# 实验要求

1. 自行实现K-Means算法和性能指标评估函数（如SSE,SC（Silhouette Coefficient）, CH（Calinski-Harabasz Index）等指标）
2. 对所给消费者数据集 `Mall_Customers.csv` 采用K-Means进行聚类分析(完成聚类、模型评估、分析结果等)
3. 分析K-Means适用性（可不做）
    - 生成特定分布的数据，观察在这些分布上的效果（不限于所给数据集，自行设计或寻找不同分布的数据集，比如环形分布等）
    - 对于给定分布数据，不同的初始簇心和K值对结果的影响
    - 不同数据规模下算法计算代价
    - 其他

各数据集来源：

1. abalone
2. concrete_data
3. housing
4. Mall_Customers

# K-Means介绍

K-Means是一种常见的聚类算法，主要用于将数据分成K个不同的簇，在这些簇中的数据点具有相似性。

- 数学原理

假设我们有一组数据集$X = \{x_1, x_2, ..., x_m\}$，其中每个数据点$x_i$都有n个维度，表示为$x_i = (x_i^1, x_i^2, ..., x_i^n)$。现在我们要将这些数据点分成K个簇$C = \{C_1, C_2, ..., C_K\}$，即K个集合，其中每个集合都包含若干个数据点。
为了计算两个点之间的距离，常用的度量方式是欧几里得距离，即：

$$distance(x_i, x_j) = sqrt\left((x_i^1 - x_j^1)^2 + (x_i^2 - x_j^2)^2 + ... + (x_i^n - x_j^n)^2\right)$$

簇中心$\mu_k$表示簇$C_k$中所有数据点的均值，即：

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

K-Means的目标是最小化每个数据点与其所属簇中心之间的距离的平方，即使得$W(C)$最小，其中$W(C)$表示所有簇内距离的总和，即：

$$W(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} distance(x_i, \mu_k)^2$$

K-Means算法中，以上公式中的$W(C)$成为"成本函数"、"失真度函数"或"平方误差SSE(Sum of Squared Errors)"，我们需要最小化这个成本函数。可以通过迭代的方式最小化成本函数，即不断的重新计算簇中心和数据点所属簇的过程，直到成本函数不再变化或达到一定的迭代次数为止。

- 算法流程

1. 随机选择k个簇心。
2. 对于每个数据点，计算它到每个簇心的距离，并将该数据点分配到距离最近的簇心所属的簇。
3. 对于每个簇，重新计算该簇的簇心。
4. 重复步骤2-3，直到簇心的位置不再改变(或达到某个提前停止条件，以降低计算代价)。