

Компьютерный практикум по статистическому анализу данных

Лабораторная работа № 3. Управляющие структуры

Сунгурова Мариян Мухсиновна

30 ноября 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Сунгурова Мариян Мухсиновна
- студентка
- НКНбд-01-21
- Российский университет дружбы народов

Введение

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

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

Теоретическое введение

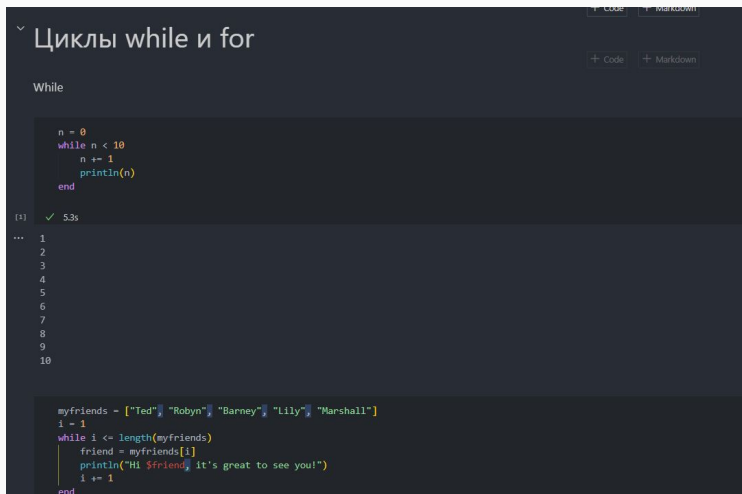
Julia — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений.([julia.lang](#)?). Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные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

Для выполнения заданий была использована официальная документация Julia([julia-doc](#)?).

Выполнение лабораторной работы

Выполнение лабораторной работы

Выполним примеры из лабораторной работы для изучения циклов и функций(рис. (fig:001?) - (fig:005?))



The screenshot shows a Jupyter Notebook interface with a dark theme. The title of the notebook is "Циклы while и for". There are two tabs at the top: "Code" and "Markdown". The first code cell is titled "While" and contains the following R code:

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

Below the code cell, there is a status bar showing a green checkmark, a plus icon, and the text "5.3s". To the left of the code cell, there is a vertical ellipsis and the numbers 1 through 10, indicating the output of the loop.

The second code cell contains the following R code:

```
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
```

Выполнение лабораторной работы

```
for
  for n in 1:2:10
    printing(n)
  end
end
(4) ✓ Obj

1
3
5
7
9

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

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 n x n из нулей:
n0 = 5; 5
A = fill{0} (n0, n0)
# формирование массива A в котором значение каждой клетки
# является суммой индексов строки и столбца:
for i in 1:n
  for j in 1:n
    A[i, j] = i + j
  end
end
A
(4) ✓ Obj

5x5 Matrix{Int64}:
```

Рис. 2: Примеры. Циклы

```
    A[i, j] = i + j
  end
end
A
(4) ✓ Obj

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
```

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. 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-3PO")
#(42)
```

[16] ✓ 0.0s

... Hi C-3PO it's great to see you!

... 1764

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

```
sayhi("C-3PO")
#(42)
```

[18] ✓ 0.0s

... Hi C-3PO it's great to see you!

... 1764

Рис. 5: Примеры. Функции

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

Выполнение лабораторной работы

```
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

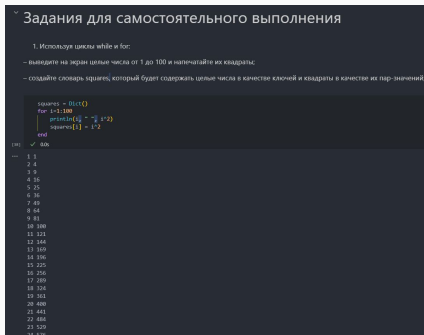
A = [i + 3*j for j in 0:2 i in 1:3]
[28] ✓ 0.0s
... 3x3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

f(A)
[29] ✓ 0.0s
... 3x3 Matrix{Int64}:
30 36 42
 0  0  0
 0  0  0
```

Рис. 7: Примеры. Функции

Выполнение лабораторной работы

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:009?))



The screenshot shows a Jupyter Notebook interface. At the top, there is a section titled "Задания для самостоятельного выполнения" (Tasks for independent execution). Below the title, there are two instructions in Russian: "1. Используя циклы `while` и `for`:" and "– выведите на экран целые числа от 1 до 100 и напечатайте их квадраты:" (print the squares). A second instruction says "– создайте словарь `squares`, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений:" (create a dictionary `squares` with integers as keys and their squares as values). Below the instructions, there is a code cell containing the following Python code:

