

Лабораторная работа №3

Компьютерный практикум по статистическому анализу данных

Николаев Д. И.

18 ноября 2023

Российский университет дружбы народов, Москва, Россия

Прагматика выполнения

- Получение навыков работы в Jupyter Notebook;
- Освоение особенностей языка Julia;
- Применение полученных знаний на практике в дальнейшем.

Цели

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).

Выполнение работы. Повторение
примеров

▼ Лабораторная работа № 3. Управляющие структуры

Повторение примеров

Циклы while и for

[4]: *# 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
```

Рис. 1: Циклы while и for (1)

Повторение примеров (2)

```
[6]: 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!
```

```
[7]: for n in 1:2:10  
      println(n)  
end
```

```
1  
3  
5  
7  
9
```

```
[8]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]  
for friend in myfriends  
  println("Hi $friend, it's great to see you!")  
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!
```

Повторение примеров (3)

```
[9]: # инициализация массива m x n из нулей:
m, n = 5, 5
A = fill(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
```

```
[9]: 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
```

Другая реализация этого же примера:

```
[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
```

```
[10]: 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
```

Ещё одна реализация этого же примера:

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

```
[11]: 5×5 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. 4: Циклы while и for (4)

Условные выражения

```
[13]: # используем `&&` для реализации операции "AND"  
# операция % вычисляет остаток от деления  
N = 5  
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
```

Buzz

```
[14]: x = 5  
      y = 10  
      (x > y) ? x : y
```

[14]: 10

Повторение примеров (6)

Функции

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

[15]: f (generic function with 1 method)

Вызов функции осуществляется по её имени с указанием аргументов, например:

```
[16]: sayhi("C-3PO")
      f(42)
```

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

[16]: 1764

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

[17]: f2 (generic function with 1 method)

Наконец, можно объявить выше определённые функции как «анонимные»:

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

[18]: #5 (generic function with 1 method)

Повторение примеров (7)

```
[20]: # задаём массив v:  
v = [3, 5, 2]  
sort(v)
```

```
[20]: 3-element Vector{Int64}:  
      2  
      3  
      5
```

```
[21]: v
```

```
[21]: 3-element Vector{Int64}:  
      3  
      5  
      2
```

```
[22]: sort!(v)
```

```
[22]: 3-element Vector{Int64}:  
      2  
      3  
      5
```

```
[23]: v
```

```
[23]: 3-element Vector{Int64}:  
      2  
      3  
      5
```

Повторение примеров (8)

```
[24]: f(x) = x^2  
      map(f, [1, 2, 3])
```

```
[24]: 3-element Vector{Int64}:  
      1  
      4  
      9
```

```
[25]: x -> x^3  
      map(x -> x^3, [1, 2, 3])
```

```
[25]: 3-element Vector{Int64}:  
      1  
      8  
     27
```

```
[26]: f(x) = x^2  
      broadcast(f, [1, 2, 3])
```

```
[26]: 3-element Vector{Int64}:  
      1  
      4  
      9
```

```
[27]: f.([1, 2, 3])
```

```
[27]: 3-element Vector{Int64}:  
      1  
      4  
      9
```


Повторение примеров (9)

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

```
[28]: 3x3 Matrix{Int64}:  
  1  2  3  
  4  5  6  
  7  8  9
```

```
[29]: # Вызываем функцию f возведения в квадрат  
f(A)
```

```
[29]: 3x3 Matrix{Int64}:  
 30  36  42  
 66  81  96  
102 126 150
```

```
[30]: B = f.(A)
```

```
[30]: 3x3 Matrix{Int64}:  
  1  4  9  
 16 25 36  
 49 64 81
```

Повторение примеров (10)

```
[31]: A .+ 2 .* f.(A) ./ A
```

```
[31]: 3x3 Matrix{Float64}:  
      3.0  6.0  9.0  
      12.0 15.0 18.0  
      21.0 24.0 27.0
```

```
[32]: @. A + 2 * f(A) / A
```

```
[32]: 3x3 Matrix{Float64}:  
      3.0  6.0  9.0  
      12.0 15.0 18.0  
      21.0 24.0 27.0
```

```
[33]: broadcast(x -> x + 2 * f(x) / x, A)
```

```
[33]: 3x3 Matrix{Float64}:  
      3.0  6.0  9.0  
      12.0 15.0 18.0  
      21.0 24.0 27.0
```

