Введение в иерархический кластерный анализ: матрица расстояний, метод агломерации, дендрограмма

Алла Тамбовцева

Реализуем в R иерархический кластерный анализ, который мы проделали на лекции вручную. У нас есть пять наблюдений и две переменные, X и Y. Запишем значения в векторы, а затем объединим их в датафрейм:

```
x <- c(2, 2, 8, 10, 5)
y <- c(6, 8, 2, 3, 5)

dat <- cbind.data.frame(x, y)
dat
```

```
##    x y
## 1  2 6
## 2  2 8
## 3  8 2
## 4 10 3
## 5  5 5
```

Примечание 1: `cbind` соответствует объединению по столбцам (от *columns*), `rbind` — объединению по строкам (от *rows*).

Примечание 2: в данном случае функция `cbind()` тоже бы подошла, только стоит иметь в виду, что она создаёт матрицу, а не датафрейм. Проблема может возникнуть тогда, когда `x` и `y` являются векторами разного типа, при объединении в матрицу все элементы будут приведены к одному типу. Победит более сильный тип: например, строковый (*character*) вытеснит числовой (*numeric*), и все элементы станут текстовыми.

Добавим к нашему датафрейму названия строк. В R есть встроенный вектор `LETTERS`, содержащий заглавные буквы английского алфавита. Возьмём оттуда первые пять букв и запишем в названия строк:

```
rownames(dat) <- LETTERS[1:5]
dat
```

```
##    x y
## A  2 6
## B  2 8
## C  8 2
## D 10 3
## E  5 5
```

Построим матрицу расстояний D, но прежде шкалируем наши данные с помощью функции scale(): вычтем из каждого значения в столбце x среднее по столбцу и поделим на стандартное отклонение по столбцу, затем проделаем то же самое для столбца y:

```
D <- dist(scale(dat))
D
```

```
##           A         B         C         D
## B 0.8377078
## C 2.3705522 3.0213059
## D 2.5649459 3.0636522 0.6985260
## E 0.9373172 1.5106530 1.5106530 1.6293801
```

Матрица в R получилась довольно экономной: она показывает только расстояния между различными точками и не дублирует одни и те же расстояния, предполагая, что матрица симметричная. По умолчанию функция dist() считает евклидово расстояние. Запросим документацию функции через ?:

```
?dist
```

Список доступных расстояний:

- euclidean: евклидово расстояние;
- maximum: расстояние Чебышёва;
- manhattan: манхэттенское расстояние;
- canberra: канберрское расстояние;
- binary: асимметричное бинарное расстояние;
- minkowski: расстояние Минковского.

Теперь запустим иерархический кластерный анализ, выберем метод ближнего соседа, метод одиночной связи (single):

```
hc <- hclust(D, method = "single")
hc
```

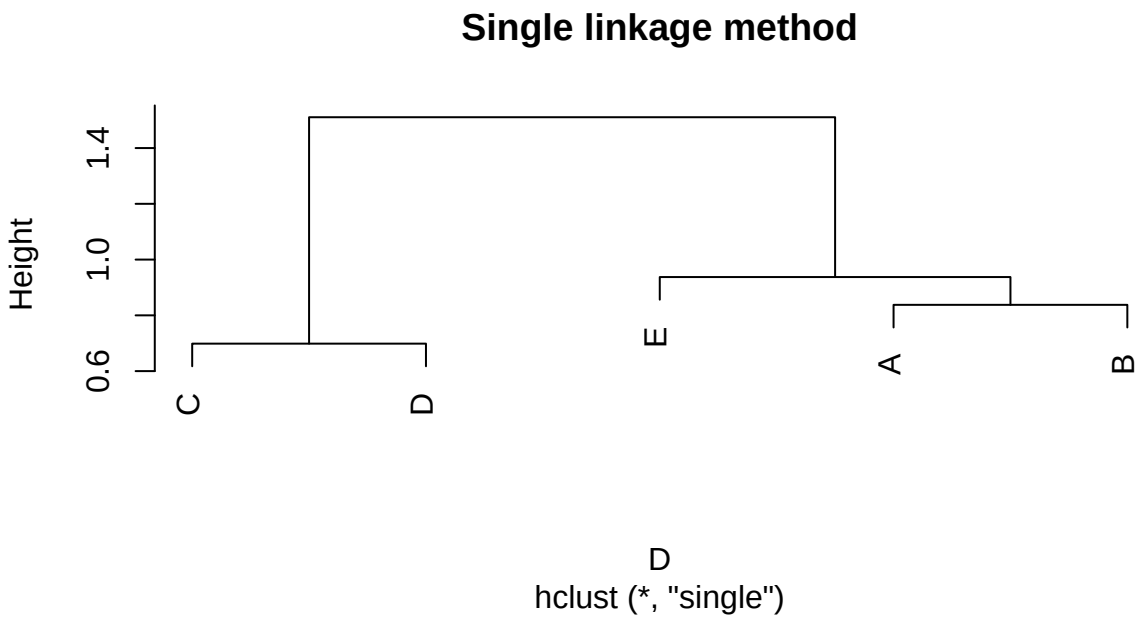
```
##
## Call:
## hclust(d = D, method = "single")
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 5
```

