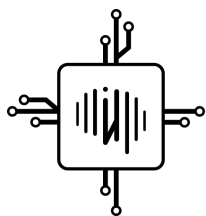


• **Развед
анализ
данных**

Введение в базовый R

Работа с векторами



Фонд
интеллект



Анастасия Жарикова

Лекция 1 - 2022

Общая информация о курсе

Преподаватели

- Жарикова Анастасия Александровна - *azharikova89@gmail.com*
- Пензар Дмитрий Дмитриевич - *dmitrypenzar1996@gmail.com*
- Валяева Анна Александровна - *valyaeva.ann@gmail.com*

Занятия

По пятницам

Лекция - 15:35-17:10

Семинар - 17:20 - 18:55

[Чат курса](#)

Содержание курса

- 15 занятий: лекция + семинар
- 15 домашних заданий
- 2 квиза
- Защита итогового проекта

Все материалы лекций и семинаров доступны слушателям курса

Что будет?



Base R

vs



Data manipulation

Statistical analysis

Visualization

Инструментарий - почему R?



В зависимости от области и задач может быть как избыточным так и недостаточным

Знакомство и установка

R и RStudio

R

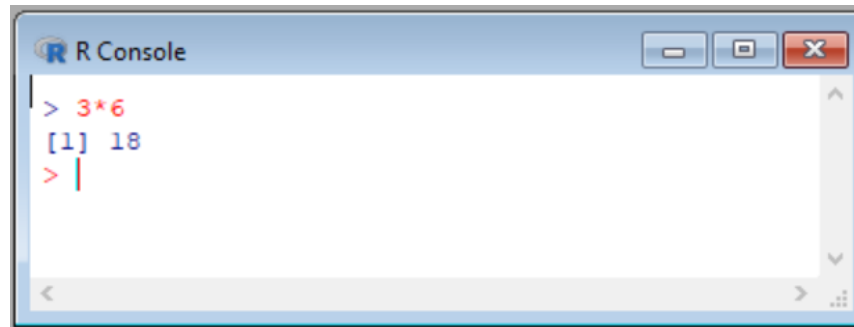
- Язык R - <https://cloud.r-project.org/>

R распространяется бесплатно с открытым исходным кодом

Доступно для Linux, macOS, Windows

Интерпретируемый язык программирования

Обширное сообщество разработчиков

A screenshot of an R Console window. The window title is "R Console". The console shows a prompt ">" followed by the command "3*6". The output is "[1] 18". The prompt ">" is followed by a vertical bar "|".

```
R Console
> 3*6
[1] 18
> |
```

R и RStudio

- RStudio - <https://rstudio.com/>

RStudio - интегрированная среда разработки (одна из)

R и RStudio позволяют

- оптимально организовать пространство для работы
- создавать и отлаживать программные сценарии
- удобно просматривать результаты работы
- получать справочную информацию
- работа с данными практически в любых форматах (импорт/экспорт)

RStudio - первые шаги

Логичная и удобная организация рабочего пространства позволяет структурировать и оптимизировать работу в целом

- Организуйте свою работу:
 - Проекты
 - Программные сценарии file.R
 - Отчеты file.Rmd
- Добавляйте комментарии в код!!!
- Сохраняйте полученные таблицы и графики в отдельные файлы

Подробности + практика - на семинаре

RStudio - общий вид

Файл не сохранен

Файл сохранен

Программные сценарии

Переменные

Environment History

Список выполненных команд

Files Plots Packages Help Viewer

Навигация по папкам и файлам

Графики

Получение справочной информации

Приглашение к началу работы

Консоль

```
1 a = c(seq(1, 10, by = 2))
2 b = c(rep(1:3, length.out = length(a)))
3 d = a * b
4 d
5
6 plot(a, b)
7
8
```

Variable	Type	Value
a	num [1:5]	1 3 5 7 9
b	int [1:5]	1 2 3 1 2
d	num [1:5]	1 6 15 7 18

```
> a = c(seq(1, 10, by = 2))
> b = c(rep(1:3, length.out = length(a)))
> d = a * b
> d
[1] 1 6 15 7 18
>
```

Программные сценарии и отчеты

file.R & file.Rmd

Программные сценарии

Первый шаг к воспроизводимости

Можно вводить команды по одной в консоль и выполнять их

Все действия будут отражены в вкладке History

Как понять, какие команды нам подошли, а какие выдали ошибку или неудовлетворительный результат?

Лучше сохранять только корректно работающие строки кода в нужной последовательности в файл, добавлять комментарии

Вы никогда не забудете, что и как вы делали

Получившийся файл всегда можно просмотреть и перезапустить, в том числе с другими параметрами и входными данными

file.R

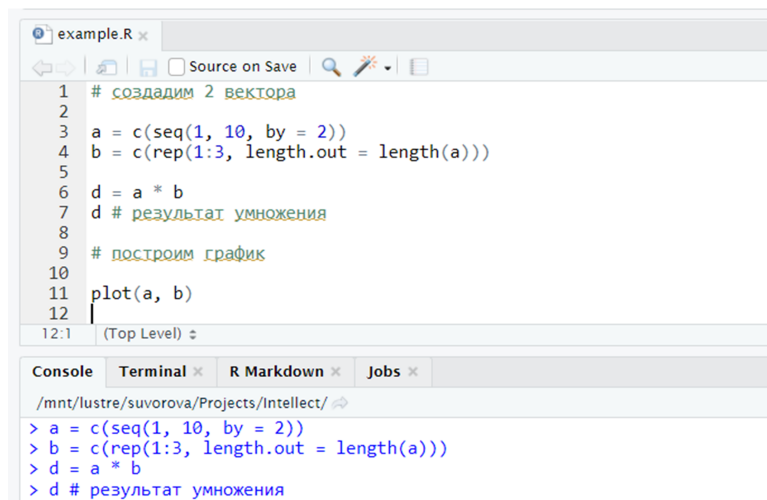
File => New File => R Script

Код + комментарии

Можно использовать в RStudio или запускать из командной строки

Ctrl + enter - запуск строки кода или выделенного фрагмента кода

["Горячие" клавиши](#)



The screenshot shows the RStudio interface. The top pane displays an R script named 'example.R' with the following code:

```
1 # создадим 2 вектора
2
3 a = c(seq(1, 10, by = 2))
4 b = c(rep(1:3, length.out = length(a)))
5
6 d = a * b
7 d # результат умножения
8
9 # построим график
10
11 plot(a, b)
12
```