Сторонние библиотеки (пакеты) в Julia

```
[34]: import Pkg  
      Pkg.add("Example")
```

```
Updating registry at `C:\Users\User\.julia\registries\General.toml`  
Resolving package versions...  
Installed Example - v0.5.3  
Updating `C:\Users\User\.julia\environments\v1.8\Project.toml`  
[7876af07] + Example v0.5.3  
Updating `C:\Users\User\.julia\environments\v1.8\Manifest.toml`  
[7876af07] + Example v0.5.3  
Precompiling project...  
✓ Example  
1 dependency successfully precompiled in 7 seconds. 287 already precompiled.
```

```
[35]: Pkg.add("Colors")  
      using Colors
```

```
Resolving package versions...  
Updating `C:\Users\User\.julia\environments\v1.8\Project.toml`  
[5ae59095] + Colors v0.12.10  
No Changes to `C:\Users\User\.julia\environments\v1.8\Manifest.toml`
```

Рис. 11: Сторонние библиотеки/пакеты (1)

Повторение примеров (12)

```
[36]: palette = distinguishable_colors(100)
```

```
[36]:
```



```
[54]: rand(palette, 3, 3)
```

```
[54]:
```



Рис. 12: Сторонние библиотеки/пакеты (2)

Выполнение работы.

Самостоятельная работа

Задание 1. Пункт 1

Самостоятельное задание

1. Используя циклы while и for:

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

```
[59]: n = 0
while n < 100
  n += 1
  print("$n --> $(n^2); ")
end
n = 0;

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; 26 --> 676; 27 --> 729; 28 --> 784; 29 --> 841; 30 --> 900; 31 --> 96
1; 32 --> 1024; 33 --> 1089; 34 --> 1156; 35 --> 1225; 36 --> 1296; 37 --> 1369; 38 --> 1444; 39 --> 1521; 40 --> 1600; 41 --> 1681; 42 --> 1764; 43 --> 1849; 44 --> 1936; 45
--> 2025; 46 --> 2116; 47 --> 2209; 48 --> 2304; 49 --> 2401; 50 --> 2500; 51 --> 2601; 52 --> 2704; 53 --> 2809; 54 --> 2916; 55 --> 3025; 56 --> 3136; 57 --> 3249; 58 --> 33
64; 59 --> 3481; 60 --> 3600; 61 --> 3721; 62 --> 3844; 63 --> 3969; 64 --> 4096; 65 --> 4225; 66 --> 4356; 67 --> 4489; 68 --> 4624; 69 --> 4761; 70 --> 4900; 71 --> 5041; 72
--> 5184; 73 --> 5329; 74 --> 5476; 75 --> 5625; 76 --> 5776; 77 --> 5929; 78 --> 6084; 79 --> 6241; 80 --> 6400; 81 --> 6561; 82 --> 6724; 83 --> 6889; 84 --> 7056; 85 --> 72
25; 86 --> 7396; 87 --> 7569; 88 --> 7744; 89 --> 7921; 90 --> 8100; 91 --> 8281; 92 --> 8464; 93 --> 8649; 94 --> 8836; 95 --> 9025; 96 --> 9216; 97 --> 9409; 98 --> 9604; 99
--> 9801; 100 --> 10000;

[60]: for i ∈ 1:100
  print("$i --> $(i^2); ")
end

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; 26 --> 676; 27 --> 729; 28 --> 784; 29 --> 841; 30 --> 900; 31 --> 96
1; 32 --> 1024; 33 --> 1089; 34 --> 1156; 35 --> 1225; 36 --> 1296; 37 --> 1369; 38 --> 1444; 39 --> 1521; 40 --> 1600; 41 --> 1681; 42 --> 1764; 43 --> 1849; 44 --> 1936; 45
--> 2025; 46 --> 2116; 47 --> 2209; 48 --> 2304; 49 --> 2401; 50 --> 2500; 51 --> 2601; 52 --> 2704; 53 --> 2809; 54 --> 2916; 55 --> 3025; 56 --> 3136; 57 --> 3249; 58 --> 33
64; 59 --> 3481; 60 --> 3600; 61 --> 3721; 62 --> 3844; 63 --> 3969; 64 --> 4096; 65 --> 4225; 66 --> 4356; 67 --> 4489; 68 --> 4624; 69 --> 4761; 70 --> 4900; 71 --> 5041; 72
--> 5184; 73 --> 5329; 74 --> 5476; 75 --> 5625; 76 --> 5776; 77 --> 5929; 78 --> 6084; 79 --> 6241; 80 --> 6400; 81 --> 6561; 82 --> 6724; 83 --> 6889; 84 --> 7056; 85 --> 72
25; 86 --> 7396; 87 --> 7569; 88 --> 7744; 89 --> 7921; 90 --> 8100; 91 --> 8281; 92 --> 8464; 93 --> 8649; 94 --> 8836; 95 --> 9025; 96 --> 9216; 97 --> 9409; 98 --> 9604; 99
--> 9801; 100 --> 10000;
```

