

Отчёт по лабораторной работе №3 по дисциплине Компьютерный практикум по статистическому анализу данных

Управляющие структуры

Шаповалова Диана Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Циклы while и for	6
2.2	Условные выражения	7
2.3	Функции	8
2.4	Сторонние библиотеки (пакеты) в Julia	9
2.5	Задания для самостоятельного выполнения	10
3	Выводы	19

Список иллюстраций

2.1	Выполняем примеры по циклам while и for	7
2.2	Выполняем примеры по условным выражениям	8
2.3	Выполняем примеры по функциям и сторонним библиотекам . . .	10
2.4	Задание №1	11
2.5	Задание №1	12
2.6	Задание №2 и 3	12
2.7	Задание №4	13
2.8	Задание №6	13
2.9	Задание №7	14
2.10	Задание №8	14
2.11	Задание №8	15
2.12	Задание №9	16
2.13	Задание №10	16
2.14	Задание №10	17
2.15	Задание №11	18

Список таблиц

1 Цель работы

Основная цель работы — освоить применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

2 Выполнение лабораторной работы

2.1 Циклы while и for

Для различных операций, связанных с перебором индексируемых элементов структур данных, традиционно используются циклы while и for. Например, while можно использовать для формирования элементов массива:

Другой пример демонстрирует использование while при работе со строковыми элементами массива, подставляя имя из массива в заданную строку приветствия и выводя получившуюся конструкцию на экран

3.2.1 Циклы while и for

[1]: *# пока n<10 прибавить к n единицу и распечатать значение:*

```
n = 0
while n < 10
  n += 1
  println(n)
end
```

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

[2]: `myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]`

```
i = 1
while i <= length(myfriends)
  friend = myfriends[i]
  println("Hi $friend, it's great to see you!")
  i += 1
end
```

```
Hi Ted, it's great to see you!
Hi Robyn, it's great to see you!
Hi Barney, it's great to see you!
Hi Lily, it's great to see you!
Hi Marshall, it's great to see you!
```

Рис. 2.1: Выполняем примеры по циклам while и for

2.2 Условные выражения

Довольно часто при решении задач требуется проверить выполнение тех или иных условий. Для этого используют условные выражения.

Например, пусть для заданного числа x требуется вывести слово «Fizz», если x делится на 3, «Buzz», если x делится на 5, и «FizzBuzz», если x делится на 3 и 5

3.2.2. Условные выражения

```
[8]: N = 3
      # используем `&&` для реализации операции "AND"
      # операция % вычисляет остаток от деления
      if (N % 3 == 0) && (N % 5 == 0)
        println("FizzBuzz")
      elseif N % 3 == 0
        println("Fizz")
      elseif N % 5 == 0
        println("Buzz")
      else
        println(N)
      end
```

Fizz

Рис. 2.2: Выполняем примеры по условным выражениям

2.3 Функции

Julia дает нам несколько разных способов написать функцию. Первый требует ключевых слов `function` и `end`.

В Julia функция `map` является функцией высшего порядка, которая принимает функцию в качестве одного из своих входных аргументов и применяет эту функцию к каждому элементу структуры данных, которая ей передаётся также в качестве аргумента.

Функция `broadcast` — ещё одна функция высшего порядка в Julia, представляющая собой обобщение функции `map`. Функция `broadcast()` будет пытаться привести все объекты к общему измерению, `map()` будет напрямую применять данную функцию поэлементно. Синтаксис для вызова `broadcast` такой же, как и для вызова `map`.

2.4 Сторонние библиотеки (пакеты) в Julia

Julia имеет более 2000 зарегистрированных пакетов, что делает их огромной частью экосистемы Julia. Есть вызовы функций первого класса для других языков, обеспечивающие интерфейсы сторонних функций. Можно вызвать функции из Python или R, например, с помощью PyCall или Rcall.

Например, добавим и загрузим пакет Colors, затем создадим палитру из 100 разных цветов, а затем определим матрицу 3×3 с элементами в форме случайного цвета из палитры, используя функцию rand.

