

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**  
**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №1**  
**Дисциплина: Обработка больших данных**

Работу выполнил: \_\_\_\_\_ А. А. Костров

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_ А. А. Яхонтов

Краснодар  
2026

## Тема. Введение в язык R.

Цель. Ознакомление с синтаксисом языка R для анализа данных.

1. Создадим вектор  $p$ , состоящий из значений 7, 6, 5, 4, и вектор  $q$ , состоящий из 0, 1, 2, 3, произведём сложение, вычитание, умножение, деление, возведение в степень этих векторов (рисунок 1).

The screenshot shows the RStudio interface. In the top-left pane, there is a code editor with the following R script:

```
1 p <- 7:4
2 q <- 0:3
3
4 sum_result <- p + q
5 sub_result <- p - q
6 mul_result <- p * q
7 div_result <- p / q
8 pow_result <- p ^ q
9
```

In the top-right pane, the "Environment" tab is selected, showing the global environment with variables  $p$ ,  $q$ ,  $div\_result$ ,  $pow\_result$ ,  $mul\_result$ ,  $sub\_result$ , and  $sum\_result$ . The values are listed as follows:

Name	Type	Length	Size	Value
$p$	integer	4	696 B	int [1:4] 7 6 5 4
$q$	integer	4	696 B	int [1:4] 0 1 2 3
$div\_result$	numeric	4	80 B	num [1:4] Inf 6 2.5 1.33
$pow\_result$	numeric	4	80 B	num [1:4] 1 6 25 64
$mul\_result$	integer	4	64 B	int [1:4] 0 6 10 12
$sub\_result$	integer	4	64 B	int [1:4] 7 5 3 1
$sum\_result$	integer	4	64 B	int [1:4] 7 7 7 7

In the bottom-left pane, the "Console" tab is active, showing the command history and the execution of the script:

```
>
>
>
>
>
>
>
>
> > source("~/Documents/kubsu-files/sementer6/big-data/practice/lab1_intro/general_task.R")
> |
```

Рисунок 1

2. Создадим вектор  $0, 2, 0, 4, \dots, 18, 0, 20$  наиболее коротким способом. Создадим вектор, в котором будут содержаться первые 20 степеней двойки. Создадим вектор из чисел 1, 10, 100, 1000, 10000 (рисунок 2).

The screenshot shows the RStudio interface. In the top-left pane, there is a code editor with the following R script:

```
1 # Task 1
2 # p <- 7:4
3 # q <- 0:3
4 #
5 # sum_result <- p + q
6 # sub_result <- p - q
7 # mul_result <- p * q
8 # div_result <- p / q
9 # pow_result <- p ^ q
10
11 alt_result <- (1:20) * c(0, 1)
12 two_result <- 2 ** (1:20)
13 ten_result <- 10 ** (0:4)
14
```

In the top-right pane, the "Environment" tab is selected, showing the global environment with variables  $alt\_result$ ,  $two\_result$ , and  $ten\_result$ . The values are listed as follows:

Name	Type	Length	Size	Value
$alt\_result$	numeric	20	208 B	num [1:20] 0 2 0 4 0 6 0 8 0 10 ...
$two\_result$	numeric	20	208 B	num [1:20] 2 4 8 16 32 ...
$ten\_result$	numeric	5	96 B	num [1:5] 1 10 100 1000 10000

In the bottom-left pane, the "Console" tab is active, showing the command history and the execution of the script:

```
14:1 (Top Level) <--> R Script <--> Console
```

Рисунок 2

3. С помощью функции *sum* посчитаем сумму последовательностей. Выведем число значений в крайней последовательности, которые больше 0,5 (рисунок 3).

The screenshot shows the RStudio interface. The left pane displays the code for 'general\_task.R':

```

1 # Task 1
2 # p <- 7:4
3 # q <- 0:3
4 #
5 # sum_result <- p + q
6 # sub_result <- p - q
7 # mul_result <- p * q
8 # div_result <- p / q
9 # pow_result <- p ^ q
10
11 # Task 2
12 # alt_result <- (1:20) * c(0, 1)
13 # two_result <- 2 ** (1:20)
14 # ten_result <- 10 *** (0:4)
15
16 tel_result <- sum( (1 / (1:50)) * (1 / (2:51)) )
17 inv_result <- sum(1 / 2 ** (0:20))
18
19 cmb <- seq(1, 28, 3) / 3 ** (0:9)
20 cmb_result <- sum(cmb)
21
22 print(sum(cmb > 0.5))
23

```

The right pane shows the 'Environment' tab with the following variables:

Name	Type	Length	Size	Value
tel_result	numeric	1	56 B	0.980392156862745
inv_result	numeric	1	56 B	1.99999904632568
cmb_result	numeric	1	56 B	3.7491744146934
cmb	numeric	10	176 B	num [1:10] 1 1.333 0.778 0.37 0 ...

