Иерархическая кластеризация. Сегментация потребителей безалкогольных напитков

Шаг 1. Чтение данных

Обеспечиваем, чтобы файл beverage.csv находился в рабочей папке

```
beverage.01 <- read.table("beverage.csv", header=T,
                          sep=";")
```

```
head(beverage.01, n=4)
```

```
##   numb.obs COKE D_COKE D_PEPSI D_7UP PEPSI SPRITE TAB SEVENUP
## 1        1    1      0       0     0     1      1   0       1
## 2        2    1      0       0     0     1      0   0       0
## 3        3    1      0       0     0     1      0   0       0
## 4        4    0      1       0     1     0      0   1       0
```

Вопросы для самопроверки

1. На что указывает опция header=T
2. На что указывает опция sep=“;”

Шаг 2. Удаление пропущенных значений

```
x <- c(1,2,3,NA,5,6,7)
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    2.25    4.00    4.00    5.75    7.00       1
```

В данной задаче пропущенных значений нет и удалять нечего.

В будущем, чтобы удалить строки, которые содержат пропущенные значения, закодированные NA делаем:

```
beverage.02 <- na.omit(beverage.01)
```

Шаг 3. Стандартизация переменных

В данной задаче переменные измерены в одной и той же шкале, поэтому стандартизировать их не надо.

В будущем, чтобы стандартизировать столбцы:

```
# Вариант 1 - к среднему 0 и стандартному отклонению 1
beverage.02 <- scale(beverage.01[,2:9], center=T, scale=T)
# Вариант 2 - к минимуму 0 и максимуму 1
maxs <- apply(beverage.01[,2:9], 2, max)
mins <- apply(beverage.01[,2:9], 2, min)
beverage.02 <- scale(beverage.01[,2:9], center = mins,
                     scale = maxs-mins)
# Вариант 3 - использовать rescaler() из пакета reshape
# Вариант 4 - использовать data.Normalization()
#             из пакета clusterSim
```

Шаг 4. Процедура кластерного анализа

Создаем матрицу попарных расстояний

```
dist.beverage <- dist(beverage.01[,2:9])
# по умолчанию используется евклидово расстояние
```

Здесь может быть проблема. Если число объектов N, то размерность матрицы dist - N x N и может быть сравнимо с размером ОЗУ.

Проводим кластерный анализ, результаты записываем в список clust.beverage

```
clust.beverage <- hclust(dist.beverage, "ward.D")
```

Смотрим краткую сводку результатов анализа

```
clust.beverage
```

```
##
## Call:
## hclust(d = dist.beverage, method = "ward.D")
##
## Cluster method   : ward.D
## Distance         : euclidean
## Number of objects: 34
```

Чаще всего предыдущие действия объединяют в одну команду

```
clust.beverage <- hclust(dist(beverage.01[,2:9]),"ward.D")
clust.beverage
```

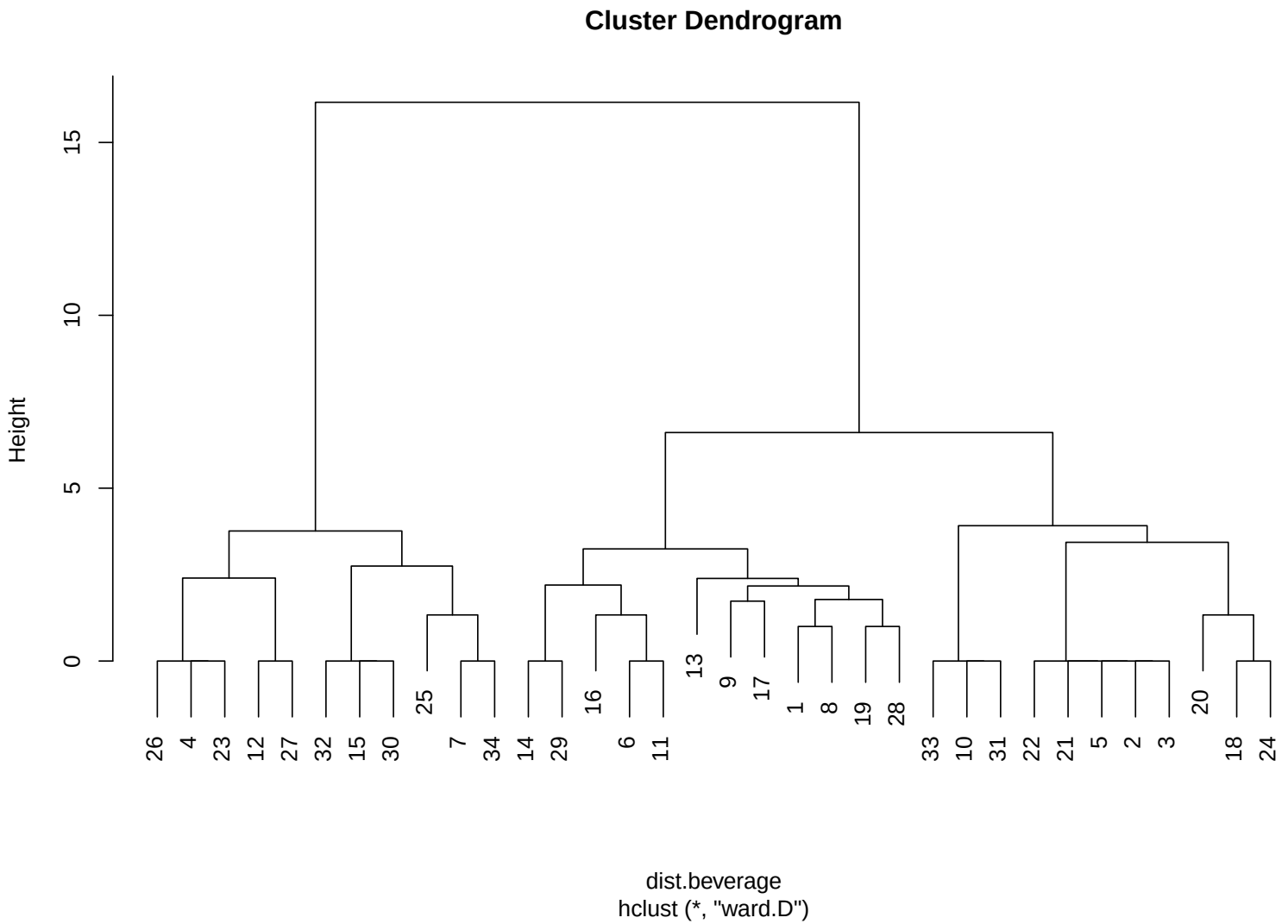
Вопросы для самопроверки:

