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**

**Отчёт лабораторная №1**

**Дисциплина: ОБРАБОТКА БОЛЬШИХ ДАННЫХ**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Арутюнян Р.Г.

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Яхонтов А.А.

Краснодар

2025

1.8. Задания к лабораторной работе

1. Создайте вектор p, состоящий из значений 7, 6, 5, 4, и вектор q, состоящий из 0, 1, 2, 3, произведите сложение, вычитание, умножение, деление, возведение в степень этих векторов.

# Создаем векторы p и q

p <- c(7, 6, 5, 4)

q <- c(0, 1, 2, 3)

# Выполняем арифметические операции

addition <- p + q # Сложение

subtraction <- p - q # Вычитание

multiplication <- p \* q # Умножение

division <- p / q # Деление (обрабатывает деление на 0 как Inf)

exponentiation <- p ^ q # Возведение в степень

# Вывод результатов

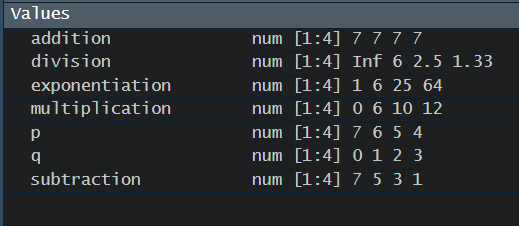
print(addition)

print(subtraction)

print(multiplication)

print(division)

print(exponentiation)



1. Создайте вектор 0, 2, 0, 4, … , 18, 0, 20 наиболее которким спосбом. Создайте вектор, в котором будут содержаться первые 20 степеней двойки. Создайте вектор из чисел 1, 10, 100, 1000, 10000.

# 1. Создание вектора 0, 2, 0, 4, ..., 18, 0, 20

vec1 <- c(rbind(rep(0, 11), seq(2, 20, by = 2)))

print(vec1)

# 2. Создание вектора с первыми 20 степенями двойки

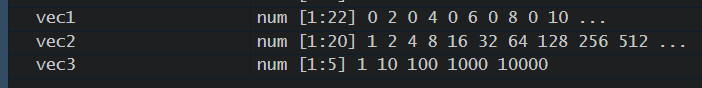
vec2 <- 2^(0:19)

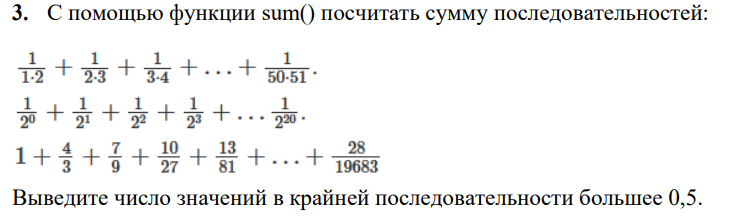
print(vec2)

# 3. Создание вектора из чисел 1, 10, 100, 1000, 10000

vec3 <- 10^(0:4)

print(vec3)





# 1. Сумма первой последовательности: 1/(1\*2) + 1/(2\*3) + ... + 1/(50\*51)

seq1 <- 1 / (1:50 \* 2:51)

sum\_seq1 <- sum(seq1)

print(sum\_seq1)

# 2. Сумма второй последовательности: 1/2^0 + 1/2^1 + ... + 1/2^20

seq2 <- 1 / 2^(0:20)

sum\_seq2 <- sum(seq2)

print(sum\_seq2)

# 3. Сумма третьей последовательности: 1 + 4/3 + 7/9 + ... + 28/19683

# Числитель — последовательность 1, 4, 7, 10, ..., 28

numerator <- seq(1, 28, by = 3)

# Знаменатель — последовательность 3^0, 3^1, ..., 3^6

denominator <- 3^(0:(length(numerator) - 1))

# Вектор последовательности

seq3 <- numerator / denominator

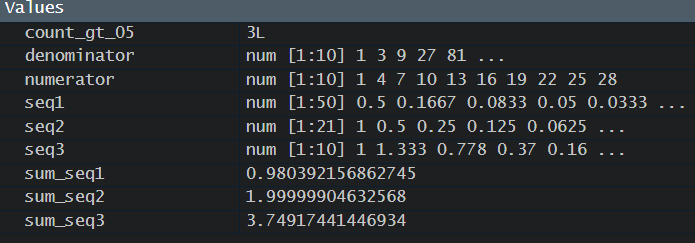
sum\_seq3 <- sum(seq3)

print(sum\_seq3)

# Количество значений в третьей последовательности больше 0.5

count\_gt\_05 <- sum(seq3 > 0.5)

print(count\_gt\_05)



4. Создайте вектор vec3 со значениями 3, 6, 9, …, 24, 27 с помощью функции seq() с нужными параметрами. Извлеките 2, 5 и 7 значения вектора, извлеките предпоследнее значение вектора (использовать функцию length()), а затем все, кроме предпоследнего. Верните все элементы вектора, кроме шестого. Попробуйте извлечь сотое значение вектора. Верните все значения вектора vec3 кроме первого и последнего. Найдите все значения вектора vec3, которые больше 22 4, но меньше 10. Найдите все значения вектора vec3, которые меньше 4 или больше 10.