Рис. 13: Задание 1. Пункт 1

Задание 1. Пункт 2 (1)

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

```
[71]: squares_1 = Dict()
      while n < 100
        n += 1
        squares_1[n] = n^2
      end
      n = 0;
      pairs(squares_1)
```

[71]: Dict{Any, Any} with 100 entries:

```
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
: => :
```

Задание 1. Пункт 2 (2)

```
[72]: squares_2 = Dict()
      for i in 1:100
          squares_2[i] = i^2
      end
      squares_2
```

```
[72]: Dict{Any, Any} with 100 entries:
```

$$\begin{array}{l} 5 \Rightarrow 25 \\ 56 \Rightarrow 3136 \\ 35 \Rightarrow 1225 \\ 55 \Rightarrow 3025 \\ 60 \Rightarrow 3600 \\ 30 \Rightarrow 900 \\ 32 \Rightarrow 1024 \\ 6 \Rightarrow 36 \\ 67 \Rightarrow 4489 \\ 45 \Rightarrow 2025 \\ 73 \Rightarrow 5329 \\ 64 \Rightarrow 4096 \\ 90 \Rightarrow 8100 \\ 4 \Rightarrow 16 \\ 13 \Rightarrow 169 \\ 54 \Rightarrow 2916 \\ 63 \Rightarrow 3969 \\ 86 \Rightarrow 7396 \\ 91 \Rightarrow 8281 \\ 62 \Rightarrow 3844 \\ 58 \Rightarrow 3364 \\ 52 \Rightarrow 2704 \\ 12 \Rightarrow 144 \\ 28 \Rightarrow 784 \\ 75 \Rightarrow 5625 \\ \vdots \Rightarrow \vdots \end{array}$$

Задание 1. Пункт 3 (1)

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

```
[73]: squares_arr_1 = (Int64[])  
      while n < 100  
        n += 1  
        push!(squares_arr_1, n^2)  
      end  
      n = 0;  
      squares_arr_1
```

```
[73]: 100-element Vector{Int64}:
```

```
 1  
 4  
 9  
16  
25  
36  
49  
64  
81  
100  
121  
144  
169  
⋮  
7921  
8100  
8281  
8464  
8649  
8836  
9025  
9216  
9409  
9604  
9801  
10000
```

Задание 1. Пункт 3 (2)

```
[75]: squares_arr_2 = (Int64[])  
      for i ∈ 1:100  
          push!(squares_arr_2, i^2)  
      end  
      squares_arr_2
```

```
[75]: 100-element Vector{Int64}:
```

```
 1  
 4  
 9  
16  
25  
36  
49  
64  
81  
100  
121  
144  
169  
⋮  
7921  
8100  
8281  
8464  
8649  
8836  
9025  
9216  
9409  
9604  
9801  
10000
```

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

```
[00]: x = 5; y = 6;

[01]: iseven(x) ? println(x) : println("Нечётное")
      iseven(y) ? println(y) : println("Нечётное")

      Нечётное
      6
```

Рис. 18: Задание 2. Тернарный условный оператор

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

```
[85]: add_one(x) = x + 1
```

```
[85]: add_one (generic function with 1 method)
```

```
[86]: add_one(1)
```

```
[86]: 2
```

Рис. 19: Задание 3.

Задание 4

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

```
[87]: A = zeros(3, 3)

[87]: 3x3 Matrix{Float64}:
 0.0  0.0  0.0
 0.0  0.0  0.0
 0.0  0.0  0.0

[92]: add_one.(A)

[92]: 3x3 Matrix{Float64}:
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0

[93]: map(add_one, A)

[93]: 3x3 Matrix{Float64}:
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0

[98]: x = 0
      for i ∈ 1:3, j ∈ 1:3
          A[i, j] = add_one(x)
          x += 1
      end
      A

[98]: 3x3 Matrix{Float64}:
 1.0  2.0  3.0
 4.0  5.0  6.0
 7.0  8.0  9.0
```

Рис. 20: Задание 4.

