Today we're going to work with single and double Gaussian data from Viets, *et. al* (2019). You have been provided with data on blackboard, as well as a skeletal script.

1. We want to explore these data, to do this we will first plot this data in a variety of ways using seaborn.
   a. Produce a swarm plot of the data
   b. Produce a violin plot of the data
   c. Produce a box plot of the data
   d. Produce a bar plot of the data

Some of the data look like they may come from a double Gaussian, rather than single Gaussian distribution. We wish to determine which model is more likely. To do this, we will calculate the Bayesian Information Criterion for each, and choose the model with the LOWEST BIC.

2. For each sample, calculate $\ln \mathcal{L}(\mu, \sigma; x)$ for a single Gaussian distribution.

   With the maximum likelihood estimation, we try to find the peak value of the log-likelihood function:

$$\ln \mathcal{L}(\theta; x) = \sum_{i=1}^{n} \ln p_\theta(x_i) \qquad \text{Eq. 1}$$

   Where, for a Gaussian,

$$p_\theta(x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \qquad \text{Eq. 2}$$

   The most probable value for μ is where:

$$\frac{\partial \ln \mathcal{L}(\mu; x)}{\partial \mu} = 0 \qquad \text{Eq. 3}$$

   If you solve for μ, you should get:

$$\mu = \frac{1}{n} \sum x_i \qquad \text{Eq. 4}$$

   i.e., the very definition of a mean. If you do this for σ, you'll derive the definition of a standard deviation.

Calculate $\ln \mathcal{L}(\mu, \sigma; x)$ for the Gaussian, using `numpy.nanmean` and `nump.nanstd` to calculate μ and σ.

3. Next we will need to calculate the most probable parameters (μ_1, σ_1, μ_2, σ_2, w) for a double Gaussian, where:

$$p_\theta(x_i) = \frac{w}{\sqrt{2\pi\sigma_1{}^2}} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1{}^2}} + \frac{(1-w)}{\sqrt{2\pi\sigma_2{}^2}} e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2{}^2}} \qquad \text{Eq. 5}$$

In this case, there is NOT an analytical solution to $\frac{\partial \ln \mathcal{L}(\theta;x)}{\partial \theta} = 0$ for either $\theta = \mu$ or σ. In order to calculate the most probable parameters, we'll need to solve for the five parameters numerically, using `scipy.optimize.minimize`.

For these methods, you need to provide the algorithm with a "guess" for what you think the parameters might be. In the code I provided, I chose means that were one standard deviation below or above the total mean, the standard deviations for a single gaussian, and a w = 0.5.

The other IMPORTANT detail is that you want to MAXIMIZE $\ln \mathcal{L}(\theta;x)$, but you are using a MINIMIZATION function, therefore you want to MINIMIZE the NEGATIVE $\ln \mathcal{L}(\theta;x)$.

Another important caveat is that these algorithms don't know the practical boundaries of your model, so I imposed that in my function for the double Gaussian, e.g. means can't be less than zero, w can't be greater than 1, etc.

Side Note: You will notice that I chose the 'Nelder-Mead' minimization function. I did this because it is better behaved due a steep Jacobian issue I encountered when using Jacobian-based algorithms. If you want to know what that means, we can talk later.

4. After you calculate the most probable parameters, recalculate $\ln \mathcal{L}(\mu,\sigma;x)$ for the double Gaussian, and compare that to the $\ln \mathcal{L}(\mu,\sigma;x)$ you calculated for the single Gaussian. In principle, ALL the $\ln \mathcal{L}(\mu,\sigma;x)$ for the double Gaussians should be more likely (i.e. should be larger). Remember, $\ln \mathcal{L}(\mu,\sigma;x)$ is always a negative number, so the larger $\ln \mathcal{L}(\mu,\sigma;x)$ is the one that is closer to zero.

5. As we learned earlier, the double Gaussian should always produce a higher likelihood because the model uses 5 vs the 2 parameters in the single Gaussian model. To compensate for this bias, calculate the BIC for both models:

$$BIC = k \ln n - 2 \ln(\hat{\mathcal{L}}) \qquad \text{Eq. 5}$$

Where k = parameter #, n = sample size (100 in this case), and

$$\hat{\mathcal{L}} = \mathcal{L}(\theta;x)_{max} \qquad \text{Eq. 6}$$

The model with the LOWEST BIC is the more LIKELY model.