Отчет по предмету Интеллектуальный анализ данных

## Задание 1

*Калганов Максим*

### Алгоритм Partition

#### Описание алгоритма:

Алгоритмами Partition называется множество алгоритмов, задачей которых является кластеризация. Основным алгоритмом Partition является алгоритм k-средних. Остальные алгоритмы являются его расширением.

**Математическое описание алгоритма**:

*Дано*:  
- набор из n наблюдений   
- k - требуемое число кластеров, .

*Требуется*:  
Разделить множество наблюдений на кластеров :

*Действие алгоритма*:  
Алгоритм k-means разбивает набор на наборов , таким образом, чтобы минимизировать сумму квадратов расстояний от каждой точки кластера до его центра (центр масс кластера). Введем обозначение, .  
Тогда действие алгоритма k-means равносильно поиску:

$$argmin\_S \sum\_{i=1}^k \sum\_{{x} \in {S\_i}} ρ(x,μ\_i)^2, \space (1)$$

где – центры кластеров, , – функция расстояния между и

*Шаги алгоритма*:

1. Начальный шаг: инициализация кластеров  
   Выбирается произвольное множество точек , рассматриваемых как начальные центры кластеров: ,
2. Распределение векторов по кластерам  
   Шаг
3. Пересчет центров кластеров  
   Шаг
4. Проверка условия останова: Если тогда

* ;
* перейти на шаг 2; иначе остановится

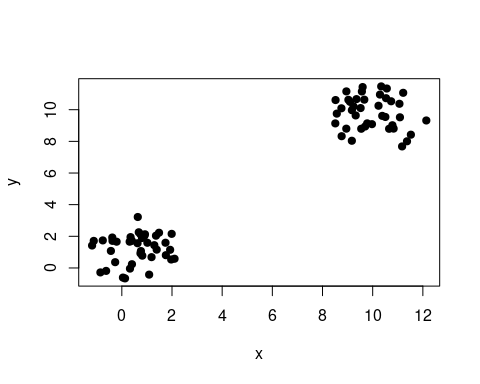
#### Реализация:

##### **Функционал**

library(ggplot2)  
  
clusterize <- function(matr, clusters){  
 diff\_c1 <- sweep(matr[, 1:2], 2, clusters[1])   
 diff\_c2 <- sweep(matr[, 1:2], 2, clusters[2])   
   
 dist\_c1 <- rowSums(diff\_c1 ^ 2)  
 dist\_c2 <- rowSums(diff\_c2 ^ 2)  
   
 matr[, 3] <- ifelse(dist\_c1 < dist\_c2, 2, 3)  
 return(matr)  
}  
  
get\_new\_clusters <- function(matr){  
 cluster1 <- subset(matr, matr[, 3] == 2)  
 cluster2 <- subset(matr, matr[, 3] == 3)  
   
 new\_cluster1 <- colMeans(cluster1)[1:2]  
 new\_cluster2 <- colMeans(cluster2)[1:2]  
   
 return(rbind(new\_cluster1, new\_cluster2))  
}  
  
init\_clusters <- function(matr){  
 centroids\_ids = sample(1:80, 2)  
   
 c1 <- data\_matrix[centroids\_ids[1], 1:2]  
 c2 <- data\_matrix[centroids\_ids[2], 1:2]  
 return(rbind(c1, c2))  
}

##### **Данные**

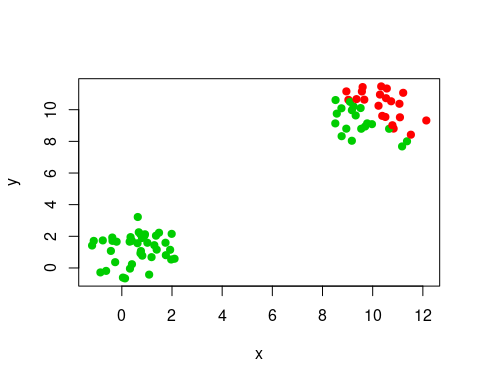
set.seed(10)  
data\_matrix <- cbind(c(rnorm(40, mean = 1, sd = 1),  
 rnorm(40, mean = 10, sd = 1)),  
 c(rnorm(40, mean = 1, sd = 1),  
 rnorm(40, mean = 10, sd = 1)),  
 rep(1, 80))  
colnames(data\_matrix) <- c("x", "y", "col")  
  
plot(data\_matrix, col=data\_matrix[, 3], pch=19)



##### **Шаги алгоритма**

Первое приближение:

clusters <- init\_clusters(data\_matrix)  
data\_matrix <- clusterize(data\_matrix, clusters)  
new\_clusters <- get\_new\_clusters(data\_matrix)  
plot(data\_matrix, col=data\_matrix[, 3], pch=19)



После сходимости метода:

while (any(clusters != new\_clusters)){  
 clusters = new\_clusters  
 data\_matrix <- clusterize(data\_matrix, clusters)  
 new\_clusters <- get\_new\_clusters(data\_matrix)  
}  
  
plot(data\_matrix, col=data\_matrix[, 3], pch=19)