По умолчанию функция hclust() использует метод дальнего соседа, метод полной связи (complete), список основных методов такой:

- complete: метод полной связи;
- single: метод одиночной связи;
- average: метод средней связи;
- median: метод медианной связи;
- centroid: метод центроидной связи.

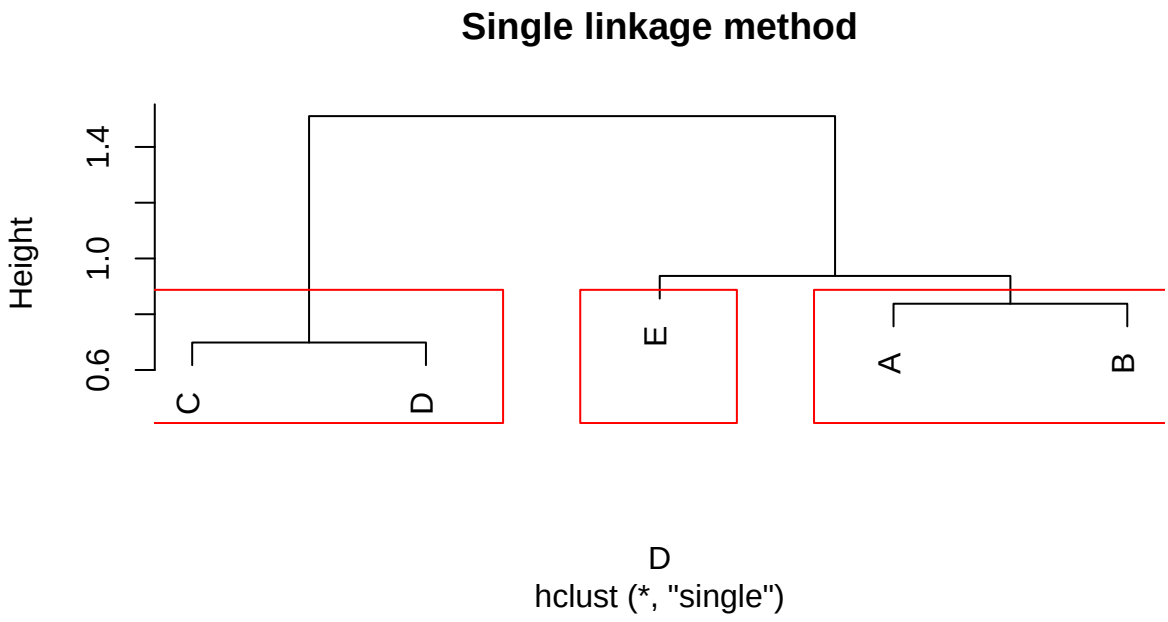
Осталось только построить дендрограмму, для этого потребуется только базовая функция `plot()`:

```
plot(hc, main = "Single linkage method")
```



Если мы определились с числом кластеров, можем выделить их на дендрограмме явно, с помощью прямоугольников:

```
plot(hc, main = "Single linkage method")
rect.hclust(hc, k = 3, border = "red")
```



Примечание: функция `rect.hclust()` добавляет прямоугольники на уже существующий график, то есть накладывает ещё один слой с графическими элементами. Поэтому эта строка с кодом должна запускаться сразу после `plot()`. Если запустить её два

раза с разным `k`, не перезапустив строку с `plot()`, прямоугольники тоже добавятся два раза, поэтому не забывайте обновлять саму дендрограмму.

Из объекта `hc`, который нам создала функция `hclust()`, можно извлекать отдельные элементы. Например, расстояния, при которых производилось объединение кластеров на каждой итерации алгоритма:

```
hc$height
```

```
## [1] 0.6985260 0.8377078 0.9373172 1.5106530
```

Расстояния отличаются от тех, которые были на лекции, так как на лекции мы пренебрегли шкалированием, ссылаясь на то, что оба показателя измерены в одних и тех же единицах измерения.