The bottom pane shows the Console with the following output:

```
/mnt/lustre/suvorova/Projects/Intellect/
> a = c(seq(1, 10, by = 2))
> b = c(rep(1:3, length.out = length(a)))
> d = a * b
> d # результат умножения
```


file.Rmd

File -> New File -> R Markdown -> пример документа (нужно сохранить) -> Knit -> .html/.pdf/.doc

Создание отчета

Можно добавлять форматированный текст, формулы, таблицы, код, результат выполнения кода, картинки, ссылки

```
1 * ---
2 title: "Example"
3 author: "Анастасия Жарикова"
4 date: "февраль 2022"
5 output: html_document
6 * ---
7
8 # Заголовок
9
10 ## Заголовок поменьше
11
12 **Давайте** *что-нибудь* --одно-- посчитаем
13
14 Например,  $5 \times 3^2(2)$ 
15
16 В серой рамочке вы увидите код, а далее в белой рамочке результат выполнения кода
17
18
19 ```{r}
20 pi^2 * 3^2
21 ```
22
23
```

Example

Анастасия Жарикова

февраль 2022

Заголовок

Заголовок поменьше

Давайте что-нибудь *одно* посчитаем

например, 5×3^2

В серой рамочке вы увидите код, а далее в белой рамочке результат выполнения кода

```
pi^2 * 3^2
```

```
## [1] 28.27433
```

Markdown

Фрагменты кода записывают в чанки (“chunk”):

```
## R Markdown
```

```
This is an R Markdown document. Markdown is a simple formatting syntax for  
see <http://rmarkdown.rstudio.com>.
```

```
When you click the **Knit** button a document will be generated that includ  
document. You can embed an R code chunk like this:
```

```
```${r cars}  
summary(cars)
```
```

← chunk

Запускается целиком или по чанкам, сразу видны ошибки и проблемы

Переменные и функции, инициализированные в одном chunk, будут работать во всей сессии

Markdown - основные настройки

Устанавливаем TRUE или FALSE

echo - показывать код

eval - выполнять код

message и warnings - показывать предупреждения

Отображать и код и результат его выполнения

```
```{r}
dim (mtcars)
```
```

Отображать и код и результат его выполнения

```
dim (mtcars)
```

```
## [1] 32 11
```

Отображать только результат выполнения кода

```
```{r, echo = F}
dim (mtcars)
```
```

Отображать только результат выполнения кода

```
## [1] 32 11
```

Код отображается, но не выполняется

```
```{r, eval = F}
dim (mtcars)
```
```

Код отображается, но не выполняется

```
dim (mtcars)
```

Выполнение кода внутри строки текста

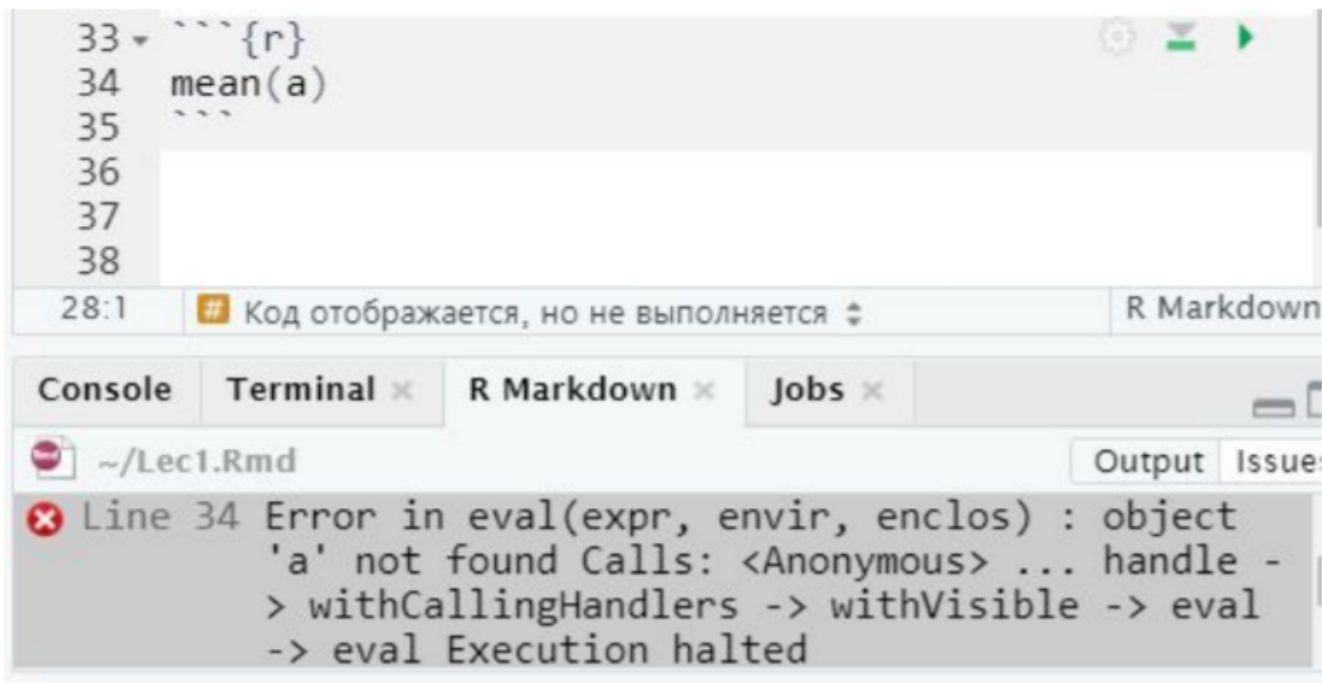
```
Two plus two equals `r 2+2`
```

Выполнение кода внутри строки текста

```
Two plus two equals 4
```

Markdown

В случае ошибки в коде итоговый документ не будет создан, появится сообщение об ошибке



The screenshot shows an R Markdown editor window. The code editor displays the following R code:

```
33 {r}
34 mean(a)
35
36
37
38
```

Below the code editor, a status bar indicates "28:1" and a warning icon with the text "Код отображается, но не выполняется" (Code is displayed but not executed). The "R Markdown" tab is active. Below the editor, a console window shows the error message:

```
~/Lec1.Rmd
Output Issue:
✖ Line 34 Error in eval(expr, envir, enclos) : object 'a' not found
Calls: <Anonymous> ... handle -> withCallingHandlers -> withVisible -> eval -> eval
Execution halted
```