5. Задайте матрицу A следующего вида:

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

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

```
[99]: 3x3 Matrix{Int64}:
```

```
 1  1  3
 5  2  6
-2 -1 -3
```

Рис. 21: Задание 5. Пункт 1

– Найдите A^3 .

```
[100]: g(x) = x^3
```

```
[100]: g (generic function with 1 method)
```

```
[105]: g(A)
```

```
[105]: 3x3 Matrix{Int64}:  
      0  0  0  
      0  0  0  
      0  0  0
```

– Замените третий столбец матрицы A на сумму второго и третьего столбцов.

```
[106]: for i ∈ 1:3  
        A[i, 3] = A[i, 1] + A[i, 2]  
    end  
A
```

```
[106]: 3×3 Matrix{Int64}:  
  1  1  2  
  5  2  7  
 -2 -1 -3
```

Рис. 23: Задание 5. Пункт 3

Задание 6 (1)

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

```
[118]: N = 15
B = zeros(15, 15)
#B = [(-1)^n * 10 * ones(N) for n in 1:N]
for i in 1:N, j in 1:N
    if isodd(j)
        B[i, j] = 10
    else
        B[i, j] = -10
    end
end
B
```

[118]: 15x15 Matrix{Float64}:

10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0
10.0	-10.0	10.0	-10.0	10.0	-10.0	...	10.0	-10.0	10.0	-10.0	10.0

Рис. 24: Задание 6 (1).

Задание 6 (2)

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

```
[120]: 15x15 Matrix{Float64}:
```

```
 1500.0 -1500.0  1500.0 -1500.0 ... -1500.0  1500.0 -1500.0  1500.0
-1500.0  1500.0 -1500.0  1500.0    1500.0 -1500.0  1500.0 -1500.0
 1500.0 -1500.0  1500.0 -1500.0   -1500.0  1500.0 -1500.0  1500.0
-1500.0  1500.0 -1500.0  1500.0    1500.0 -1500.0  1500.0 -1500.0
 1500.0 -1500.0  1500.0 -1500.0   -1500.0  1500.0 -1500.0  1500.0
-1500.0  1500.0 -1500.0  1500.0 ...  1500.0 -1500.0  1500.0 -1500.0
 1500.0 -1500.0  1500.0 -1500.0   -1500.0  1500.0 -1500.0  1500.0
-1500.0  1500.0 -1500.0  1500.0    1500.0 -1500.0  1500.0 -1500.0
 1500.0 -1500.0  1500.0 -1500.0   -1500.0  1500.0 -1500.0  1500.0
-1500.0  1500.0 -1500.0  1500.0    1500.0 -1500.0  1500.0 -1500.0
 1500.0 -1500.0  1500.0 -1500.0   -1500.0  1500.0 -1500.0  1500.0
-1500.0  1500.0 -1500.0  1500.0    1500.0 -1500.0  1500.0 -1500.0
 1500.0 -1500.0  1500.0 -1500.0   -1500.0  1500.0 -1500.0  1500.0
```

Рис. 25: Задание 6 (2).

Задание 7 (1)

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 :

$$Z_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, Z_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}, Z_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, Z_4 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

```
[129]: E = [1 for i in 1:6, j in 1:6]
```

```
[129]: 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
```

```
[133]: Z = [0 for i in 1:6, j in 1:6]
```

```
[133]: 6x6 Matrix{Int64}:
```

```
 0 0 0 0 0 0
 0 0 0 0 0 0
 0 0 0 0 0 0
 0 0 0 0 0 0
 0 0 0 0 0 0
 0 0 0 0 0 0
```

Рис. 26: Задание 7.

