EE104 Homework 9

1. **Imputing missing entries in data.** In `impute.json`, you will find two datasets, comprising a train and a test set each. The first dataset is a 750×2 matrix `U1_train` and a 750×2-vector `U1_test` consisting of raw training and test data. (There is no output data.) is a 750×5 matrix `U2_train` and a 750×5-vector `U2_test` consisting of raw training and test data.

   We will work with $x = \phi(u) = u$. In this exercise, you will fit data models and impute missing entries in data, determining how many clusters there are in the data by validating how well each data model does at imputing data.

   Using the following implausibility functions, fit the data models using the training data. For each of the data models, give the model parameter θ. Submit your code used to generate these models.

   To validate each data model, you will remove an element from each test data record at random, impute the entry (substitute the missing with values from the following models) according to the data model, and compute the average RMS imputation error for each model. In addition, for each of the data models, plot the training data, in blue, and the test data, in red, on a scatter plot. (There should be one scatter plot for each implausibility function.)

   a) For part (a), we will only use the dataset `U1_train` and `U1_test`.

      For each of the 750 samples, hide one of the variables by replacing it with `missing`, a Julia keyword.

      Hint. In Julia, `missing` signifies missing data. https://docs.julialang.org/en/v1/manual/missing/ has some information on how this datatype propagates. The functions `ismissing(x)` and `skipmissing(x)` are quite useful.

      Hint. One way of creating an array that can accept both `missing` and `Float64` variables is `X_redacted = Array{Union{Missing,Float64}}(missing, size(X))`

      Hint. Read the problem statement for part (b) before solving part (a). You should write your code in such a way that it can handle any number of input features.

      Next, implement imputation methods based on the following implausibilty functions.

      i. Sum squares implausibility function: $\ell_\theta(x) = \|x - \theta\|_2^2$.
      ii. Sum absolute implausibility function: $\ell_\theta(x) = \|x - \theta\|_1$.
      iii. k-means implausibility function, with $k = 5$: $\ell_\theta(x) = \min_{j=1,\ldots,5} \|x - \theta_j\|_2^2$.
      iv. k-means implausibility function, with $k = 10$: $\ell_\theta(x) = \min_{j=1,\ldots,10} \|x - \theta_j\|_2^2$.
      v. k-means implausibility function, with $k = 15$: $\ell_\theta(x) = \min_{j=1,\ldots,15} \|x - \theta_j\|_2^2$.
      vi. k-means implausibility function, with $k = 20$: $\ell_\theta(x) = \min_{j=1,\ldots,20} \|x - \theta_j\|_2^2$.

      Hint. For methods (i) and (ii), you can use `Statistics` functions to compute θ.

      Hint. For the k-means implausibility function, you can import `Clustering` and use `Clustering.kmeans`[1]. The `kmeans(X, K)` function takes a matrix `X` where each **column** is a data point and `K`, the number of clusters.

[1] https://juliastats.org/Clustering.jl/stable/kmeans.html

Evaluate each method by computing the average RMS imputation error for each model. In addition, for each of the data models, plot the training data, in blue, and the test data, in red, on a scatter plot. (There should be one scatter plot for each implausibility function.)

Based on your results, give a guess for how many clusters are in the data. Provide a short justification for your answer.

b) To validate that your implementation is capable of handling dimensions greater than just two, apply the exact same "hiding" function and subsequent imputation functions to the dataset `U2_train` and `U2_test`. Because it has 5 dimensional features, we will make do with only plotting the first two dimensions. Again, or each of the data models, plot the training data, in blue, and the test data, in red, on a scatter plot. (There should be one scatter plot for each implausibility function.) Also, report the average RMS imputation error for each model.

   Based on your results, give a guess for how many clusters are in the data. Provide a short justification for your answer.