```
squares = {}
for i in range(1, 101):
    squares[i] = i**2
```

The code is executed, and the output is displayed below it. The output shows the first 24 items of the dictionary, with the key and its square value separated by a colon and a space.

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225, 16: 256, 17: 289, 18: 324, 19: 361, 20: 400, 21: 441, 22: 484, 23: 529, 24: 576}
```

Рис. 9: Задание 1

Выполнение лабораторной работы

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

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

```
2. Напишите условный оператор, который выводит число, если число четное, и строку нечетное, если число нечетное. Перепишите код, используя тернарный оператор.  
  
num = 10  
if (num % 2 == 0)  
    console.log(num)  
else  
    console.log("Нечетное")  
end  
  
console.log(0 > 0 ? num : "Нечетное")  
11 ✓ 10  
12  
13 10
```

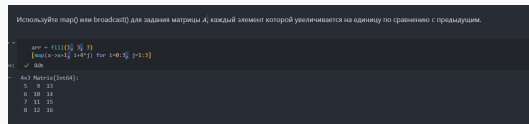
Рис. 11: Задание 2

В третьем напомним 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)  
✓ 0.0s  
11
```

Рис. 12: Задание 3

А в четвертом зададим матрицу A , каждый элемент которой увеличивается на единицу по сравнению с предыдущим(рис. (fig:013?))



```
Используйте map() или broadcast() для задания матрицы A, каждый элемент которой увеличивается на единицу по сравнению с предыдущим.  
  
arr = f111(10, 3, 3)  
(map(x->x+1, 1+4*j) for i=0:3, j=1:3)  
In [ ]: arr  
Out[ ]: array([[5, 6, 7],  
[6, 7, 8],  
[7, 8, 9],  
[8, 9, 10]])
```

Рис. 13: Задание 4

Зададим матрицу A , найдем ее куб и изменим столбец(рис. (fig:0014?))

```
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 столбцов
```

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

```
function cube_f(a)  
    a^3  
end  
cube_f(A)
```

```
(18) ✓ 0.0s  
... 3x3 Matrix{Int64}:  
 1  1  27  
 5  2  216  
 -8 -1  -27
```

```
cube_f(A)
```

```
(19) ✓ 0.0s  
... 3x3 Matrix{Int64}:  
 1  0  0  
 -4 -3 -12  
 1  1  4
```

```
A
```

```
(20) ✓ 0.0s  
... 3x3 Matrix{Int64}:  
 1  1  4  
 5  2  4  
 -2 -1  4
```

Рис. 14: Задание 5

Зададим матрицу и умножим её на обратную себе же(рис. (fig:016?))

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

```
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. 16: Задание 6

Выполнение лабораторной работы

При помощи циклов преобразуем матрицы в различные виды (рис. (fig:017?))

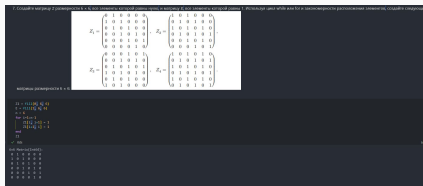
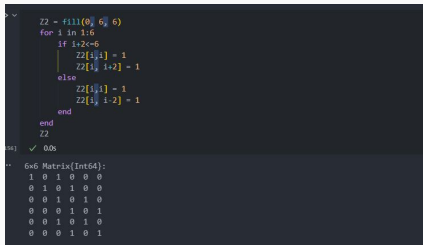


Рис. 17: Задание 7



```
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

6x6 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

Создадим функцию эквивалентную одноименной функции из языка R (рис. (fig:020?))