Задание 7 (2)

```
[135]: Z_1 = Int.(zeros(6, 6))  
      Z_2 = Int.(zeros(6, 6))  
      Z_3 = Int.(zeros(6, 6))  
      Z_4 = Int.(zeros(6, 6))
```

```
[135]: 6×6 Matrix{Int64}:  
 0 0 0 0 0 0  
 0 0 0 0 0 0  
 0 0 0 0 0 0  
 0 0 0 0 0 0  
 0 0 0 0 0 0  
 0 0 0 0 0 0
```

```
[141]: for i ∈ 1:6, j ∈ 1:6  
      if (j == i + 1) || (j == i - 1)  
          Z_1[i, j] = E[i, j]  
      else  
          Z_1[i, j] = Z[i, j]  
      end  
end  
Z_1
```

```
[141]: 6×6 Matrix{Int64}:  
 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
```

```
[142]: for i ∈ 1:6, j ∈ 1:6
        if (j == i + 2) || (j == i - 2) || (j == i)
            Z_2[i, j] = E[i, j]
        else
            Z_2[i, j] = Z[i, j]
        end
    end
    Z_2
```

```
[142]: 6×6 Matrix{Int64}:
 1  0  1  0  0  0
 0  1  0  1  0  0
 1  0  1  0  1  0
 0  1  0  1  0  1
 0  0  1  0  1  0
 0  0  0  1  0  1
```

Задание 7 (4)

```
[144]: for i ∈ 1:6, j ∈ 1:6
        if (j == 7 - i) || (j == 5 - i) || (j == 9 - i)
            Z_3[i, j] = E[i, j]
        else
            Z_3[i, j] = Z[i, j]
        end
    end
Z_3
```

```
[144]: 6×6 Matrix{Int64}:
 0  0  0  1  0  1
 0  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  0
 1  0  1  0  0  0
```

Рис. 29: Задание 7. Матрица Z_3

Задание 7 (5)

```
[146]: for i ∈ 1:6, j ∈ 1:6
        if abs(i - j) == 0 || abs(i - j) == 2 || abs(i - j) == 4  # (j == i + 2) || (j == i - 2) || (j == i) || (j == i + 4) || (j == i - 4)
            Z_4[i, j] = E[i, j]
        else
            Z_4[i, j] = Z[i, j]
        end
    end
    Z_4

[146]: 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. 30: Задание 7. Матрица Z_4

Задание 8. Пункт 1

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)`. Таким образом, функция вида `outer(A,B,"")` должна быть эквивалентна произведению матриц A и B размерностями $L \times M$ и $M \times N$ соответственно, где элементы результирующей матрицы C имеют вид $C_{ij} = \sum_{k=1}^M A_{ik} B_{kj}$ (или в тензорном

виде $C_i^j = \sum_{k=1}^M A_k^i B_j^k$).

```
[168]: function outer(x, y, operation)
      Res = zeros(length(x), length(y))
      for i in 1:length(x), j in 1:length(y)
        Res[i, j] = operation(x[i], y[j])
      end
      return Res
    end
```

```
[168]: outer (generic function with 1 method)
```

```
[170]: x = [1, 2, 3];
      y = [4, 5, 6];
      outer(x, y, +)
```

```
[170]: 3x3 Matrix{Float64}:
      5.0  6.0  7.0
      6.0  7.0  8.0
      7.0  8.0  9.0
```

Рис. 31: Задание 8. Пункт 1. Написание функции `outer`

Задание 8. Пункт 2 (1)

– Используя написанную вами функцию `outer()`, создайте матрицы следующей структуры: $A_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 8 & 16 & 32 \\ 3 & 9 & 27 & 81 & 243 \\ 4 & 16 & 64 & 256 & 1024 \end{pmatrix}$, $A_3 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{pmatrix}$,

$$A_4 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \\ : & : & : & : & : & : & : & : & : & : \\ 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}, A_5 = \begin{pmatrix} 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 \\ 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 \\ 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{pmatrix}$$

```
[176]: a_1 = [0, 1, 2, 3, 4]
      b_1 = [0, 1, 2, 3, 4]
      A_1 = outer(a_1, b_1, +)
```

```
[176]: 5x5 Matrix(Float64):
 0.0 1.0 2.0 3.0 4.0
 1.0 2.0 3.0 4.0 5.0
 2.0 3.0 4.0 5.0 6.0
 3.0 4.0 5.0 6.0 7.0
 4.0 5.0 6.0 7.0 8.0
```

Рис. 32: Задание 8. Пункт 2. Матрица A_1

Задание 8. Пункт 2 (2)

```
[179]: a_2 = a_1  
b_2 = [1, 2, 3, 4, 5]  
A_2 = outer(a_2, b_2, ^)
```

```
[179]: 5x5 Matrix{Float64}:  
 0.0  0.0  0.0  0.0  0.0  
 1.0  1.0  1.0  1.0  1.0  
 2.0  4.0  8.0 16.0 32.0  
 3.0  9.0 27.0 81.0 243.0  
 4.0 16.0 64.0 256.0 1024.0
```

