

Компьютерный практикум по статистическому анализу данных

Лабораторная работа № 3. Управляющие структуры

Сунгурова Мариян Мухсиновна

Содержание

1	Введение	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Выводы	22
	Список литературы	23

Список иллюстраций

3.1	Примеры. Циклы	6
3.2	Примеры. Циклы	7
3.3	Примеры. Циклы	8
3.4	Примеры. Условия	9
3.5	Примеры. Функции	10
3.6	Примеры. Функции	11
3.7	Примеры. Функции	12
3.8	Примеры. Функции	13
3.9	Задание 1	14
3.10	Задание 1	15
3.11	Задание 2	15
3.12	Задание 3	16
3.13	Задание 4	16
3.14	Задание 5	16
3.15	Задание 5	17
3.16	Задание 6	17
3.17	Задание 7	18
3.18	Задание 7	18
3.19	Задание 7	19
3.20	Задания 8	19
3.21	Задания 8	20
3.22	Задания 9	20
3.23	Задание 10	21
3.24	Задание 11	21

1 Введение

Цель работы

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

Задачи

1. Используя Jupyter Lab, повторите примеры из раздела 3.2.
2. Выполните задания для самостоятельной работы (раздел 3.4).

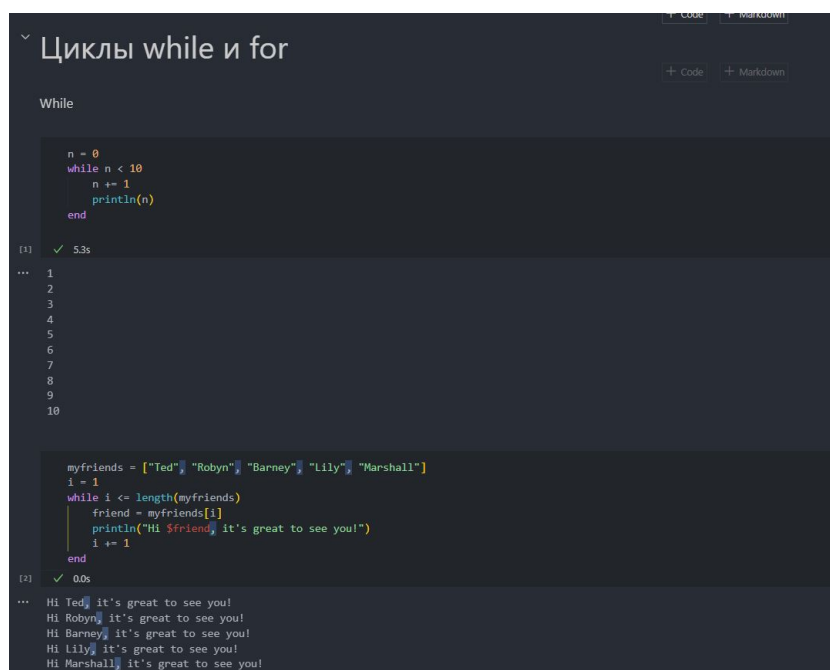
2 Теоретическое введение

Julia — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений.[1]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia[2].

3 Выполнение лабораторной работы

Выполним примеры из лабораторной работы для изучения циклов и функций(рис. fig. 3.1 - fig. 3.5)



```
Циклы while и for

While

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

(1) ✓ 53s
...
1
2
3
4
5
6
7
8
9
10

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

(2) ✓ 00s
...
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. 3.1: Примеры. Циклы

```
for
  for n in 1:2:10
  | println(n)
  end
[3] ✓ 0.0s

... 1
    3
    5
    7
    9

myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
for friend in myfriends
| println("Hi $friend, it's great to see you!")
end
[4] ✓ 0.0s

... 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!

