

Компьютерный практикум по статистическому анализу данных

Лабораторная работа № 3. Управляющие структуры

Демидова Екатерина Алексеевна

Содержание

1	Введение	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Выводы	16
	Список литературы	17

Список иллюстраций

3.1	Примеры. Циклы	6
3.2	Примеры. Условия	6
3.3	Примеры. Циклы	7
3.4	Примеры. Условия	8
3.5	Примеры. Функции	9
3.6	Задание 1	9
3.7	Задание 2,3,4	10
3.8	Задание 5	11
3.9	Задание 6	12
3.10	Задание 7	13
3.11	Задания 8, 9	14
3.12	Задание 10,11	15

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 2.2.
2. Выполните задания для самостоятельной работы (раздел 2.4).

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. 3.1 - 3.5)

```
# цикл while повторять, если условие и тело цикла истинны
i = 0
while i < 10:
    print(i)
    i += 1

myFriends = ["Ted", "Moby", "Barney", "Lily", "Marshall"]
l = 1
while l <= len(myFriends):
    friend = myFriends[l]
    print(f"Hi {friend}, it's great to see you!")
    l += 1
end

1) За
1
2
3
4
5
6
7
8
9
10
Hi Ted, it's great to see you!
Hi Moby, 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!

D)
for n in 1..10:
    print(n)
    myFriends = ["Ted", "Moby", "Barney", "Lily", "Marshall"]
    for friend in myFriends:
        print(f"Hi {friend}, it's great to see you!")
    end
end

1) За
1
2
3
4
5
6
7
8
9
10
Hi Ted, it's great to see you!
Hi Moby, 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

```
# n = 5, 5
A = fill{0, 10, 10}
# заполнение массива A значением каждый элемент
for i in 1:n
    for j in 1:n
        A[i, j] = i + j
    end
end
A

1) За
5x5 Matrix{Int64}:
 5  6  7  8  9
 6  7  8  9 10
 7  8  9 10 11
 8  9 10 11 12
 9 10 11 12 13

D)
# заполнение 10x10 матрицы значениями "Fizz"
# заполнение 5 элементов матрицы 0, остальные
n = 10
if (i % 3 == 0) && (i % 5 == 0)
    print(f"{i} = 0")
elseif i % 3 == 0
    print(f"{i} = Fizz")
elseif i % 5 == 0
    print(f"{i} = Buzz")
else
    print(i)
end

1) За
0
1
2
3
4
5
6
7
8
9
10

D)
0 1 2 3 4 5 6 7 8 9 10
```

Рис. 3.2: Примеры. Условия

```
# пока n<10 прибавить к n единицу и распечатать значение:
n = 0
while n < 10
  n += 1
  println(n)
end

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

[1] ✓ 2.6s

...
1
2
3
4
5
6
7
8
9
10
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!

▷
for n in 1:2:10
  println(n)
end
myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]
for friend in myfriends
  println("Hi $friend, it's great to see you!")
end

[3] ✓ 0.0s

...
1
3
5
7
9
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. 3.3: Примеры. Циклы

```

A = zeros(0, (m, n))
# формирование массива, 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

[5] ✓ 2.3s

... 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

# используем `&&` для реализации операции "AND"
# операция % вычисляет остаток от деления
N = 10
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

[7] ✓ 0.0s

... Buzz

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

[8] ✓ 0.1s

... 10

▷ function sayhi(name)
  println("Hi $name, it's great to see you!")
end
# функция возведения в квадрат:
function f(x)
  x^2
end

[9] ✓ 0.2s

... f (generic function with 1 method)
```

Рис. 3.4: Примеры. Условия


```

function sayhi(name)
    println("Hi $name, it's great to see you!")
end
# функция возведения в квадрат:
function f(x)
    x^2
end

[9] ✓ 0.2s

... f (generic function with 1 method)

f(x) = x^2
broadcast(f, [1, 2, 3])
f.([1, 2, 3])

# Задаём матрицу A:
A = [i + 3*j for j in 0:2, i in 1:3]
f(A)
@. A + 2 * f(A) / A

[11] ✓ 1.0s

... 3x3 Matrix{Float64}:
 3.0  6.0  9.0
12.0 15.0 18.0
21.0 24.0 27.0

```

Рис. 3.5: Примеры. Функции

В первом задании рассмотрим цикл while и создадим словарь, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений(рис. 3.6)

```

1. Используя циклы while и for:
• выведите на экран целые числа от 1 до 100 и напечатайте их квадраты;
• создайте словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;

1 = 1
squares = Dict{<
while i<=100
    println(i, " ", i^2)
    squares[i] = i^2
    i+=1
end

[17]

... 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
    25 625
    ...
    97 9409
    98 9604
    99 9801
    100 10000

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings.

