

## Урок 7. Структуры данных

Помимо векторов в R существуют и другие, более сложные структуры данных. Безусловно, чаще всего работать приходится с таблицами (датафреймами). Для примера объединим векторы в маленький датафрейм:

```
age <- c(32, 44, 21)
educ <- c("М", "М", "В")
dat <- cbind.data.frame(age, educ)
dat
```

```
##   age educ
## 1  32    М
## 2  44    М
## 3  21    В
```

Функция `cbind.data.frame()` объединяет векторы как столбцы таблицы (*c* от *columns*) и сохраняет тип каждого вектора. То есть, датафрейм может хранить данные разных типов, один столбец содержит один тип, так как столбец является вектором. Более детально с таблицами мы поработаем позже, когда будем обсуждать загрузку данных из файлов и предварительную обработку данных, а сейчас познакомимся с двумя структурами попроще – матрицами. Позже, обсуждая реальные данные, мы увидим, что работа с таблицами во многом похожа на работу с матрицами.

Что такое матрица в R? Это двумерный массив, таблица с элементами одного типа, чаще всего числами. Зачем матрицы нужны? Иногда можно столкнуться с функциями и библиотеками в R, которые принимают на вход исключительно матрицы, а не датафреймы. Например, с такими функциями можно столкнуться в сетевом анализе и в кластерном анализе. Если необходимо превратить датафрейм в матрицу, можно по аналогии с преобразованием типов воспользоваться функцией с префиксом `as`:

```
mat <- as.matrix(dat)
mat
```

```
##      age  educ
## [1,] "32" "М"
## [2,] "44" "М"
## [3,] "21" "В"
```

```
class(mat) # получилось!
```

```
## [1] "matrix"
```

**Важно:** матрицы всегда состоят из элементов одного типа, поэтому попытки включать элементы разных типов приводят к проблеме, которую мы наблюдали на векторах: один тип вытесняет другой. Так, здесь все элементы стали строковыми. В отличие от матриц, датафреймы могут хранить разные типы, но все равно в одном столбце только один конкретный - как будто это отдельный вектор. Создать матрицу с нуля можно множеством способов. Во-первых, можно объединить векторы и сделать их столбцами матрицы (опять *c* от *columns*):

```
educ <- c(2, 2, 1) # переопределим вектор educ для удобства
mcols <- cbind(age, educ)
mcols
```

```
##      age educ
## [1,]  32    2
## [2,]  44    2
## [3,]  21    1
```

Или строками (*r* от *rows*):

```
mrows <- rbind(age, educ)
mrows
```

```
##      [,1] [,2] [,3]
## age    32   44   21
## educ     2    2    1
```

Во-вторых, можно разбить один длинный вектор на строки или столбцы, если позволяет размерность (очевидно, мы не сможем из вектора в 12 элементов сделать матрицу 3 на 5). Возьмем последовательность чисел от 1 до 12 и запишем их по столбцам:

```
matrix(1:12, ncol = 3, nrow = 4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

В-третьих, можно создать матрицу заданной размерности (число строк и число столбцов) из одинаковых элементов, например, для того, чтобы потом заменить некоторые из них по условиям.

```
matrix(0, ncol = 5, nrow = 2) # матрица из нулей
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
```

К элементам матрицы нужно обращаться, задав номер строки и номер столбца, на пересечении которых он находится. Оба индекса вводятся в квадратных скобках через запятую:

```
m <- matrix(1:20, nrow = 4, ncol = 5)
m # допустим, есть такая матрица
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

```
m[2, 3] # 2ая строка 3й столбец
```

```
## [1] 10
```

Присвоив выбранному элементу значение, мы можем его заменить:

```
m[2, 3] <- 0
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6    0   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

Если нам нужно выбрать целую строку или целый столбец, один из индексов можно оставить пустым:

```
m[2,] # 2ая строка, все столбцы
```

```
## [1]  2  6  0 14 18
```

```
m[,3] # все строки, 3й столбец
```

```
## [1] 9 0 11 12
```

На этом мы завершаем знакомство со структурами данных и первый вводный модуль по программированию в R. Впереди вас ждет практическое задание и домашнее задание (оно уже проверяется и оценивается), а после мы поговорим об управляющих конструкциях и функциях, которые пригодятся нам в дальнейшем для создания своих интерактивных панелей. Желаю удачи!