# инициализация массива m x n из нулей:
m, n = 5, 5
A = fill{0}(m, n)
# формирование массива A в котором значение каждой записи
# является суммой индексов строки и столбца:
for i in 1:m
  for j in 1:n
  | A[i, j] = i + j
  end
end
A
[6] ✓ 0.0s

... 5x5 Matrix{Int64}:
```

Рис. 3.2: Примеры. Циклы

```

    | A[i, j] = i + j
    end
end
A

[6] ✓ 0.0s

... 5x5 Matrix{Int64}:
 2 3 4 5 6
 3 4 5 6 7
 4 5 6 7 8
 5 6 7 8 9
 6 7 8 9 10

# инициализация массива m x n из нулей:
B = fill{0, (m, n)}
for i in 1:m, j in 1:n
    | B[i, j] = i + j
end
B

[7] ✓ 0.0s

... 5x5 Matrix{Int64}:
 2 3 4 5 6
 3 4 5 6 7
 4 5 6 7 8
 5 6 7 8 9
 6 7 8 9 10

C = [i + j for i in 1:m, j in 1:n]
C

[8] ✓ 0.0s

... 5x5 Matrix{Int64}:
 2 3 4 5 6
 3 4 5 6 7
 4 5 6 7 8
 5 6 7 8 9
 6 7 8 9 10
```

Рис. 3.3: Примеры. Циклы

Условные выражения

```
N = 10
# 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
```

[10] ✓ 0.0s

... Buzz

```
x = 5
y = 10
(x > y) ? x : y
```

[11] ✓ 0.1s

... 10

Рис. 3.4: Примеры. Условия

Функции

```
function sayhi(name)
|   println("Hi $name, it's great to see you!")
end
[14] ✓ 0.2s
... sayhi (generic function with 1 method)

function f(x)
|   x^2
end
[15] ✓ 0.0s
... f (generic function with 1 method)

sayhi("C-3P0")
f(42)
[18] ✓ 0.0s
... Hi C-3P0, it's great to see you!
... 1764

sayhi2(name) = println("Hi $name, it's great to see you!")
f2(x) = x^2

sayhi("C-3P0")
f(42)
[20] ✓ 0.0s
... Hi C-3P0, it's great to see you!
... 1764
```

Рис. 3.5: Примеры. Функции

```
sayhi3 = name -> println("Hi $name, it's great to see you!")
f3 = x -> x^2

sayhi("C-3P0")
f(42)

[21] ✓ 0.0s
... Hi C-3P0 it's great to see you!
... 1764

v = [3, 5, 2]
sort(v)
v

[22] ✓ 0.3s
... 3-element Vector{Int64}:
 3
 5
 2

sort!(v)
v

[23] ✓ 0.0s
... 3-element Vector{Int64}:
 2
 3
 5

map(f, [1,2,3])

[24] ✓ 0.0s
... 3-element Vector{Int64}:
 1
 4
 9
```

Рис. 3.6: Примеры. Функции

```
> map(x -> x^3, [1, 2, 3])
[25] ✓ 0.0s
... 3-element Vector{Int64}:
      1
      8
     27

broadcast(f, [1, 2, 3])
[26] ✓ 0.0s
... 3-element Vector{Int64}:
      1
      4
      9

f.([1, 2, 3])
[27] ✓ 0.0s
... 3-element Vector{Int64}:
      1
      4
      9

A = [i + 3*j for j in 0:2, i in 1:3]
[28] ✓ 0.0s
... 3x3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

f(A)
[29] ✓ 0.6s
... 3x3 Matrix{Int64}:
30  36  42
55  61  66
80  86  91
```

Рис. 3.7: Примеры. Функции

