**NOTE:** These practices are intended as a reinforcement of the concepts shown in this morning lesson. Therefore, you should avoid the use of external libraries (*Of course, there exist more efficient implementations of the algorithms that we are working on*), do not worry for efficiency and focus on the concepts.

1. **VARIABLE RANKING BY MUTUAL INFORMATION**
    a. Download the Congressional Voting Records Data Set http://archive.ics.uci.edu/ml/machine-learning-databases/voting-records/house-votes-84.data

    b. Program the Mutual Information between two discrete classifications with the programming language of your choice (If you know it, I suggest you to use *awk*).
        i. You must compute for each possibly value ($l = \{\mathrm{y}, \mathrm{n}, ?\}$) of the feature (columns 2 to 17 of the dataset), its probability ($p(l)$). Do the same for each possibly value ($j = \{\mathrm{republican}, \mathrm{democrat}\}$) of the ground truth classification (first column of the dataset) and compute also the joint probability of both ($p(l, j)$). Then apply the formula:

$$MI(k, G) = \sum_{l=1}^{k} \sum_{j=1}^{G} p(l, j) \log \frac{p(l, j)}{p(l)p(j)}$$

    c. Rank the utility of the features to reproduce the ground truth classification according with the mutual information criterion.
2. **K-MEANS**
    a. Download the S3 dataset from http://cs.uef.fi/sipu/datasets/s3.txt
    b. Program (again use the programming language of your choice) a naïve implementation of Lloyd's algorithm for k-means and apply it to this dataset (k=15).
        i. **Input:** dataset and number of clusters

ii. Use Euclidean distance

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

iii. Randomly initialize $k$ elements as the centers of the $k$ clusters

iv. Assign the elements to the same cluster as their nearest center.

v. Compute the new centers as the average positions of all the elements of the cluster.

vi. Repeat step iv and check the assignation, if it is the same as the previous one, we are at convergence: stop.

vii. **Output:** Objective function ($O(z) = \sum_{l=1}^{k} \sum_{i=1}^{n} \delta_{z_i l} ||\vec{x_i} - \vec{c_l}||^2$) and cluster assignation.

3. If you have time:
   a. Apply the algorithm with $k$=15 for 100 times, obtain the best value of the objective function and the average one. Plot the assignation for the best case.
   b. Perform the scree plot (logarithm of the objective function as function of $k$ with $k$ ranging from 2 to 20) for this dataset.