3.2.3. Функции

```
[9]: # Задаём матрицу A:
A = [i + 3*j for j in 0:2, i in 1:3]

[9]: 3x3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

[13]: # Вызываем функцию f возведения в квадрат
f(x) = x^2
f(A)

[13]: 3x3 Matrix{Int64}:
30  36  42
66  81  96
102 126 150

[14]: B = f.(A)

[14]: 3x3 Matrix{Int64}:
 1  4  9
16 25 36
49 64 81

[35]: using Colors
palette = distinguishable_colors(100)
rand(palette, 3, 3)

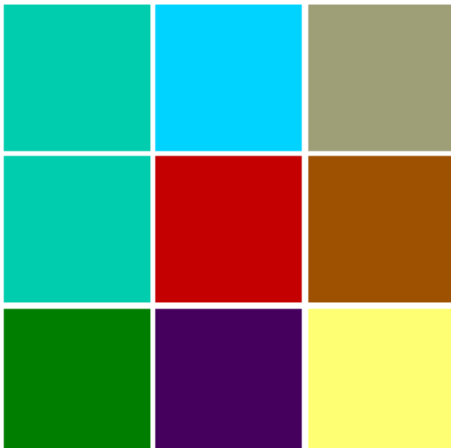
[35]: 
```

Рис. 2.3: Выполняем примеры по функциям и сторонним библиотекам

2.5 Задания для самостоятельного выполнения

1. Используя циклы while и for:

– выведите на экран целые числа от 1 до 100 и напечатайте их квадраты;

3.4. Задания для самостоятельного выполнения

1. Используя циклы while и for:

– выведите на экран целые числа от 1 до 100 и напечатайте их квадраты;

```
[37]: for i in 1:100  
      println("Число: $i, Квадрат: $(i^2)")  
    end
```

```
Число: 1, Квадрат: 1  
Число: 2, Квадрат: 4  
Число: 3, Квадрат: 9  
Число: 4, Квадрат: 16  
Число: 5, Квадрат: 25  
Число: 6, Квадрат: 36  
Число: 7, Квадрат: 49  
Число: 8, Квадрат: 64  
Число: 9, Квадрат: 81  
Число: 10, Квадрат: 100  
Число: 11, Квадрат: 121  
Число: 12, Квадрат: 144  
Число: 13, Квадрат: 169  
Число: 14, Квадрат: 196  
Число: 15, Квадрат: 225  
Число: 16, Квадрат: 256  
Число: 17, Квадрат: 289  
Число: 18, Квадрат: 324  
Число: 19, Квадрат: 361  
Число: 20, Квадрат: 400  
Число: 21, Квадрат: 441  
Число: 22, Квадрат: 484  
Число: 23, Квадрат: 529  
Число: 24, Квадрат: 576  
Число: 25, Квадрат: 625  
Число: 26, Квадрат: 676  
Число: 27, Квадрат: 729  
Число: 28, Квадрат: 784  
Число: 29, Квадрат: 841  
Число: 30, Квадрат: 900  
Число: 31, Квадрат: 961  
Число: 32, Квадрат: 1024  
Число: 33, Квадрат: 1089
```

Рис. 2.4: Задание №1

- создайте словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;
- создайте массив squares_arr, содержащий квадраты всех чисел от 1 до 100.