Читаем ошибку - исправляем - перезапускаем

Markdown - почитать

[Markdown в R](#)

[Markdown в R - reference](#)

Навигация и организация

С чем мы работаем и где находимся?

```
getwd() # узнать рабочую директорию
setwd("path/to/Newdir") # задать рабочую директорию
dir() # список файлов в рабочей директории
dir.create("New_dir") # создать директорию New_dir
dir.exists("Dir") # проверить, существует ли директория Dir
unlink("Dir", recursive = T) # удалить директорию Dir
file.exists("file.txt") # проверить, существует ли файл file.txt
file.remove("file.txt") # удалить файл file.txt
sessionInfo() # информация о загруженной сессии R
```

Информация о загруженной сессии R

Знать версии программ и пакетов необходимо для возможности воспроизведения результатов

```
sessionInfo()
```

```
R version 3.6.3 (2020-02-29)
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
Running under: Scientific Linux release 6.8 (Carbon)
```

```
Matrix products: default
```

```
BLAS: /mnt/lustre/tools/R/R-3.6.3/lib64/R/lib/libRblas.so
```

```
LAPACK: /mnt/lustre/tools/R/R-3.6.3/lib64/R/lib/libRlapack.so
```

```
locale:
```

```
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8     LC_NAME=C
[9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
other attached packages:
```

```
[1] cowplot_1.1.1  forcats_0.5.1  stringr_1.4.0  dplyr_1.0.7
[5] purrr_0.3.4    readr_2.1.1    tidyr_1.1.4    tibble_3.1.6
[9] ggplot2_3.3.5  tidyverse_1.3.1
```

```
loaded via a namespace (and not attached):
```

```
[1] tidyselect_1.1.1  xfun_0.29      bslib_0.3.1
[4] haven_2.4.3       colorspace_2.0-2 vctrs_0.3.8
[7] generics_0.1.1   htmltools_0.5.2 yaml_2.2.1
[10] utf8_1.2.2        rlang_0.4.12   jquerylib_0.1.4
[13] pillar_1.6.4     withr_2.4.3    glue_1.6.0
[16] DBI_1.1.1         dbplyr_2.1.1   uuid_0.1-4
[19] modelr_0.1.8     readxl_1.3.1   lifecycle_1.0.1
```


Работа с векторами

Основной тип данных - вектор

```
x <- 1:5
```

```
x
```

```
[1] 1 2 3 4 5
```

Основной тип данных - вектор

```
x <- 1:5
```

```
[1] 1 2 3 4 5
```

```
x
```

```
length(x)
```

```
[1] 5
```

Основной тип данных - вектор

```
x <- 1:5  
x  
length(x)  
3 # это тоже вектор длины 1
```

[1] 1 2 3 4 5

[1] 5

[1] 3

Основной тип данных - вектор

```
x <- 1:5 [1] 1 2 3 4 5
x
length(x) [1] 5
3 # это тоже вектор длины 1
z <- c("a", "b", "d") # тоже вектор, только [1] 3
строковый
```

Основной тип данных - вектор

```
x <- 1:5 [1] 1 2 3 4 5
x
length(x) [1] 5
3 # это тоже вектор длины 1
z <- c("a", "b", "d") # тоже вектор, только [1] 3
    строковый
z [1] "a" "b" "d"
```

Внимание!

При перезаписывании содержимого переменной никаких предупреждений не будет

```
x <- 2  
x
```

[1] 2

```
x <- 5  
x
```

[1] 5

Будьте внимательны при выборе имен для переменных

Индексация

Индексация элементов в R начинается с 1 (не с 0)

```
x <- 15:50  
x
```

```
[1] 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39  
[26] 40 41 42 43 44 45 46 47 48 49 50
```


Способы создания вектора

Оператор c()

```
c(1, 2, 3)
```

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

Объединение

```
x <- 1:3  
x
```

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

```
x <- c(x, 5:7)  
x
```

```
[1] 1 2 3 5 6 7
```

Способы создания вектора

Последовательности

```
1:10
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
5:-3
```

```
[1] 5 4 3 2 1 0 -1 -2 -3
```

```
seq(from = 1, to = 8, by = 2)
```

```
[1] 1 3 5 7
```

Способы создания вектора

Повторы

```
rep(1:3, times = 3)
```

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

```
rep(1:3, each = 3)
```

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

```
rep(1:3, length.out = 8)
```

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

Способы создания вектора

Взять 10 случайных чисел от 1 до 30

```
sample(1:30, 10, replace = T)
```

```
[1] 6 2 5 14 21 25 10 28 23 18
```

```
sample(1:30, 10, replace = T)
```

```
[1] 26 1 24 1 14 18 28 19 15 11
```

```
set.seed(123)
```

```
sample(1:30, 10, replace = T)
```

```
[1] 15 19 14 3 10 18 22 11 5 20
```

```
set.seed(123)
```

```
sample(1:30, 10, replace = T)
```

```
[1] 15 19 14 3 10 18 22 11 5 20
```

Способы создания вектора

Нужно сгенерировать заданное количество чисел из известного распределения

- `rnorm(n, mean, sd)` - нормальное распределение
- `runif(n, min, max)` - равномерное распределение
- `rbinom(n, size, prob)` - биномиальное распределение
- `rpois(n, lambda)` - распределение Пуассона

```
set.seed(123)  
rpois(20,10)
```

```
[1] 8 9 14 10 10 15 11 5 4 13 11 11 10 8 15 11 3 7 6 8
```

Немного о функциях

Разберем функцию `rnorm`

В вкладке "Help" можно вызвать справочную информацию о функции

`rnorm(n, mean = 0, sd = 1)`

У функции есть параметры с значениями по умолчанию

```
set.seed(123)
rnorm(3) # получим 3 случайных значения из стандартного нормального распределения
```

```
[1] -0.5604756 -0.2301775 1.5587083
```

```
set.seed(123)
rnorm(3, mean = 1, sd = 2) # указываем значения параметров
```

```
[1] -0.1209513 0.5396450 4.1174166
```

```
set.seed(123)
rnorm(3, 1, 2) # чаще всего можно не указывать имена параметров в явном виде
```

```
[1] -0.1209513 0.5396450 4.1174166
```