```

4. В файле N.js функция isPrime() возвращает Boolean, true означает, что переданное значение является простым числом, false означает, что значение не является простым.
Напишите функцию isPrime(), которая принимает на вход число и возвращает Boolean.

Примерный вывод функции isPrime() для функции isPrime() ниже. В функции должна быть следующая логика: isPrime(2) вернет true, isPrime(3) вернет true, isPrime(4) вернет false, isPrime(5) вернет true, isPrime(6) вернет false, isPrime(7) вернет true, isPrime(8) вернет false, isPrime(9) вернет false, isPrime(10) вернет false.

function isPrime(n) {
  // напишите код функции isPrime()
  return true;
}

// isPrime(2) должно вернуть true
// isPrime(3) должно вернуть true
// isPrime(4) должно вернуть false
// isPrime(5) должно вернуть true
// isPrime(6) должно вернуть false
// isPrime(7) должно вернуть true
// isPrime(8) должно вернуть false
// isPrime(9) должно вернуть false
// isPrime(10) должно вернуть false

let isPrime = isPrime(2);
let isPrime = isPrime(3);
let isPrime = isPrime(4);
let isPrime = isPrime(5);
let isPrime = isPrime(6);
let isPrime = isPrime(7);
let isPrime = isPrime(8);
let isPrime = isPrime(9);
let isPrime = isPrime(10);

// isPrime(2) должно вернуть true
// isPrime(3) должно вернуть true
// isPrime(4) должно вернуть false
// isPrime(5) должно вернуть true
// isPrime(6) должно вернуть false
// isPrime(7) должно вернуть true
// isPrime(8) должно вернуть false
// isPrime(9) должно вернуть false
// isPrime(10) должно вернуть false

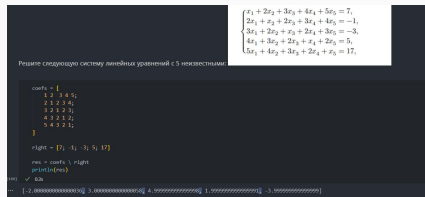
```

Рис. 20: Задания 8

```
A3 = outer(outer([0 1 2 3 4 5 6 7 8 9],[0 1 2 3 4 5 6 7 8 9],"*"),[0 1 2 3 4 5 6 7 8 9])
A3
ans =
0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9
4 5 6 7 8 9
5 6 7 8 9
6 7 8 9
7 8 9
8 9
9

A6 = outer(outer([0 1 2 3 4 5 6 7 8 9],[0 1 2 3 4 5 6 7 8 9],"*"),[0 1 2 3 4 5 6 7 8 9])
A6
ans =
0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9
4 5 6 7 8 9
5 6 7 8 9
6 7 8 9
7 8 9
8 9
9
```


Решим линейное уравнение в матричном виде в задании 9(рис. (fig:022?))



Решите следующую систему линейных уравнений с 5 неизвестными:

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

```
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)
```

```
[ -2.00000000000000e+00  3.00000000000000e+00  4.00000000000000e+00  1.00000000000000e+00  3.00000000000000e+00]
```

Рис. 22: Задания 9

В 10 задании произведем анализ количества элементов матрицы, удовлетворяющих необходимым условиям(рис. (fig:012?))

```
10. Создайте матрицу  $M$  размерности  $6 \times 10$ , 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,  $N = 4$ ).  
– 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_n = [ 1 for i=1:6 if sum([i,j].==7)==2]  
println(count_n)  
count_k = [(i,j) for i=1:6, j=2:5 if (11-i) && sum(M[:,i]+M[:,j])>K]  
println(count_k)
```

```
241] ✓ 0.5s  
21  
Int64[]  
[(1, 4), (5, 4), (6, 4), (4, 5)]
```

Рис. 23: Задание 10

Выполнение лабораторной работы

В задании 10 найдем значения двух сумм(рис. (fig:012?))

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

11. Вычислите выражения

```
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)
```

✓ 0.1s

```
639215.2833333338
89912.02146097131
```

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.