– создайте словарь `squares`, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;

```
[38]: squares = Dict{  
      Dict{Any, Any}(5 => 25, 56 => 3136, 35 => 1225, 55 => 3025, 60 => 3600, 30 => 900, 32 => 1024, 6 => 36, 67 => 4489, 45 => 2025, 73 => 5329, 64 => 4096, 90 => 8100, 4 => 16, 13 => 169, 54 => 2916, 63 => 3969, 86 => 7396, 91 => 8281, 62 => 3844, 58 => 3364, 52 => 2704, 12 => 144, 28 => 784, 75 => 5625, 23 => 529, 92 => 8464, 41 => 1681, 43 => 1849, 11 => 121, 36 => 1296, 68 => 4624, 69 => 4761, 98 => 9604, 82 => 6724, 85 => 7225, 39 => 1521, 84 => 7056, 77 => 5929, 7 => 49, 25 => 625, 95 => 9025, 71 => 5041, 66 => 4356, 76 => 5776, 34 => 1156, 50 => 2500, 59 => 3481, 93 => 8649, 2 => 4, 10 => 100, 18 => 324, 26 => 676, 27 => 729, 42 => 1764, 87 => 7569, 100 => 10000, 79 => 6241, 16 => 256, 20 => 400, 81 => 6561, 19 => 361, 49 => 2401, 44 => 1936, 9 => 81, 31 => 961, 74 => 5476, 61 => 3721, 29 => 841, 94 => 8836, 46 => 2116, 57 => 3249, 70 => 4900, 21 => 441, 38 => 1444, 88 => 7744, 78 => 6084, 72 => 5184, 24 => 576, 8 => 64, 17 => 289, 37 => 1369, 1 => 1, 53 => 2809, 22 => 484, 47 => 2209, 83 => 6889, 99 => 9801, 89 => 7921, 14 => 196, 3 => 9, 80 => 6400, 96 => 9216, 51 => 2601, 33 => 1089, 40 => 1600, 48 => 2304, 15 => 225, 65 => 4225, 97 => 9409)  
      println(squares)
```

– создайте массив `squares_arr`, содержащий квадраты всех чисел от 1 до 100.

```
[40]: squares_arr = []  
      i = 1  
      while i <= 100  
          push!(squares_arr, i^2)  
          i += 1  
      end  
      println(squares_arr)  
Any[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961, 1024, 1089, 1156, 1225, 1296, 1369, 1444, 1521, 1600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2304, 2401, 2500, 2601, 2704, 2809, 2916, 3025, 3136, 3249, 3364, 3481, 3600, 3721, 3844, 3969, 4096, 4225, 4356, 4489, 4624, 4761, 4900, 5041, 5184, 5329, 5476, 5625, 5776, 5929, 6084, 6241, 6400, 6561, 6724, 6889, 7056, 7225, 7396, 7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9604, 9801, 10000]
```

Рис. 2.5: Задание №1

2. Напишите условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишите код, используя тернарный оператор.
3. Напишите функцию `add_one`, которая добавляет 1 к своему входу.

2. Напишите условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишите код, используя тернарный оператор.

```
[42]: function even_odd(n)  
      if n % 2 == 0  
          println(n)  
      else  
          println("Нечетное")  
      end  
      end  
      even_odd(3)  
      even_odd(4)  
      Нечетное  
      4
```

```
[44]: function even_odd_ternary(n)  
      println(n % 2 == 0 ? n : "нечётное")  
      end  
      even_odd(3)  
      even_odd(4)  
      Нечетное  
      4
```

3. Напишите функцию `add_one`, которая добавляет 1 к своему входу

```
[49]: function add_one(x)  
      return x+1  
      end  
      println(add_one(-1))  
      println(add_one(3))  
      0  
      4
```

Рис. 2.6: Задание №2 и 3

4. Используйте `map()` или `broadcast()` для задания матрицы \square , каждый элемент которой увеличивается на единицу по сравнению с предыдущим.

4. Используйте `map()` или `broadcast()` для задания матрицы A , каждый элемент которой увеличивается на единицу по сравнению с предыдущим. [1](#)

```
[57]: add_one(x) = x+1
A = reshape(1:9, 3, 3)
a = map(add_one, A)
println("Матрица A:")
println(A)
println("Матрица a:")
println(a)

Матрица A:
[1 4 7; 2 5 8; 3 6 9]
Матрица a:
[2 5 8; 3 6 9; 4 7 10]
```

Рис. 2.7: Задание №4

6. Создайте матрицу B с элементами $B_{i1} = 10$, $B_{i2} = -10$, $B_{i3} = 10$, $i = 1, 2, \dots, 15$.