Вектор - данные одного типа

```
x <- c(0.5, 1.2, 3.6) # точка - разделитель  
разрядов  
typeof(x) # double or numeric
```

```
[1] "double"
```

Вектор - данные одного типа

```
x <- c(0.5, 1.2, 3.6) # точка - разделитель  
разрядов  
typeof(x) # double or numeric
```

```
[1] "double"
```

```
[1] "double"
```

```
x <- c(5, 2, 3)  
typeof(x) # double
```


Вектор - данные одного типа

```
x <- c(0.5, 1.2, 3.6) # точка - разделитель  
разрядов
```

```
typeof(x) # double or numeric
```

```
x <- c(5, 2, 3)
```

```
typeof(x) # double
```

```
x <- c(0.5, 1.2, 3)
```

```
typeof(x) # double
```

```
[1] "double"
```

```
[1] "double"
```

```
[1] "double"
```

Вектор - данные одного типа

```
x <- c(0.5, 1.2, 3.6) # точка - разделитель  
разрядов
```

```
typeof(x) # double or numeric
```

```
[1] "double"
```

```
[1] "double"
```

```
x <- c(5, 2, 3)
```

```
typeof(x) # double
```

```
[1] "double"
```

```
x <- c(0.5, 1.2, 3)
```

```
typeof(x) # double
```

```
[1] "integer"
```

```
x <- 1:5
```

```
typeof(x) # integer
```

Вектор - данные одного типа

```
x <- c(0.5, 1.2, 3.6) # точка - разделитель  
                      разрядов  
typeof(x) # double or numeric [1] "double"  
  
x <- c(5, 2, 3) [1] "double"  
typeof(x) # double  
  
x <- c(0.5, 1.2, 3) [1] "integer"  
typeof(x) # double  
  
x <- 1:5 [1] "logical"  
typeof(x) # integer  
  
x <- c(T,F,F,T)  
typeof(x) # logical
```

Вектор - данные одного типа

```
x <- c(0.5, 1.2, 3.6) # точка - разделитель  
разрядов
```

```
typeof(x) # double or numeric
```

```
x <- c(5, 2, 3)
```

```
typeof(x) # double
```

```
x <- c(0.5, 1.2, 3)
```

```
typeof(x) # double
```

```
x <- 1:5
```

```
typeof(x) # integer
```

```
x <- c(T,F,F,T)
```

```
typeof(x) # logical
```

```
x <- c('a', 'b', 'c')
```

```
typeof(x) # character
```

```
[1] "double"
```

```
[1] "double"
```

```
[1] "double"
```

```
[1] "integer"
```

```
[1] "logical"
```

```
[1] "character"
```

Приведение типов

```
x <- c(F, 1, 2, T)
typeof(x)
```

```
[1] "double"
```

Приведение типов

```
x <- c(F, 1, 2, T)
```

```
typeof(x)
```

```
x
```

```
[1] "double"
```

```
[1] 0 1 2 1
```

Приведение типов

```
x <- c(F, 1, 2, T)
```

```
typeof(x)
```

```
x
```

```
[1] "double"
```

```
[1] 0 1 2 1
```

```
x <- c(1, 2.8)
```

```
typeof(x)
```

```
[1] "double"
```

Приведение типов

```
x <- c(F, 1, 2, T)
typeof(x)
x
x <- c(1, 2.8)
typeof(x)
x
```

[1] "double"

[1] 0 1 2 1

[1] "double"

[1] 1.0 2.8

Приведение типов

```
x <- c(F, 1, 2, T)
```

```
typeof(x)
```

```
x
```

```
x <- c(1, 2.8)
```

```
typeof(x)
```

```
x
```

```
x <- c(1, 'a', F, 5.5)
```

```
typeof(x)
```

```
[1] "double"
```

```
[1] 0 1 2 1
```

```
[1] "double"
```

```
[1] 1.0 2.8
```

```
[1] "character"
```

Приведение типов

```
x <- c(F, 1, 2, T)
```

```
typeof(x)
```

```
x
```

```
x <- c(1, 2.8)
```

```
typeof(x)
```

```
x
```

```
x <- c(1, 'a', F, 5.5)
```

```
typeof(x)
```

```
x
```

```
[1] "double"
```

```
[1] 0 1 2 1
```

```
[1] "double"
```

```
[1] 1.0 2.8
```

```
[1] "character"
```

```
[1] "1"      "a"      "FALSE" "5.5"
```

Преобразование типов

```
x <- c(F, F, T, T)
typeof(x)
```

```
[1] "logical"
```

Преобразование типов

```
x <- c(F, F, T, T)
```

```
typeof(x)
```

```
x
```

```
[1] "logical"
```

```
[1] FALSE FALSE TRUE TRUE
```

Преобразование типов

```
x <- c(F, F, T, T)
```

```
typeof(x)
```

```
x
```

```
x <- as.numeric(x)
```

```
typeof(x)
```

```
[1] "logical"
```

```
[1] FALSE FALSE TRUE TRUE
```

```
[1] "double"
```

Преобразование типов

```
x <- c(F, F, T, T)
```

```
typeof(x)
```

```
x
```

```
x <- as.numeric(x)
```

```
typeof(x)
```

```
x
```

```
[1] "logical"
```

```
[1] FALSE FALSE TRUE TRUE
```

```
[1] "double"
```

```
[1] 0 0 1 1
```

Имена элементов вектора

```
x <- 1:3
```

```
x
```

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

Имена элементов вектора

```
x <- 1:3
```

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

```
x
```

```
names(x)
```

```
NULL
```


Имена элементов вектора

```
x <- 1:3
```

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

```
x
```

```
names(x)
```

```
NULL
```

```
names(x) <- c('a', 'b', 'c')
```

Имена элементов вектора

```
x <- 1:3  
x  
names(x)  
names(x) <- c('a', 'b', 'c')  
x
```

[1] 1 2 3

NULL

a b c

1 2 3

Имена элементов вектора

```
x <- 1:3
```

```
x
```

```
names(x)
```

```
names(x) <- c('a', 'b', 'c')
```

```
x
```

```
names(x)
```

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

```
NULL
```

```
a b c
```

```
1 2 3
```

```
[1] "a" "b" "c"
```

Имена элементов вектора

```
x <- 1:3  
x  
names(x)  
names(x) <- c('a', 'b', 'c')  
x  
names(x)
```