# 1. Создаём вектор vec3 с использованием seq()

vec3 <- seq(3, 27, by = 3)

print(vec3)

# 2. Извлекаем 2-е, 5-е и 7-е значения

extracted\_values <- vec3[c(2, 5, 7)]

print(extracted\_values)

# 3. Извлекаем предпоследнее значение вектора

penultimate <- vec3[length(vec3) - 1]

print(penultimate)

# 4. Извлекаем все, кроме предпоследнего

all\_but\_penultimate <- vec3[-(length(vec3) - 1)]

print(all\_but\_penultimate)

# 5. Извлекаем все значения, кроме шестого

all\_but\_sixth <- vec3[-6]

print(all\_but\_sixth)

# 6. Пытаемся извлечь сотое значение

hundredth\_value <- vec3[100]

print(hundredth\_value) # Возвращает NA, так как индекса нет

# 7. Возвращаем все значения, кроме первого и последнего

all\_but\_first\_last <- vec3[-c(1, length(vec3))]

print(all\_but\_first\_last)

# 8. Все значения больше 4, но меньше 10

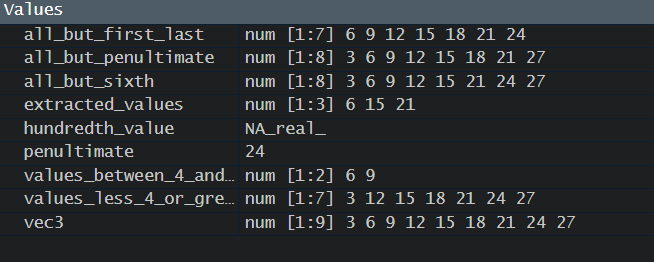
values\_between\_4\_and\_10 <- vec3[vec3 > 4 & vec3 < 10]

print(values\_between\_4\_and\_10)

# 9. Все значения меньше 4 или больше 10

values\_less\_4\_or\_greater\_10 <- vec3[vec3 < 4 | vec3 > 10]

print(values\_less\_4\_or\_greater\_10)



1.9 Задания к лабораторной работе(6,17)

6. Напишите код, который позволяет создать вектор из id (уникальных номеров) респондентов, если известно, что в опросе участвовало 100 респондентов. То же задание, но известно лишь, что число респондентов <=200 и это число вводится пользователем

# Подключение пакета

library(svDialogs)

# Ввод значения через диалоговое окно

input <- dlgInput("Введите количество респондентов (не больше 200):")$res

# Проверяем, является ли введённое значение числом

if (!is.na(as.numeric(input)) && as.numeric(input) > 0) {

num\_respondents <- as.integer(input)

# Проверяем, что число респондентов <= 200

if (num\_respondents <= 200) {

id\_vector <- 1:num\_respondents

print(id\_vector)

} else {

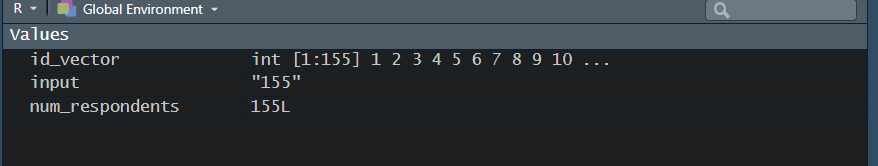
print("Ошибка: число респондентов должно быть не больше 200.")

}

} else {

print("Ошибка: введите корректное число.")

}



17. Дан вектор случайных чисел vес, длиной 80. Найти в v те элементы, значение которых больше среднего арифметического, взятого от всех элементов ветора v. Сформировать из v матрицу 8х10 и вывести на экран веторы четных и нечетных столбцов.

# 1. Генерация вектора случайных чисел длиной 80

set.seed(123) # Для воспроизводимости(фиксирует генератор случайных чисел (чтобы при повторном запуске получались те же числа).

vec <- runif(80, min = 0, max = 100) # Случайные числа от 0 до 100

print(vec)

# 2. Вычисление среднего арифметического

mean\_value <- mean(vec)

# 3. Нахождение элементов, превышающих среднее арифметическое

greater\_than\_mean <- vec[vec > mean\_value]

print("Элементы больше среднего арифметического:")

print(greater\_than\_mean)

# 4. Формирование матрицы 8x10

matrix\_8x10 <- matrix(vec, nrow = 8, ncol = 10, byrow = FALSE)

print("Матрица 8x10:")

print(matrix\_8x10)

# 5. Извлечение четных и нечетных столбцов

even\_columns <- matrix\_8x10[, seq(2, ncol(matrix\_8x10), by = 2)]

odd\_columns <- matrix\_8x10[, seq(1, ncol(matrix\_8x10), by = 2)]

# 6. Вывод четных и нечетных столбцов

print("Четные столбцы матрицы:")

print(even\_columns)

print("Нечетные столбцы матрицы:")

print(odd\_columns)