```
f(A)
[29] ✓ 0.6s

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

f.(A)
[31] ✓ 0.0s

... 3x3 Matrix{Int64}:
      1  4  9
     16 25 36
     49 64 81
```

Рис. 3.8: Примеры. Функции

В первом задании рассмотрим цикл `for` и создадим словарь, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений(рис. fig. 3.9)

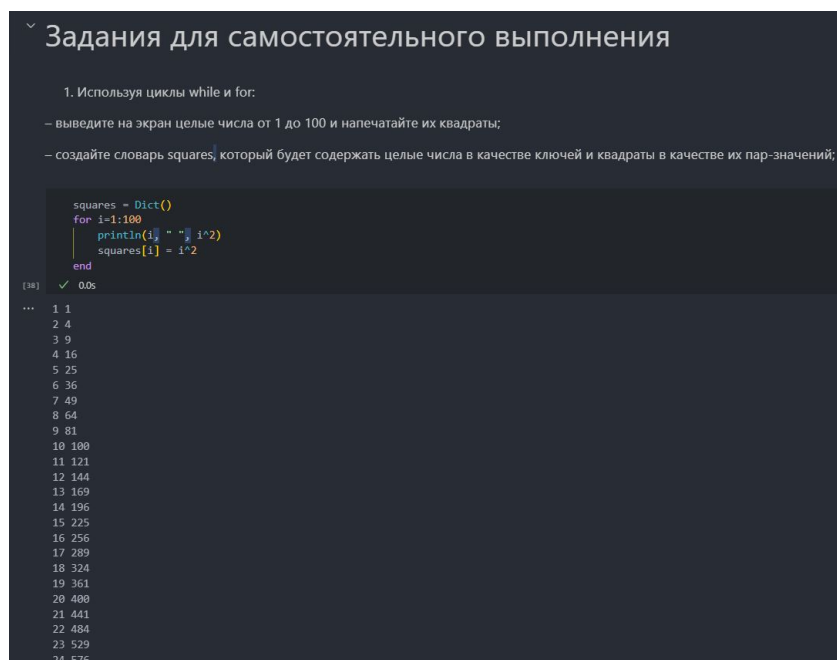


Рис. 3.9: Задание 1

Создадим список с квадратами чисел от 1 до 100:

```
C = [i^2 for i in 1:100]
✓ 0.0s

100-element Vector{Int64}:
 1
 4
 9
16
25
36
49
64
81
100
 ⋮
8464
8649
8836
9025
9216
9409
9604
9801
10000
```

Рис. 3.10: Задание 1

Во втором задании напишем цикл на определение четности числа при помощи условных операторов

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

num = 10
if (num%2==0)
    println(num)
else
    println("Нечётное")
end

(num%2==0) ? num : "Нечётное"
✓ 0.0s
... 10
... 10
```

Рис. 3.11: Задание 2

В третьем напишем простую функцию прибавления единицы,

```

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

function add_one(n)
    n+=1
end
add_one(num)
48] ✓ 0.0s
** 11

```

Рис. 3.12: Задание 3

А в четвертом зададим матрицу A, каждый элемент которой увеличивается на единицу по сравнению с предыдущим(рис. fig. 3.13)

```

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

arr = fill(1, 3, 3)
[map(x->x+1, i+4*j) for i=0:3, j=1:3]
4] ✓ 0.0s
4x3 Matrix{Int64}:
 5  9 13
 6 10 14
 7 11 15
 8 12 16

```

Рис. 3.13: Задание 4

Зададим матрицу A, найдем ее куб и изменим столбец(рис. fig. ??)

```

5. Задайте матрицу A следующего вида:

A = \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix}.

. – Найдите A^3
. – Замените третий столбец матрицы A на сумму второго и третьего столбца

A = [ 1 1 3; 5 2 6; -2 -1 -3]

function cube_f(e1)
    e1^3
end

cube_f(A)
[88] ✓ 0.0s

3x3 Matrix{Int64}:
 1  1 27
125  8 216
-8 -1 -27

cube_f(A)
[79] ✓ 0.0s

3x3 Matrix{Int64}:
 1  0  0
-4 -3 -12
 1  1  4

A
[81] ✓ 0.0s

3x3 Matrix{Int64}:

```

Рис. 3.14: Задание 5


```

A
✓ 0.0s

3x3 Matrix{Int64}:
 1  1  3
 5  2  6
-2 -1 -3

A[:, 3] = A[:, 2] + A[:, 3]
A
✓ 0.0s

3x3 Matrix{Int64}:
 1  1  4
 5  2  8
-2 -1 -4

```

Рис. 3.15: Задание 5

Зададим матрицу и умножим её на обратную себе же(рис. fig. 3.16)

```

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

B = [ 10*(-1)^(((-1)^j)) for i=1:15, j=1:3 ]
✓ 0.0s

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

C = B'*B
✓ 0.0s

3x3 Matrix{Int64}:
1500 -1500 1500
-1500 1500 -1500
1500 -1500 1500