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

```
NULL
```

```
a b c
```

```
1 2 3
```

```
[1] "a" "b" "c"
```

значение

- индекс или порядковый номер
- имя (может быть пустым)



Операции с векторами

Операции с векторами

```
x <- 1:5
```

```
x
```

```
[1] 1 2 3 4 5
```

Операции с векторами

```
x <- 1:5
```

```
[1] 1 2 3 4 5
```

```
x
```

```
x + 3
```

```
[1] 4 5 6 7 8
```

Операции с векторами

```
x <- 1:5
```

```
x
```

```
x + 3
```

```
y <- 6:10
```

```
[1] 1 2 3 4 5
```

```
[1] 4 5 6 7 8
```


Операции с векторами

```
x <- 1:5
```

```
[1] 1 2 3 4 5
```

```
x
```

```
x + 3
```

```
[1] 4 5 6 7 8
```

```
y <- 6:10
```

```
y
```

```
[1] 6 7 8 9 10
```

Операции с векторами

```
x <- 1:5
```

```
x
```

```
x + 3
```

```
y <- 6:10
```

```
y
```

```
x + y
```

```
[1] 1 2 3 4 5
```

```
[1] 4 5 6 7 8
```

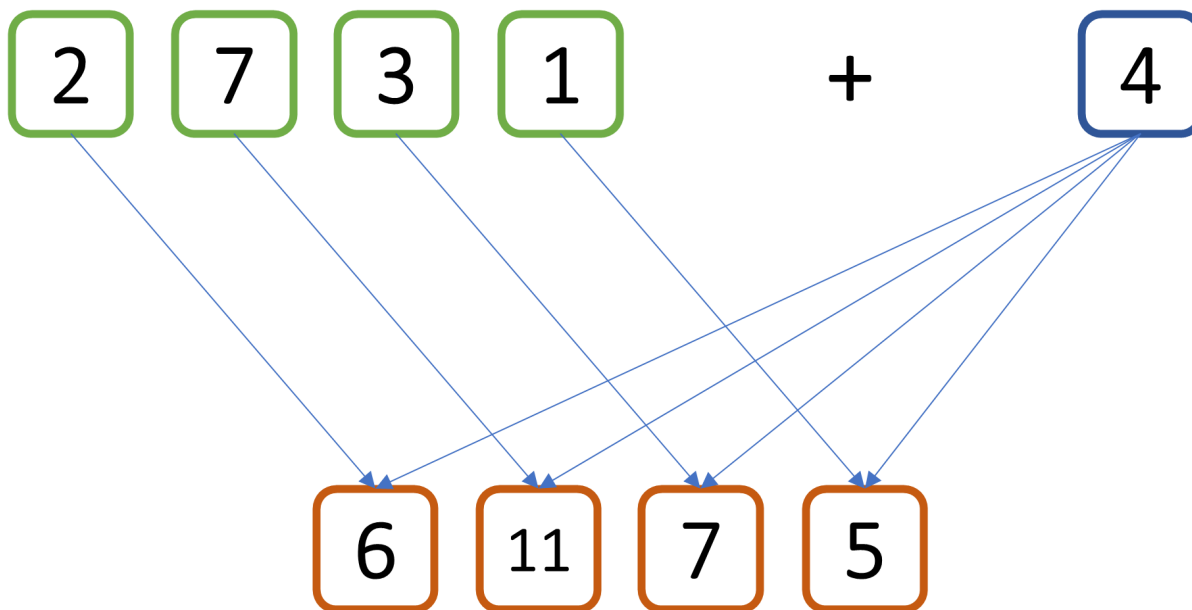
```
[1] 6 7 8 9 10
```

```
[1] 7 9 11 13 15
```

Операции с векторами

```
> x <- c(2, 7, 3, 1)
```

```
> x + 4
```



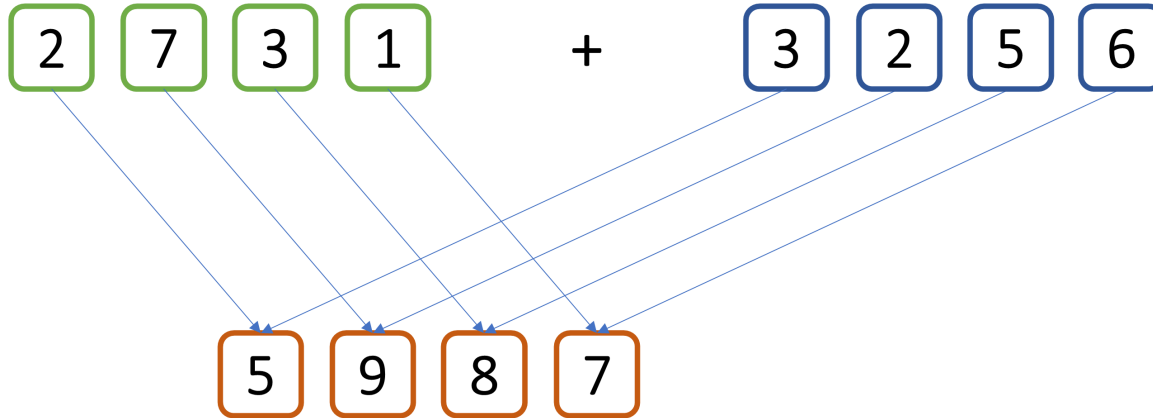
К каждому элементу вектора x прибавляем 4

Операции с векторами

```
> x <- c(2, 7, 3, 1)
```

```
> y <- c(3, 2, 5, 6)
```

```
> x + y
```



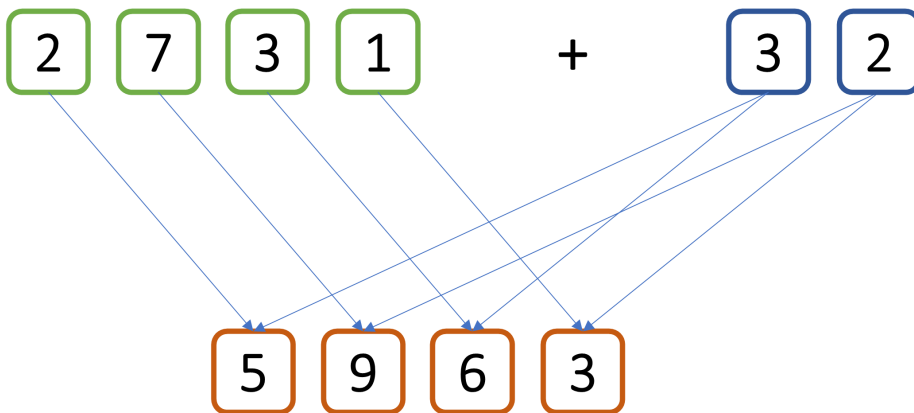
К первому элементу вектора x прибавляем первый элемент вектора y
К второму элементу вектора x прибавляем второй элемент вектора y
и т.д.