Вычислите матрицу $C = B * B^T$

6. Создайте матрицу B с элементами $B_{i1} = 10$, $B_{i2} = -10$, $B_{i3} = 10$, $i = 1, 2, \dots, 15$.

Вычислите матрицу $C = (B^T) * B$

```
[73]: B = repeat([10 -10 10], 15, 1)
```

```
[73]: 15x3 Matrix{Int64}:
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
 10 -10 10
```

```
[74]: C = B' * B
```

```
[74]: 3x3 Matrix{Int64}:
 1500 -1500 1500
-1500 1500 -1500
 1500 -1500 1500
```

Рис. 2.8: Задание №6

7. Создайте матрицу A размерности 6×6 , все элементы которой равны нулю, и матрицу B , все элементы которой равны 1. Используя цикл `while` или `for` и

закономерности расположения элементов, создайте следующие матрицы размерности 6×6

7. Создайте матрицу Z размерности 6×6 , все элементы которой равны нулю, и матрицу E , все элементы которой равны 1. Используя цикл `while` или `for` и закономерности расположения элементов, создайте следующие матрицы размерности 6×6 :

```
# Функция для красивого вывода матриц
function print_matrix(mat)
    for row in eachrow(mat)
        println(row)
    end
    println()
end

Z = zeros{Int, 6, 6} # Матрица Z размерности 6x6, все элементы равны нулю

# Функция для создания матрицы Z_1
function create_Z1()
    Z1 = copy(Z) # Копируем матрицу Z
    for i in 1:6
        for j in 1:6
            if abs(i - j) == 1 || i == j
                Z1[i, j] = 1
            end
        end
    end
    return Z1
end

# Матрица Z1
Z1 = copy(Z)
for i in 1:6
    for j in 1:6
        if abs(i - j) == 1 || i == j
```

Рис. 2.9: Задание №7

8. В языке R есть функция `outer()`. Фактически, это матричное умножение с возможностью изменить применяемую операцию (например, заменить произведение на сложение или возведение в степень).

– Напишите свою функцию, аналогичную функции `outer()` языка R. Функция должна иметь следующий интерфейс: `outer(x,y,operation)`.

№8. В языке R есть функция `outer()`. Фактически, это матричное умножение с возможностью изменить применяемую операцию (например, заменить произведение на сложение или возведение в степень):

8.1) Напишите свою функцию, аналогичную функции `outer()` языка R. Функция должна иметь следующий интерфейс: `outer(x,y,operation)`:

```
function outer(x, y, operation)
    return [operation(xi, yj) for xi in x, yj in y]
end
```

outer (generic function with 1 method)

Рис. 2.10: Задание №8

– Используя написанную вами функцию `outer()`, создайте матрицы следующей структуры

Матрица A1:

```
[0, 1, 2, 3, 4]
[1, 2, 3, 4, 5]
[2, 3, 4, 5, 6]
[3, 4, 5, 6, 7]
[4, 5, 6, 7, 8]
```

Матрица A2:

```
[0, 0, 0, 0, 0]
[1, 1, 1, 1, 1]
[2, 4, 8, 16, 32]
[3, 9, 27, 81, 243]
[4, 16, 64, 256, 1024]
```

Матрица A3:

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

Матрица A4:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
[2, 3, 4, 5, 6, 7, 8, 9, 0, 1]
[3, 4, 5, 6, 7, 8, 9, 0, 1, 2]
[4, 5, 6, 7, 8, 9, 0, 1, 2, 3]
[5, 6, 7, 8, 9, 0, 1, 2, 3, 4]
[6, 7, 8, 9, 0, 1, 2, 3, 4, 5]
[7, 8, 9, 0, 1, 2, 3, 4, 5, 6]
[8, 9, 0, 1, 2, 3, 4, 5, 6, 7]
[9, 0, 1, 2, 3, 4, 5, 6, 7, 8]
```

Матрица A5:

```
[0, 8, 7, 6, 5, 4, 3, 2, 1]
[1, 0, 8, 7, 6, 5, 4, 3, 2]
[2, 1, 0, 8, 7, 6, 5, 4, 3]
[3, 2, 1, 0, 8, 7, 6, 5, 4]
[4, 3, 2, 1, 0, 8, 7, 6, 5]
[5, 4, 3, 2, 1, 0, 8, 7, 6]
[6, 5, 4, 3, 2, 1, 0, 8, 7]
```

Рис. 2.11: Задание №8

9. Решите следующую систему линейных уравнений с 5 неизвестными

▼ 9.Решите следующую систему линейных уравнений с 5 неизвестными:

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 = 7,$$

$$2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 = -1,$$

$$3x_1 + 2x_2 + x_3 + 2x_4 + 3x_5 = -3,$$

$$4x_1 + 3x_2 + 2x_3 + x_4 + 2x_5 = 5,$$

$$5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 = 17$$

```
[83]: A = [1 2 3 4 5; 2 1 2 3 4; 3 2 1 2 3; 4 3 2 1 2; 5 4 3 2 1]
      y = [7; -1; -3; 5; 17]
      x = A \ y
      println(x)

[-2.0000000000000036, 3.0000000000000058, 4.999999999999998, 1.999999999999991, -3.999999999999999]
```

Рис. 2.12: Задание №9

10. Создайте матрицу \square размерности 6×10 , элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности $1, 2, \dots, 10$.

– Найдите число элементов в каждой строке матрицы \square , которые больше числа \square (например, $\square = 4$).

– Определите, в каких строках матрицы \square число \square (например, $\square = 7$) встречается ровно 2 раза?

10. Создайте матрицу M размерности 6×10 , элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности $1, 2, \dots, 10$.

– Найдите число элементов в каждой строке матрицы M , которые больше числа N (например, $N = 4$).

```
34]: M = rand(1:10, 6, 10)

34]: 6x10 Matrix{Int64}:
      4 1 3 3 5 10 9 8 4 10
      2 5 1 4 6 4 3 5 1 7
      2 9 6 1 10 8 3 9 5 4
      8 5 5 7 4 6 3 9 7 6
      5 1 2 6 4 6 2 5 1 6
      10 8 9 3 3 4 7 7 1 5

38]: N = 4
      count = [sum(M[i, :] .> N) for i in 1:6]
      println(count)

[5, 4, 6, 8, 5, 6]

– Определите, в каких строках матрицы  $M$  число  $M$  (например,  $M = 7$ ) встречается ровно 2 раза?

39]: two_M = [1 for i in 1:6 if sum(M[i, :] .== 7) == 2]
      println(two_M)

[4, 6]
```

Рис. 2.13: Задание №10

– Определите все пары столбцов матрицы M , сумма элементов которых больше K (например, $K = 75$).

– Определите все пары столбцов матрицы M , сумма элементов которых больше K (например, $K = 75$)

```
33]: column_pairs = []
K = 75
for i in 1:10
    for j in i+1:10
        sum_columns = sum(M[:, i]) + sum(M[:, j]) # сумма элементов пары столбцов
        if sum_columns > K
            push!(column_pairs, (i, j))
        end
    end
end
println(column_pairs)

Any[(6, 8), (6, 10), (8, 10)]
```

Рис. 2.14: Задание №10

11. Вычислите

11. Вычислите: $\sum_{i=1}^{20} \sum_{j=1}^5 i^4 / (3 + j)$

```
[95]: sum_res= 0
      for i in 1:20
        inner_sum = 0
        for j in 1:5
          inner_sum += i^4 / (3+j)
        end
        sum_res += inner_sum
      end
      println(sum_res)
```

639215.2833333334

$\sum_{i=1}^{20} \sum_{j=1}^5 i^4 / (3 + ij)$

```
[96]: sum_res= 0
      for i in 1:20
        inner_sum = 0
        for j in 1:5
          inner_sum += i^4 / (3+i*j)
        end
        sum_res += inner_sum
      end
      println(sum_res)
```

89912.02146097136

Рис. 2.15: Задание №11

3 Выводы

Мы освоили применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.