```

Рис. 3.16: Задание 6

При помощи циклов преобразуем матрицы в различные виды(рис. fig. 3.17)

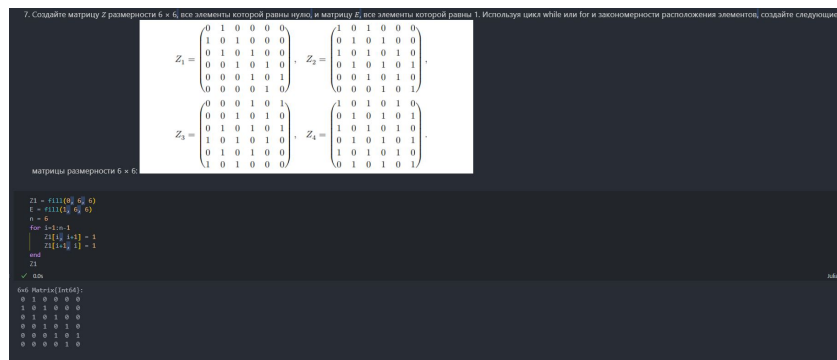


Рис. 3.17: Задание 7

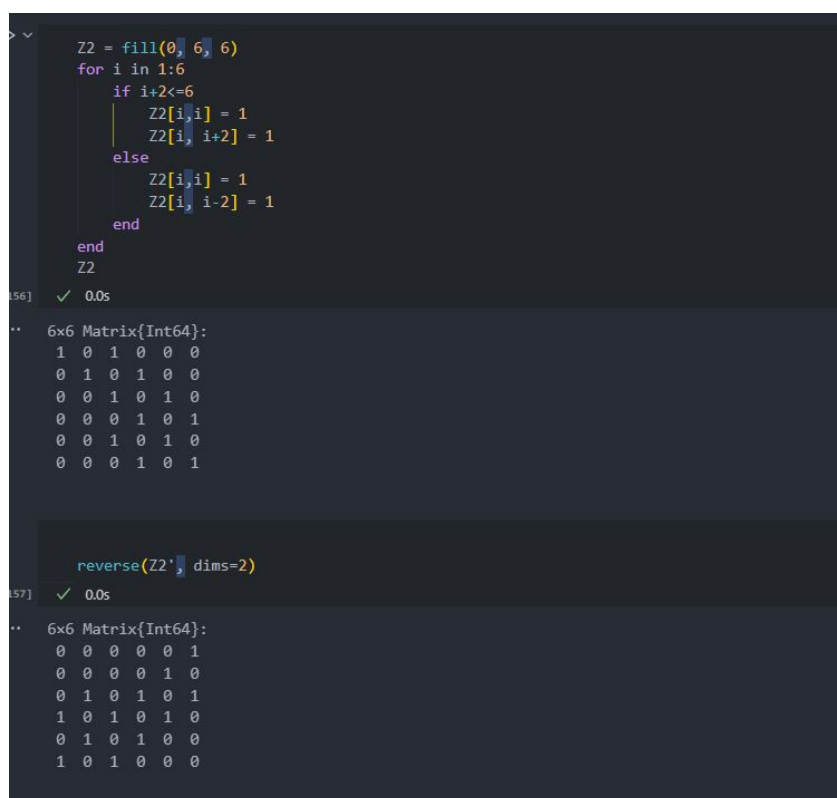


Рис. 3.18: Задание 7

```

Z4 = fill(0, 6, 6)
for i in 1:6
    for j in 1:6
        if (i + j) % 2 != 1
            Z4[i, j] = 1
        end
    end
end
Z4

```

✓ 0.0s

6×6 Matrix{Int64}:

1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1

Рис. 3.19: Задание 7

Создадим функцию эквивалентную одноименной функции из языка R(рис. fig. 3.20)

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` из языка R. Функция должна иметь следующий интерфейс: `outer(x,y,operation)`. Таким образом, функция вида `outer(A,B,*)` должна быть эквивалентна произведению матриц `A` и `B` размерностями `L × M` и `M × N` соответственно

```

function outer(x, y, operation)
    return map([DataArray{Int64}, DataArray{Int64}], operation, y)
end
A1 = outer([0 1 2 3 4] [0 1 2 3 4] [0 1 2 3 4])
A1

