

Основы программирования в R

Алла Тамбовцева

Практикум 1: ввод и вывод, векторы

Часть 1: ввод и вывод

Задача 1

Напишите код, который запрашивает у пользователя его имя с клавиатуры и выводит на экран сообщение вида "Hello, name!".

```
name <- readline(prompt = "Enter your name: ")
```

```
# 1  
cat("Hello, ", name, "!", sep = "")
```

```
## Hello, Alla!
```

```
# 2  
cat(sprintf("Hello, %s!", name))
```

```
## Hello, Alla!
```

Задача 2

Напишите код, который запрашивает у пользователя два дробных числа (одно число — один запрос) и выводит на экран их сумму. Решите эту задачу в предположении, что:

- пользователь вводит дробные числа с десятичным разделителем в виде точки;
- пользователь вводит дробные числа с десятичным разделителем в виде запятой.

```
# dots  
x1 <- as.numeric(readline("Enter a number: "))  
x2 <- as.numeric(readline("Enter a number: "))
```

```
cat(x1 + x2)
```

```
## 13.3
```

```
# commas  
a_str <- gsub(",", ".",  
             readline("Enter a number: "))  
b_str <- gsub(",", ".",  
             readline("Enter a number: "))
```

```
a <- as.numeric(a_str)  
b <- as.numeric(b_str)  
cat(a + b)
```

```
## 6.3
```

Задача 3

Напишите код, который запрашивает у пользователя значение скорости в километрах в час и выводит на экран значение скорости в метрах в секунду.

```
vel <- as.numeric(readline("Enter km/h:"))
```

```
cat(vel * 1000 / 3600)
```

```
## 13.88889
```

Часть 2: векторы

Задача 1

Дан вектор `g`, в котором хранятся следующие значения:

```
1, 0, 2, 3, 6, 8, 12, 15, 0, NA, NA, 9, 4, 16, 2, 0
```

Создайте вектор `g`. Используя `R`, выведите на экран:

- первый элемент вектора;
- последний элемент вектора;
- элементы вектора с третьего по пятый включительно;
- элементы вектора, которые равны 2;
- элементы вектора, которые больше 4;
- элементы вектора, которые кратны 3 (делятся на 3 без остатка);
- элементы вектора, которые больше 4 и кратны 3;
- элементы вектора, которые или меньше 1, или больше 5;
- индексы элементов, которые равны 0;
- индексы элементов, которые не меньше 2 и не больше 8.

```
g <- c(1, 0, 2, 3, 6, 8, 12, 15, 0, NA, NA, 9, 4, 16, 2, 0)
```

```
# 1  
g[1]
```

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

```
# 2  
g[-1] #
```

```
## [1] 0 2 3 6 8 12 15 0 NA NA 9 4 16 2 0
```

```
g[length(g)]
```

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

```
# 3  
g[3:5]
```

```
## [1] 2 3 6
```

```
# 4  
g[g == 2]
```

```
## [1] 2 NA NA 2
```

```

# 5
g[g > 4]

## [1] 6 8 12 15 NA NA 9 16

# 6
g[g %% 3 == 0]

## [1] 0 3 6 12 15 0 NA NA 9 0

# 7
g[g > 4 & g %% 3 == 0]

## [1] 6 12 15 NA NA 9

# 8
g[g < 1 | g > 5]

## [1] 0 6 8 12 15 0 NA NA 9 16 0

# 9
which(g == 0)

## [1] 2 9 16

# 10
which(g >= 2 & g <= 8)

## [1] 3 4 5 6 13 15

```

Задача 2

Напишите код, который заменяет последний элемент любого вектора на пропущенное значение (NA). Ваш код должен работать для вектора любой длины.

```

g[length(g)] <- NA
g

## [1] 1 0 2 3 6 8 12 15 0 NA NA 9 4 16 2 NA

```

Задача 3

Напишите код, который выводит на экран индексы пропущенных значений в векторе.

```

which(is.na(g))

## [1] 10 11 16

```

Задача 4

Напишите код, который считает, сколько пропущенных значений в векторе.

```

# 1
length(which(is.na(g)))

## [1] 3

```

```
# 2
sum(is.na(g))
```

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

Задача 5

Напишите код, который позволяет создать вектор из id (уникальных номеров) респондентов, если известно, что в опросе участвовало 100 респондентов.

```
ids <- 1:100
ids
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
## [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
## [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
## [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
## [91] 91 92 93 94 95 96 97 98 99 100
```

Задача 6

Решение задачи предполагает рациональное решение с помощью функций R, не нужно создавать вектора перечислением большого числа элементов вручную.

Известно, что в таблице хранятся показатели по 3 странам за 5 лет (см. ниже).

Создайте вектор который мог бы послужить столбцом `country` в этой таблице. Создайте вектор, который мог бы послужить столбцом `year` в этой таблице.

Таблица выглядит так:

n	country	year
1	France	2000
2	France	2001
3	France	2002
4	France	2003
5	France	2004
6	Italy	2000
7	Italy	2001
8	Italy	2002
9	Italy	2003
10	Italy	2004
11	Spain	2000
12	Spain	2001
13	Spain	2002
14	Spain	2003
15	Spain	2004

```
country <- rep(c("France", "Italy", "Spain"), each = 5)
year <- rep(2000:2004, 3)
dat <- cbind.data.frame(country, year)
dat
```

```
## country year
```

```
## 1 France 2000
## 2 France 2001
## 3 France 2002
## 4 France 2003
## 5 France 2004
## 6 Italy 2000
## 7 Italy 2001
## 8 Italy 2002
## 9 Italy 2003
## 10 Italy 2004
## 11 Spain 2000
## 12 Spain 2001
## 13 Spain 2002
## 14 Spain 2003
## 15 Spain 2004
```

Задача 7 Исследователь решил создать вектор `income`, в котором сохранены доходы нескольких респондентов:

```
income <- c(10000, 32000, 28000, 150000, 65000, 1573)
```

Исследователю нужно получить вектор `income_class`, состоящий из 0 и 1: 0 ставится, если доход респондента ниже среднего дохода, а 1 — если больше или равен среднему доходу. Создайте вектор `income_class`.

Подсказка. Сначала можно посчитать среднее значение по вектору `income` и сохранить его в какую-нибудь переменную. Пользоваться встроенной функцией `mean()` нельзя.

```
m <- sum(income) / length(income)
income_class <- as.integer(income >= m)
income_class
```

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