Операции с векторами

```
> x <- c(2, 7, 3, 1)
```

```
> y <- c(3, 2)
```

```
> x + y
```



К первому элементу вектора x прибавляем первый элемент вектора y

К второму элементу вектора x прибавляем второй элемент вектора y

К третьему элементу вектора x прибавляем первый элемент вектора y

К четвертому элементу вектора x прибавляем второй элемент вектора y

Операции с векторами

```
x <- 1:5
```

```
x
```

```
[1] 1 2 3 4 5
```

Операции с векторами

```
x <- 1:5
```

```
x
```

```
x+3
```

```
[1] 1 2 3 4 5
```

```
[1] 4 5 6 7 8
```

Операции с векторами

```
x <- 1:5
```

```
x
```

```
x+3
```

```
y <- 6:10
```

```
y
```

```
[1] 1 2 3 4 5
```

```
[1] 4 5 6 7 8
```

```
[1] 6 7 8 9 10
```


Операции с векторами

```
x <- 1:5
```

```
x
```

```
x+3
```

```
y <- 6:10
```

```
y
```

```
x + y
```

```
[1] 1 2 3 4 5
```

```
[1] 4 5 6 7 8
```

```
[1] 6 7 8 9 10
```

```
[1] 7 9 11 13 15
```

Операции с векторами

```
x <- 1:5
```

```
x
```

```
x+3
```

```
y <- 6:10
```

```
y
```

```
x + y
```

```
x^2
```

```
[1] 1 2 3 4 5
```

```
[1] 4 5 6 7 8
```

```
[1] 6 7 8 9 10
```

```
[1] 7 9 11 13 15
```

```
[1] 1 4 9 16 25
```

Операции с векторами

```
x <- 1:5  
x  
x+3  
y <- 6:10  
y  
x + y  
x^2  
-x
```

```
[1] 1 2 3 4 5
```

```
[1] 4 5 6 7 8
```

```
[1] 6 7 8 9 10
```

```
[1] 7 9 11 13 15
```

```
[1] 1 4 9 16 25
```

```
[1] -1 -2 -3 -4 -5
```

Операции с векторами

```
x <- 1:5  
x
```

```
[1] 1 2 3 4 5
```

```
y <- 1:2  
y
```

```
[1] 1 2
```

```
x+y
```

```
Warning in x + y: longer object length is not a multiple of shorter object  
length
```

```
[1] 2 4 4 6 6
```

Операции с векторами

```
x <- 1:6
```

```
x
```

```
[1] 1 2 3 4 5 6
```

Операции с векторами

```
x <- 1:6
```

```
x
```

```
x > 4
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

Операции с векторами

```
x <- 1:6
```

```
x
```

```
x > 4
```

```
x >= 4
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE TRUE TRUE
```

Операции с векторами

```
x <- 1:6
```

```
x
```

```
x > 4
```

```
x >= 4
```

```
x != 4
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE TRUE TRUE
```

```
[1] TRUE TRUE TRUE FALSE TRUE TRUE
```


Операции с векторами

```
x <- 1:6  
x  
x > 4  
x >= 4  
x != 4  
x == 4
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE TRUE TRUE
```

```
[1] TRUE TRUE TRUE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE FALSE FALSE
```

Операции с векторами

```
x <- 1:6  
x  
x > 4  
x >= 4  
x != 4  
x == 4  
replace(x, x == 2, 12)
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE TRUE TRUE
```

```
[1] TRUE TRUE TRUE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE FALSE FALSE
```

```
[1] 1 12 3 4 5 6
```

Операции с векторами

```
x <- 1:6  
x  
x > 4  
x >= 4  
x != 4  
x == 4  
replace(x, x == 2, 12)  
x %in% c(1, 4)
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE TRUE TRUE
```

```
[1] TRUE TRUE TRUE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE FALSE FALSE
```

```
[1] 1 12 3 4 5 6
```

```
[1] TRUE FALSE FALSE TRUE FALSE FALSE
```

Операции с векторами

```
x <- 1:6  
x  
x > 4  
x >= 4  
x != 4  
x == 4  
replace(x, x == 2, 12)  
x %in% c(1, 4)  
x = 4
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE TRUE TRUE
```

```
[1] TRUE TRUE TRUE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE FALSE FALSE
```

```
[1] 1 12 3 4 5 6
```

```
[1] TRUE FALSE FALSE TRUE FALSE FALSE
```

Операции с векторами

```
x <- 1:6
x
x > 4
x >= 4
x != 4
x == 4
replace(x, x == 2, 12)
x %in% c(1, 4)
x = 4
x
```

```
[1] 1 2 3 4 5 6
```

```
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE TRUE TRUE
```

```
[1] TRUE TRUE TRUE FALSE TRUE TRUE
```

```
[1] FALSE FALSE FALSE TRUE FALSE FALSE
```

```
[1] 1 12 3 4 5 6
```

```
[1] TRUE FALSE FALSE TRUE FALSE FALSE
```

```
[1] 4
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
[1] 5 16 8 32 56 2
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```


Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
x[c(2, 5)] # взять 2ой и 5ый элементы
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```

```
[1] 16 56
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
x[c(2, 5)] # взять 2ой и 5ый элементы
```

```
x[-1] # без первого элемента
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```

```
[1] 16 56
```

```
[1] 16 8 32 56 2
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
x[c(2, 5)] # взять 2ой и 5ый элементы
```

```
x[-1] # без первого элемента
```

```
x > 10
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```

```
[1] 16 56
```

```
[1] 16 8 32 56 2
```

```
[1] FALSE TRUE FALSE TRUE TRUE FALSE
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
x[c(2, 5)] # взять 2ой и 5ый элементы
```

```
x[-1] # без первого элемента
```

```
x > 10
```

```
x[x > 10] # отобразить элементы больше 10
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```