```

✓ 0.0s

5x5 Matrix{Int64}:

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

```

A2 = outer([0 1 2 3 4] [1 2 3 4 5] [0 1 2 3 4])
A2

```

✓ 0.0s

5x5 Matrix{Int64}:

0	0	0	0	0
1	1	1	1	1
2	4	9	16	25
3	9	27	81	243
4	16	64	256	1024

Рис. 3.20: Задания 8

```

A3 = outer(outer([0, 1, 2, 3, 4], [0, 1, 2, 3, 4], "+"), 5, "%")
A3
#2] ✓ 0.0s

5x5 Matrix{Int64}:
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

A4 = outer(outer([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], "+"), 10, "%")
A4
#7] ✓ 0.0s

10x10 Matrix{Int64}:
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

A5 = abs.(outer([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], "-"))
A5
#1] ✓ 0.0s

10x10 Matrix{Int64}:
0 1 2 3 4 5 6 7 8 9
1 0 1 2 3 4 5 6 7 8
2 1 0 1 2 3 4 5 6 7
3 2 1 0 1 2 3 4 5 6
4 3 2 1 0 1 2 3 4 5
5 4 3 2 1 0 1 2 3 4

```

Рис. 3.21: Задания 8

Решим линейное уравнение в матричном виде в задании 9(рис. fig. 3.22)

```

coefs = [
    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;
]

right = [7; -1; -3; 5; 17]

res = coefs \ right
println(res)
#99] ✓ 0.3s
... [-2.00000000000000036 3.00000000000000058 4.9999999999999998 1.9999999999999991 -3.9999999999999999]

```

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 = 7, \\ 2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 = -1, \\ 3x_1 + 2x_2 + x_3 + 2x_4 + 3x_5 = -3, \\ 4x_1 + 3x_2 + 2x_3 + x_4 + 2x_5 = 5, \\ 5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 = 17, \end{cases}$$

Решите следующую систему линейных уравнений с 5 неизвестными:

Рис. 3.22: Задания 9

В 10 задании произведем анализ количества элементов матрицы, удовлетворяющих необходимым условиям(рис. fig. 3.12)

```

10. Создайте матрицу M размерности 6 × 10, элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности 1, 2, ..., 10.
- Найдите число элементов в каждой строке матрицы M, которые больше числа N (например, N = 4).
- Определите, в каких строках матрицы M число M (например, M = 7) встречается ровно 2 раза?
- Определите все пары столбцов матрицы M, сумма элементов которых больше K (например, K = 75).

M = rand(1:10, 6, 10)
N = 4
K = 75

count_N = sum(M > N)
println(count_N)
count_2 = [ 1 for i=1:6 if sum([i,j] == 7) == 2]
println(count_2)
count_K = [ [i,j] for i=1:6, j=2:5 if (i1-j) && sum(M[i,j] + M[i,j+1]) > K]
println(count_K)

```

31
Int64[]
[[1, 4], [5, 4], [5, 4], [6, 4], [4, 5]]

Рис. 3.23: Задание 10

В задании 10 найдем значения двух сумм(рис. fig. 3.12)

11. Вычислите выражения

$$\begin{aligned}
 & - \sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+j)}, \\
 & - \sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+ij)}.
 \end{aligned}$$

```

sum1 = sum(i^4/(3+j) for i=1:20, j=1:5)
println(sum1)
sum2 = sum(i^4/(3+i*j) for i=1:20, j=1:5)
println(sum2)

```

31
0.1s
639215.2833333338
89912.02146097131

Рис. 3.24: Задание 11

4 Выводы

В результате выполнения данной лабораторной работы были освоены циклы, функции и сторонние для Julia пакеты для решения задач линейной алгебры и работы с матрицами.

Список литературы

1. JuliaLang [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://julialang.org/> (дата обращения: 11.10.2024).
2. Julia 1.11 Documentation [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://docs.julialang.org/en/v1/> (дата обращения: 11.10.2024).