1. Что вычисляется с помощью команды `dist(beverage.01[,2:9])`?
2. Почему именно 2:9 в команде `dist(beverage.01[,2:9])`?

Если нам нужно расстояние, которое не реализовано в функции `dist`, то создаем матрицу расстояний X и приписываем ей класс `dist` командой `as.dist(X)`.

Шаг 5. Построение дендрограммы

```
plot(clust.beverage)
```



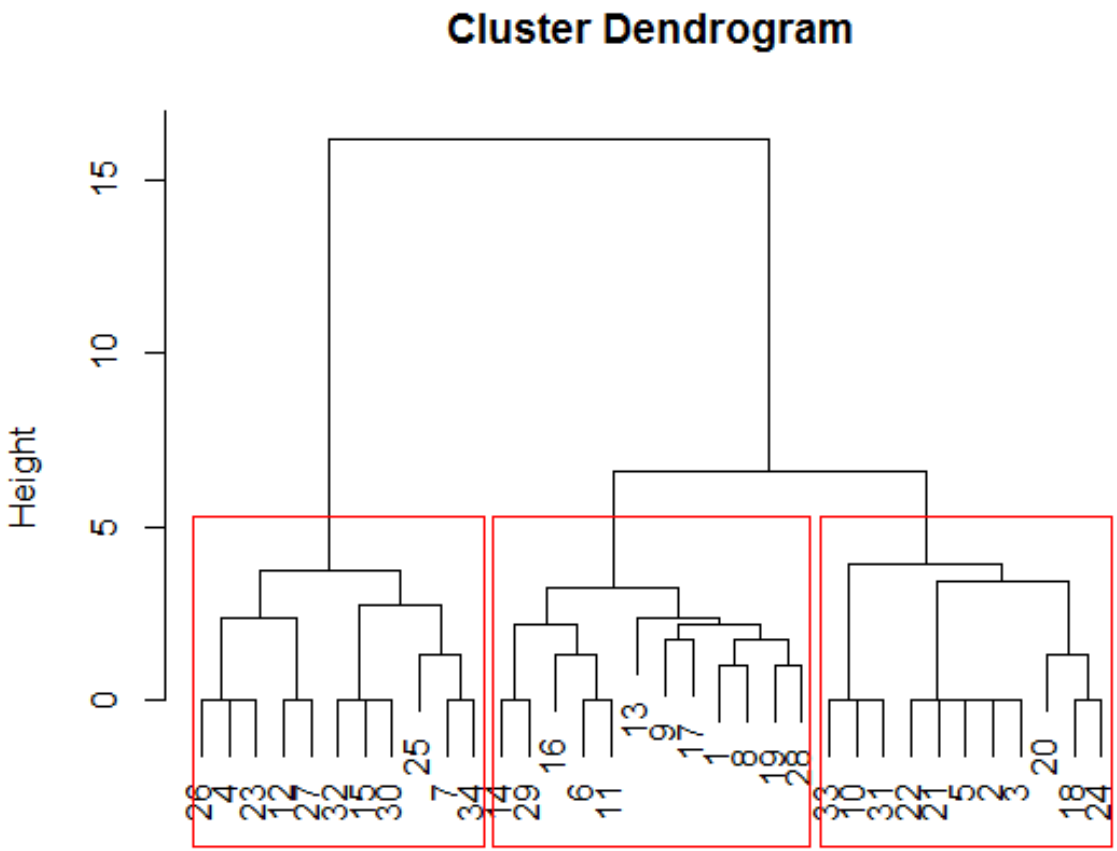
Шаг 6. Определение числа кластеров

Ответ: 3 кластера (а может быть 2. . .)