The bottom pane shows the file browser with a folder named 'RStudioProjects'.

Рисунок 3

4. Создадим вектор *vec3* со значениями 3, 6, 9, ..., 24, 27 с помощью функции *seq* с нужными параметрами. Извлечём 2-ое, 5-ое и 7-ое значения вектора, извлечём предпоследнее значение вектора (используем функцию *length*), а затем все, кроме предпоследнего. Вернём все элементы вектора, кроме шестого. Попробуем извлечь сотовое значение вектора. Вернём все значения вектора *vec3*, кроме первого и последнего. Найдём все значения вектора *vec3*, которые больше 4, но меньше 10. Найдём все значения вектора *vec3*, которые меньше 4 или больше 10 (рисунок 4).

The screenshot shows the RStudio interface. The left pane displays the code for 'general\_task.R':

```

25 vec3 <- seq(3, 27, by = 3)
26 print(vec3[2, 5, 7])
27
28 print(vec3[length(vec3) - 1])
29
30 print(vec3[ -(length(vec3) - 1) ])
31
32 print(vec3[-6])
33
34 print(vec3[100])
35
36 print( vec3[ -c(1, length(vec3)) ] )
37
38 print(vec3[vec3 > 4 & vec3 < 10])
39
40 print(vec3[vec3 < 4 | vec3 > 10])
41

```

The right pane shows the 'Environment' tab with the variable:

Name	Type	Length	Size	Value
vec3	numeric	9	176 B	num [1:9] 3 6 9 12 15 18 21 24 ...

The bottom pane shows the file browser with a folder named 'RStudioProjects'.

Рисунок 4

5. Создадим случайный вектор *vect* натуральных чисел от 1 до 100 и определим, сколько из них кратны любому из чисел в диапазоне от 2 до 5. Числа для определения кратности вводим с клавиатуры (рисунок 5).

The screenshot shows the RStudio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Left Panel:** A code editor window titled "variant\_task.R\*" containing R code. The code generates a vector of 100 random integers, reads a string from standard input, scans it for integers, and then checks if any of the divisors in that range (2 to 5) are factors of any number in the vector. It prints the count of such numbers.
- Environment Tab:** Shows the global environment with variables like vect, result, is\_multiple, etc., and their values.
- Console Tab:** Shows the R session output. It starts with the R version (R 4.5.2), loads the script, and prints the result (3 4).
- File Explorer:** Shows the file structure under "Home".

Рисунок 5

6. Высотная поясность на северном склоне Западного Кавказа, согласно Большой Российской энциклопедии устроена следующим образом:

- до 500 м — степь и лесостепь
- до 800 м — низкогорные широколиственные леса (дуб, граб)
- до 1300 м — среднегорные широколиственные леса (бук)
- до 1600 м — смешанные леса (ель, пихта, бук)
- до 2300 м — криволесия (береза, бук, клён)
- до 2500 м — субальпийские и альпийские луга
- до 3300 м — субнивальная зона (фрагментарная растительность)
- выше (условно до 5000 м) — гляциально-нивальная зона

Создадим фрейм данных, включающий три столбца: минимальная высота пояса (*Hmin*), максимальная высота пояса (*Hmax*) и название высотного пояса (*Zone* – закодируем для краткости). Минимальную высоту надо вычислить на основе максимальной, приняв, что для нижнего пояса она условно равна 400 м. Напишем программу, которая просит пользователя ввести высоту и возвращает соответствующий введенной высоте (рисунок 6).

The screenshot shows the RStudio interface with the following components:

- Script Editor (variant\_task.R):**

```

24
25 zones_df <- data.frame(
26   Hmax = c(500, 800, 1300, 1600, 2300, 2500, 3300, 5000),
27   Zone = c("Стель", "Дуб", "Бук", "Ель", "Кривол", "Луг", "Субнив", "Нивал")
28 )
29 zones_df$Hmin <- c(400, zones_df$Hmax[-nrow(zones_df)] + 1)
30
31 cat("Введите высоту (м): ")
32 h <- as.numeric(readline())
33 if (!is.na(h)) {
34   zone <- zones_df[h >= zones_df$Hmin & h <= zones_df$Hmax, ]
35   if (nrow(zone) > 0) print(zone) else cat("Высота вне диапазона\n")
36 }
37

```
- Environment View:** Shows the global environment with objects: zones\_df (data.frame), zone (data.frame), and h (numeric).
- Console View:** Displays the R session history, including the execution of the script and user input.
- File Browser (Files View):** Shows the local directory structure under "Home".

Рисунок 6