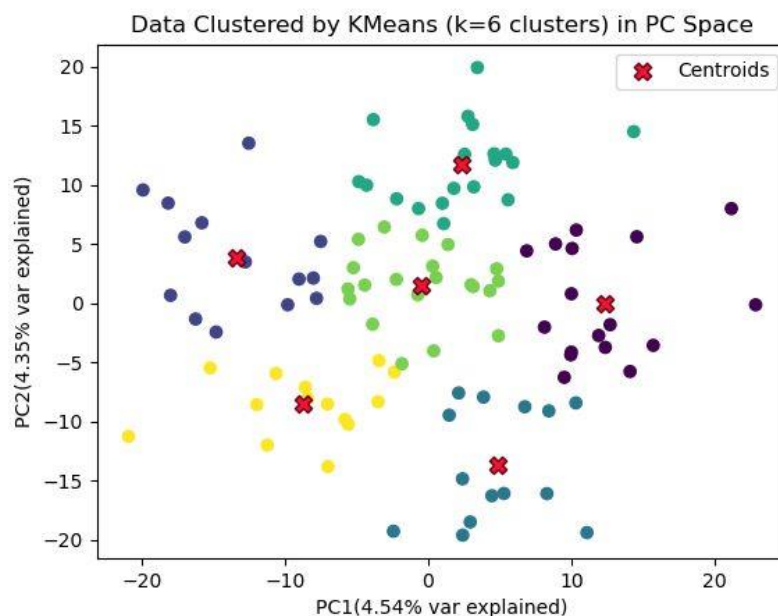


Due to the low total percent of explained variance by PC1 and PC2 (PC1 – 4.54% var, PC2 – 4.35% var explained – 8.89% total), we would not recommend analyzing this data using just two dimensions as most of the information is lost.

- c) To choose the optimal number of clusters, we've plotted the following figure in order to use the *elbow method*:



Unfortunately, we can't easily distinguish an "elbow". So, we've decided to apply a more empirical method: we've calculated the second derivative of the above function to find where the rate of change of the slope was the highest. From this analysis we've determined that the optimal amount of clusters for this data is 6.



Theoretically, were the data not randomly simulated, the different clusters could represent different patterns of activity across areas of the brain (measured by different electrodes) from which we could distinguish different phenotypes (i.e., healthy vs. AD patients).

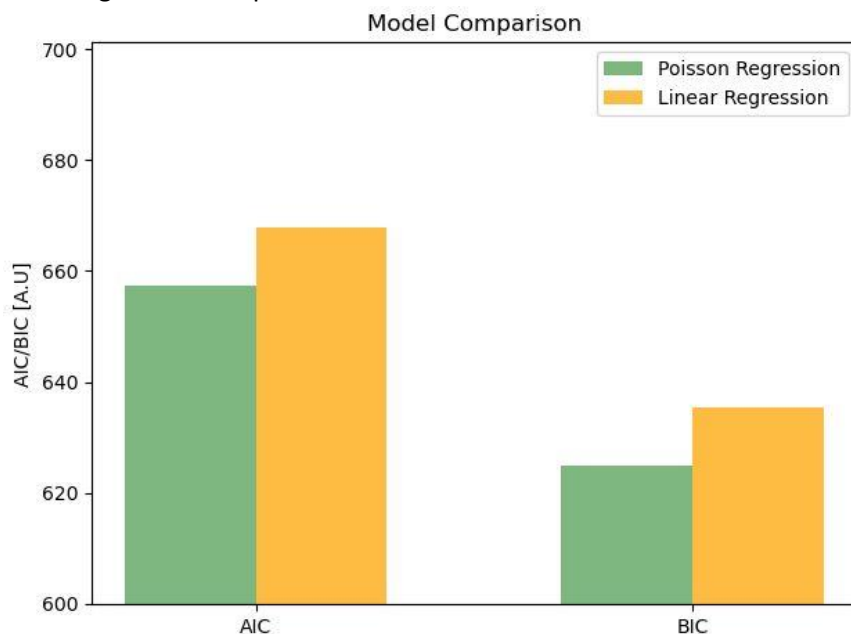
The main caveats of K-means – it's a hard clustering method meaning each point must be assigned to a specific cluster, not allowing for ambiguity. Also, it is dependent on starting conditions, meaning a different amount of clusters and different starting points for the centroids might end up giving different results. The first caveat can be resolved by using a soft clustering algorithm which assigns each point probabilistically to each cluster. As for the second, it can be addressed by running the algorithm multiple times with different starting conditions and seeing which result fits best.

d) The 5 electrodes with the heighest PC1 loading scores (from lowest to heighest) are: [54 8 64 30 48]
 =====
 The Theta vector for the Poisson regression is: The Theta vector the linear regression is:

Intercept	2.269959	Intercept	8.308432
E54	0.006173	E54	0.095724
E8	0.009198	E8	0.143154
E64	0.000177	E64	0.003877
E30	0.014725	E30	0.226723

The theta values, excluding θ_0 which represents the intercept, represent how much each of the four electrodes help predict the fifth – theoretically meaning that a positive theta value corresponds to an electrode measuring from a brain area that has an excitatory effect on the brain area from which the predicted electrode records. The higher the absolute value the stronger the influence.

e) In this figure we compare the two models with *AIC* and *BIC*:



as can be seen in the figure above, in both criteria the **linear regression** is better.

The simulated value for Arthur's E48 electrode is: 10.9 Hz,
 while the predicted value is: 15.964 Hz

The two values differ, which means the model's prediction is not perfect, but it is also not that bad as the difference is less than 1 STD (5.75Hz).