Сделаем красиво

На дендрограмме красными прямоугольниками выделим 3 кластера

```
rect.hclust(clust.beverage, k=3, border="red")
```



Разделим наблюдения на 3 кластера.

Вектор `groups` содержит номер кластера, в который попал классифицируемый объект

```
groups <- cutree(clust.beverage, k=3)
groups
```

```
##  [1] 1 2 2 3 2 1 3 1 1 2 1 3 1 1 3 1 1 2 1 2 2 2 3 2 3 3 3 1
```

Кластеры не обязательно будут пронумерованы слева направо.

Шаг 7. Интерпретация результатов

Что общего у объектов кластера?

Для каждого напитка определяем, какой процент потребителей в кластере пил этот напиток

```
colMeans(beverage.01[groups==1, 2:9])*100 # 1-й кластер
```

```
##      COKE    D_COKE   D_PEPSI     D_7UP     PEPSI    SPRITE       TAB
## 75.000000 25.000000  8.333333  8.333333 41.666667 91.666667  8.333333
##   SEVENUP
## 50.000000
```

```
colMeans(beverage.01[groups==2, 2:9])*100 # 2-й кластер
```

```
##       COKE     D_COKE    D_PEPSI      D_7UP      PEPSI     SPRITE
## 100.000000  27.272727   9.090909   0.000000 100.000000   0.000000
##        TAB    SEVENUP
##   0.000000  27.272727
```

```
colMeans(beverage.01[groups==3, 2:9])*100 # 3-й кластер
```

```
##      COKE    D_COKE   D_PEPSI     D_7UP     PEPSI    SPRITE       TAB
##   0.00000 100.00000  54.54545  54.54545   0.00000   0.00000  90.90909
##   SEVENUP
##   0.00000
```

Вариант интерпретации

- 3-й кластер: поклонники диетических напитков и здорового образа жизни.
- 2-й кластер: любители “классики”.
- 1-й кластер: недостаточный объем выборки не позволяет интерпретировать этот кластер.

Вопросы для самопроверки

1. Зачем было умножать на 100?
2. Почему средние арифметические дают долю потребителей?

Промежуточные итоги

Мы проделали следующее:

1. Провели кластеризацию.
2. Определили по дендрограмме число кластеров.
3. Проинтерпретировали результаты.

Что произойдет, если мы выберем решение с двумя кластерами?

Обзор результатов процедуры кластерного анализа

Какие результаты хранятся в списке clust.beverage?

```
names(clust.beverage)
```

```
## [1] "merge"       "height"      "order"       "labels"
## [6] "call"        "dist.method"
```

История объединения кластеров

```
clust.beverage$merge
```

```
##       [,1] [,2]
##  [1,]   -2   -3
##  [2,]   -5    1
##  [3,]  -21    2
##  [4,]  -22    3
##  [5,]   -4  -23
##  [6,]  -26    5
##  [7,]   -6  -11
##  [8,]   -7  -34
##  [9,]  -10  -31
## [10,]  -33    9
## [11,]  -12  -27
## [12,]  -14  -29
## [13,]  -15  -30
## [14,]  -32   13
## [15,]  -18  -24
```

Расстояния между кластерами в момент объединения

```
clust.beverage$height
```

```
##  [1]  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
##  [8]  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
## [15]  0.000000  1.000000  1.000000  1.333333  1.333333  1.333333  1.732051
## [22]  1.780239  2.170763  2.198038  2.389331  2.400000  2.747547  3.243086
## [29]  3.434434  3.761694  3.914506  6.609302 16.161976
```

Порядок следования объектов на дендрограмме

```
clust.beverage$order
```

```
##  [1] 26  4 23 12 27 32 15 30 25  7 34 14 29 16  6 11 13
## [24] 33 10 31 22 21  5  2  3 20 18 24
```

Еще...

```
# Метки классифицируемых объектов
clust.beverage$labels
```

```
## NULL
```

```
# Метод вычисления расстояний между кластерами
clust.beverage$method
```

```
## [1] "ward.D"
```

```
# Текст выполняемой команды
clust.beverage$call
```

```
## hclust(d = dist.beverage, method = "ward.D")
```

```
# Метод вычисления расстояний между объектами
clust.beverage$dist.method
```

```
## [1] "euclidean"
```

График “каменистая осыпь”

```
nclust <- seq(length(clust.beverage$height),1,-1)
plot(nclust, clust.beverage$height, type = "b",
     xlab = "Number of clusters",
     ylab = "Height",
     main = "Scree plot")
```