```
[1] 16 56
```

```
[1] 16 8 32 56 2
```

```
[1] FALSE TRUE FALSE TRUE TRUE FALSE
```

```
[1] 16 32 56
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
x[c(2, 5)] # взять 2ой и 5ый элементы
```

```
x[-1] # без первого элемента
```

```
x > 10
```

```
x[x > 10] # отобразить элементы больше 10
```

```
x[x >= 5 & x < 10] # отобразить элементы больше  
или равные 5 И меньше 10
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```

```
[1] 16 56
```

```
[1] 16 8 32 56 2
```

```
[1] FALSE TRUE FALSE TRUE TRUE FALSE
```

```
[1] 16 32 56
```

```
[1] 5 8
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
x[c(2, 5)] # взять 2ой и 5ый элементы
```

```
x[-1] # без первого элемента
```

```
x > 10
```

```
x[x > 10] # отобразить элементы больше 10
```

```
x[x >= 5 & x < 10] # отобразить элементы больше  
или равные 5 И меньше 10
```

```
x[x >= 5 | x < 10] # отобразить элементы больше  
или равные 5 ИЛИ меньше 10
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```

```
[1] 16 56
```

```
[1] 16 8 32 56 2
```

```
[1] FALSE TRUE FALSE TRUE TRUE FALSE
```

```
[1] 16 32 56
```

```
[1] 5 8
```

```
[1] 5 16 8 32 56 2
```

Срезы: по индексам

```
x <- c(5, 16, 8, 32, 56, 2)
```

```
x
```

```
x[1] # взять 1ый элемент
```

```
x[2:4] # взять с 2го по 4ый элементы
```

```
x[c(2, 5)] # взять 2ой и 5ый элементы
```

```
x[-1] # без первого элемента
```

```
x > 10
```

```
x[x > 10] # отобразить элементы больше 10
```

```
x[x >= 5 & x < 10] # отобразить элементы больше  
или равные 5 И меньше 10
```

```
x[x >= 5 | x < 10] # отобразить элементы больше  
или равные 5 ИЛИ меньше 10
```

```
x[2] <- 100 # заменить второй элемент вектора на  
100
```

```
[1] 5 16 8 32 56 2
```

```
[1] 5
```

```
[1] 16 8 32
```

```
[1] 16 56
```

```
[1] 16 8 32 56 2
```

```
[1] FALSE TRUE FALSE TRUE TRUE FALSE
```

```
[1] 16 32 56
```

```
[1] 5 8
```

```
[1] 5 16 8 32 56 2
```

Срезы: по именам

```
x <- c(5,16,8,32,56,2)
names(x) <- paste('v', 1:length(x), sep = '')
x
```

```
v1 v2 v3 v4 v5 v6
5 16 8 32 56 2
```


Срезы: по именам

```
x <- c(5,16,8,32,56,2)
names(x) <- paste('v', 1:length(x), sep = '')
x
x['v3'] # взять элемент вектора x с именем v3
```

```
v1 v2 v3 v4 v5 v6
5 16 8 32 56 2

v3
8
```

Срезы: по именам

```
x <- c(5,16,8,32,56,2)
names(x) <- paste('v', 1:length(x), sep = '')
x
x['v3'] # взять элемент вектора x с именем v3
x[c('v1', 'v3')]
```

```
v1 v2 v3 v4 v5 v6
5 16 8 32 56 2
```

```
v3
8
```

```
v1 v3
5 8
```

Срезы: по именам

```
x <- c(5,16,8,32,56,2)
names(x) <- paste('v', 1:length(x), sep = '')
x
x['v3'] # взять элемент вектора x с именем v3
x[c('v1', 'v3')]
x[c(1, 3)]
```

```
v1 v2 v3 v4 v5 v6
5 16 8 32 56 2
```

```
v3
8
```

```
v1 v3
5 8
```

```
v1 v3
5 8
```

Срезы: по именам

```
x <- c(5,16,8,32,56,2)
names(x) <- paste('v', 1:length(x), sep = '')
x
x['v3'] # взять элемент вектора x с именем v3
x[c('v1', 'v3')]
x[c(1, 3)]
```

```
v1 v2 v3 v4 v5 v6
5 16 8 32 56 2
```

```
v3
8
```

```
v1 v3
5 8
```

```
v1 v3
5 8
```

Полезные функции

```
x <- 1:6  
y <- rep(7, 4)  
x
```

```
[1] 1 2 3 4 5 6
```

```
y
```

```
[1] 7 7 7 7
```

```
any(x == 5)
```

```
[1] TRUE
```

```
any(x == 12)
```

```
[1] FALSE
```

```
all(x == 3)
```

```
[1] FALSE
```

```
all(y == 7)
```

Работа с переменными

Работа с переменными

```
ls() # список переменных из глобального окружения  
rm(a, b) # удалить переменные a и b  
rm(list=ls()) # удалить все переменные из глобального окружения
```

Названия переменных **НЕ** должны:

- начинаться с цифр
- содержать пробелы
- содержать специальные символы (+, -, /, \, *, ^, ...)
- содержать зарезервированные слова (if, else, repeat, while, function, for, in, next, break, TRUE, FALSE, NULL, Inf, NaN, NA, ...)

Работа с переменными

```
ls()
```

```
[1] "x" "y" "z"
```

```
rm(list=ls())  
ls()
```

```
character(0)
```


Матрицы

Что такое матрица?

```
x <- 1:6  
dim(x)
```

NULL

Что такое матрица?