```

Рис. 3.6: Задание 1

Во втором задании напишем цикл на определение четности числа, в третьем

напишем простую функцию прибавления единицы, а в четвертом зададим матрицу A, каждый элемент которой увеличивается на единицу по сравнению с предыдущим(рис. 3.7)

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

num = 4
if iseven(num)
| println(num)
else
| println("нечётное")
end
iseven(num) ? println(num) : println("нечётное")

[27]
... 4
4

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

function add_one(x)
| x+=1
end
add_one(4)

[28]
... 5

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

A = fill{1,4,4}
map(e->e+1, 1+4*j) for i=0:3, j=1:4]

[32]
✓ 0.0s
... 4x4 Matrix{Int64}:
5  9 13 17
6 10 14 18
7 11 15 19
8 12 16 20
```

Рис. 3.7: Задание 2,3,4

Зададим матрицу A, найдем ее куб и изменим столбец(рис. 3.8)

```
3.
• Найдите  $A^3$ .
• Замените третий столбец матрицы  $A$  на сумму второго и третьего столбцов.

13] ✓ 0.0s
A = [1 1 3; 5 2 6; -2 -1 -3]
3x3 Matrix{Int64}:
 1  1  3
 5  2  6
-2 -1 -3

14] ✓ 0.0s
A^3
3x3 Matrix{Int64}:
 0  0  0
 0  0  0
 0  0  0

15] ✓ 0.5s
A[:,3] = A[:,2]+A[:,1]
3-element Vector{Int64}:
 2
 7
-3

16] ✓ 0.0s
A
3x3 Matrix{Int64}:
 1  1  2
 5  2  7
-2 -1 -3
```

Рис. 3.8: Задание 5

Зададим матрицу и умножим её на обратную себе же(рис. 3.9)

```
№ 6

B = [10*((-1)^(j-1)) for i=1:15, j=1:3]
[17] ✓ 0.0s

... 15×3 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

B'*B
[18] ✓ 0.7s

... 3×3 Matrix{Int64}:
 1500 -1500  1500
-1500  1500 -1500
 1500 -1500  1500
```

Рис. 3.9: Задание 6

При помощи циклов преобразуем матрицы в различные виды(рис. 3.10)

```

Nº 7

Z = fill(0,6,6)
E = fill(1, 6,6)
[21] ✓ 0.0s

... 6x6 Matrix{Int64}:
 1 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 1

▷
n = 6
Z1 = fill(0,n,n)
for i in 1:n
    if i==1
        Z1[i,i+1] = 1
    elseif i==n
        Z1[i,i-1] = 1
    else
        Z1[i,i+1] = 1
        Z1[i,i-1] = 1
    end
end
print(Z1)
[22] ✓ 1.1s

... [0 1 0 0 0 0; 1 0 1 0 0 0; 0 1 0 1 0 0; 0 0 1 0 1 0; 0 0 0 1 0 1; 0 0 0 0 1 0]

Z2 = fill(0,n,n)
for i in 1:6
    if i+2<6
        Z2[i,i] = 1
        Z2[i,i+2] = 1
    else
        Z2[i,i] = 1
        Z2[i,i-2] = 1
    end
end
print(Z2)
[23] ✓ 0.0s

... [1 0 1 0 0 0; 0 1 0 1 0 0; 0 0 1 0 1 0; 0 1 0 1 0 0; 0 0 1 0 1 0; 0 0 0 1 0 1]

```

Рис. 3.10: Задание 7

Создадим функцию эквивалентную одноименной функции из языка R, а в 9 задании решим линейное уравнение в матричном виде(рис. 3.11)

```
№ 8

function outer(x,y,operation)
  return eval(Meta.parse("$x $operation $y"))
end
outer([1 2],[1;2],*)

[24] ✓ 0.2s

... 1-element Vector{Int64}:
     5

№ 9

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

[7]

... 5-element Vector{Float64}:
 -1.9999999999999997
  2.9999999999999996
  4.9999999999999998
  2.0000000000000001
 -4.0
```

Рис. 3.11: Задания 8, 9

В 10 задании произведем анализ количества элементов матрицы, удовлетворяющих необходимым условиям, а в 10 найдем значения математических выражений(рис. 3.12)

```

№ 10

M = rand(1:10,6,10)
N = 4
K = 75
count1 = sum(M.>N)
count2 = [1 for i=1:6 if sum(M[i,:].==N) == 2]
col = [(i,j) for i=1:6, j=2:5 if (i!=j && sum(M[:,i] + M[:,j])>K)]

[78] ... 8-element Vector{Tuple{Int64, Int64}}:
      (4, 2)
      (4, 3)
      (1, 4)
      (2, 4)
      (3, 4)
      (5, 4)
      (6, 4)
      (4, 5)

№ 11

sum([i^4/(3+j) for i=1:20, j=1:5])
sum([i^4/(3+j*j) for j=1:5, i=1:20])

[25] ✓ 0.3s

... 407971.2192982457

A = [1 2]
size(A)[1]

[26] ✓ 0.0s

... 1

```

Рис. 3.12: Задание 10,11

4 Выводы

В результате выполнения работы освоили применение циклов функций и сторонних для Julia пакетов 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).