```
x <- 1:6  
dim(x)
```

NULL

```
m1 <- matrix(1:6, nrow=2, ncol=3)  
m1
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

```
dim(m1) # 2 строки и 3 столбца
```

```
[1] 2 3
```

Матрицы

- Внешне похожи на таблицу
- Вектор с атрибутом `dim`
- Все элементы одного типа
- Математические операции работают быстрее

Способы создания матрицы

```
m1 <- matrix(1:6, nrow=2, ncol=3)  
m1
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

Способы создания матрицы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
```

```
m1
```

```
m2 <- matrix(1:6, nrow=2, ncol=3, byrow=T)
```

```
m2
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

```
      [,1] [,2] [,3]  
[1,]    1    2    3  
[2,]    4    5    6
```

Способы создания матрицы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m2 <- matrix(1:6, nrow=2, ncol=3, byrow=T)
m2
as.matrix(1:3)
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

```
      [,1]
[1,]    1
[2,]    2
[3,]    3
```

Способы создания матрицы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m2 <- matrix(1:6, nrow=2, ncol=3, byrow=T)
m2
as.matrix(1:3)
v <- 1:6
dim(v)
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

```
      [,1]
[1,]    1
[2,]    2
[3,]    3
```

```
NULL
```


Способы создания матрицы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m2 <- matrix(1:6, nrow=2, ncol=3, byrow=T)
m2
as.matrix(1:3)
v <- 1:6
dim(v)
dim(v) = c(2,3)
v
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

```
      [,1]
[1,]    1
[2,]    2
[3,]    3
```

NULL

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Способы создания матрицы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m2 <- matrix(1:6, nrow=2, ncol=3, byrow=T)
m2
as.matrix(1:3)
v <- 1:6
dim(v)
dim(v) = c(2,3)
v
is.matrix(v)
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

```
      [,1]
[1,]    1
[2,]    2
[3,]    3
```

NULL

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
[1] TRUE
```

Способы создания матрицы

```
a <- c(1,2,3)  
b <- c(4,5,6)
```

Способы создания матрицы

```
a <- c(1,2,3)
b <- c(4,5,6)
rbind(a,b)
```

```
 [,1] [,2] [,3]
a    1    2    3
b    4    5    6
```

Способы создания матрицы

```
a <- c(1,2,3)
b <- c(4,5,6)
rbind(a,b)
cbind(a,b)
```

```
      [,1] [,2] [,3]
a      1    2    3
b      4    5    6

      a b
[1,] 1 4
[2,] 2 5
[3,] 3 6
```

Способы создания матрицы

```
a <- c(1,2,3)
b <- c(4,5,6)
rbind(a,b)
cbind(a,b)
```

```
      [,1] [,2] [,3]
a      1   2   3
b      4   5   6

      a b
[1,] 1 4
[2,] 2 5
[3,] 3 6
```

Матрицы

Срезы

```
m1 <- matrix(1:6, nrow=2, ncol=3)  
m1
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

Матрицы

Срезы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
```

```
m1
```

```
m1[2,3] # взять элемент на пересечении 2 строки  
и 3 столбца
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6  
  
[1] 6
```


Матрицы

Срезы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m1[2,3] # взять элемент на пересечении 2 строки
        и 3 столбца
m1[2,] # взять вторую строку
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
[1] 6
```

```
[1] 2 4 6
```

Матрицы

Срезы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
```

```
m1
```

```
m1[2,3] # взять элемент на пересечении 2 строки  
и 3 столбца
```

```
m1[2,] # взять вторую строку
```

```
m1[,3] # взять третий столбец
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

```
[1] 6
```

```
[1] 2 4 6
```

```
[1] 5 6
```

Матрицы

Срезы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m1[2,3] # взять элемент на пересечении 2 строки
        и 3 столбца
m1[2,]  # взять вторую строку
m1[,3]  # взять третий столбец
m1[,3, drop = F] # взять третий столбец в виде
                матрицы
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
[1] 6
```

```
[1] 2 4 6
```

```
[1] 5 6
```

```
      [,1]
[1,]    5
[2,]    6
```

Матрицы

Срезы

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m1[2,3] # взять элемент на пересечении 2 строки
        и 3 столбца
m1[2,]  # взять вторую строку
m1[,3]  # взять третий столбец
m1[,3, drop = F] # взять третий столбец в виде
                 матрицы
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
[1] 6
```

```
[1] 2 4 6
```

```
[1] 5 6
```

```
      [,1]
[1,]    5
[2,]    6
```

Матрицы

Изменение значений

```
m1 <- matrix(1:6, nrow=2, ncol=3)  
m1
```

```
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

Матрицы

Изменение значений

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m1[2,3] = 9
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Матрицы

Изменение значений

```
m1 <- matrix(1:6, nrow=2, ncol=3)
m1
m1[2,3] = 9
m1
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    9
```

Матрицы

Создать пустую матрицу

```
m1 <- matrix(nrow=2, ncol=3)  
m1
```

```
      [,1] [,2] [,3]  
[1,]  NA  NA  NA  
[2,]  NA  NA  NA
```


Матрицы

```
m1 <- matrix(1:6,nrow=2, ncol=3)
m1
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Матрицы

```
m1 <- matrix(1:6,nrow=2, ncol=3)
```

```
m1
```

```
rownames(m1)
```

```
colnames(m1)
```

```
      [,1] [,2] [,3]
```

```
[1,]    1    3    5
```

```
[2,]    2    4    6
```

```
NULL
```

```
NULL
```

Матрицы

```
m1 <- matrix(1:6,nrow=2, ncol=3)
m1
rownames(m1)
colnames(m1)
rownames(m1) = c('r1','r2')
colnames(m1) = c('c1','c2','c3')
m1
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

NULL

NULL

```
      c1 c2 c3
r1    1  3  5
r2    2  4  6
```

Матрицы

```
m1 <- matrix(1:6,nrow=2, ncol=3)
m1
rownames(m1)
colnames(m1)
rownames(m1) = c('r1','r2')
colnames(m1) = c('c1','c2','c3')
m1
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
NULL
```

```
NULL
```

```
      c1 c2 c3
r1    1  3  5
r2    2  4  6
```

Получение справочной информации

Справки и помощь

```
x <- c(5,16,8,32,56,2)
typeof(x)
```

```
[1] "double"
```

Справки и помощь

```
x <- c(5,16,8,32,56,2)
```

```
typeof(x)
```

```
is.vector(x)
```

```
[1] "double"
```

```
[1] TRUE
```

Справки и помощь

```
x <- c(5,16,8,32,56,2)
```

```
typeof(x)
```

```
is.vector(x)
```

```
is.logical(x)
```

```
[1] "double"
```

```
[1] TRUE
```

```
[1] FALSE
```


Справки и помощь

```
x <- c(5,16,8,32,56,2)
```

```
typeof(x)
```

```
is.vector(x)
```

```
is.logical(x)
```

```
is.integer(x)
```

```
[1] "double"
```

```
[1] TRUE
```

```
[1] FALSE
```

```
[1] FALSE
```

- ?seq
- help(seq)
- example(seq)

[Cheat Sheets](#)