```
[190]: a_3 = [0, 1, 2, 3, 4]  
b_3 = [0, 1, 2, 3, 4]  
A_3 = outer(a_3, b_3, +) .%5
```

```
[190]: 5x5 Matrix{Float64}:  
 0.0  1.0  2.0  3.0  4.0  
 1.0  2.0  3.0  4.0  0.0  
 2.0  3.0  4.0  0.0  1.0  
 3.0  4.0  0.0  1.0  2.0  
 4.0  0.0  1.0  2.0  3.0
```

Задание 8. Пункт 2 (3)

```
[192]: a_4 = [i for i in 0:9]
       b_4 = [i for i in 0:9]
       A_4 = outer(a_4, b_4, +) .%10
```

```
[192]: 10x10 Matrix{Float64}:
 0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0
 1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0
 2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0
 3.0  4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0
 4.0  5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0
 5.0  6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0
 6.0  7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0
 7.0  8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0
 8.0  9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0
 9.0  0.0  1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0
```

```
[199]: a_5 = [i for i in 0:8]
       b_5 = [i for i in 9:-1:1]
       A_5 = (outer(a_5, b_5, +) .%9)
```

```
[199]: 9x9 Matrix{Float64}:
 0.0  8.0  7.0  6.0  5.0  4.0  3.0  2.0  1.0
 1.0  0.0  8.0  7.0  6.0  5.0  4.0  3.0  2.0
 2.0  1.0  0.0  8.0  7.0  6.0  5.0  4.0  3.0
 3.0  2.0  1.0  0.0  8.0  7.0  6.0  5.0  4.0
 4.0  3.0  2.0  1.0  0.0  8.0  7.0  6.0  5.0
 5.0  4.0  3.0  2.0  1.0  0.0  8.0  7.0  6.0
 6.0  5.0  4.0  3.0  2.0  1.0  0.0  8.0  7.0
 7.0  6.0  5.0  4.0  3.0  2.0  1.0  0.0  8.0
 8.0  7.0  6.0  5.0  4.0  3.0  2.0  1.0  0.0
```

Задание 9 (1)

9. Решите следующую систему линейных уравнений с 5 неизвестными:

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

рассмотрев соответствующее матричное уравнение $Ax = y$. Обратите внимание на особый вид матрицы A . Метод, используемый для решения данной системы уравнений, должен быть легко обобщаем на случай большего числа уравнений, где матрица A будет иметь такую же структуру.

```
[349]: M = 5
A = zeros(M, M)
for k = 1:M
    for i = k:M
        A[k, i] = i - k + 1
        A[i, k] = i - k + 1
    end
end
A
```

```
[349]: 5x5 Matrix{Float64}:
 1.0  2.0  3.0  4.0  5.0
 2.0  1.0  2.0  3.0  4.0
 3.0  2.0  1.0  2.0  3.0
 4.0  3.0  2.0  1.0  2.0
 5.0  4.0  3.0  2.0  1.0
```

```
[350]: b = [7, -1, -3, 5, 17]
```

```
[350]: 5-element Vector{Int64}:
 7
 -1
 -3
 5
 17
```

Рис. 35: Задание 9. Матрица коэффициентов

Задание 9 (2)

```
[351]: function Gauss_Straight(A)
        for i = 1:(size(A)[1])
            for k = (i+1):(size(A)[2]-1)
                A[k, :] = A[k, :] - A[i, :]*(A[k, i] / A[i, i])
            end
        end
        return A
    end

    function Gauss_Reverse(_A)
        x = zeros(size(_A)[1])
        for i = (size(_A)[1]):-1:1
            b = _A[i, size(_A)[2]]
            for j = (size(_A)[1]):-1:i
                if j != i
                    b -= x[j]*_A[i, j]
                end
            end
            x[i] = b / _A[i, i]
        end
        return x
    end

    function Gauss(A, b)
        A_ = hcat(A, b)
        _A_ = Gauss_Straight(A_)
        X = Gauss_Reverse(_A_)
        return X
    end
```

[351]: Gauss (generic function with 2 methods)

Задание 9 (3)

```
[352]: _A = hcat(A, b)
        #print(size(_A))
        A__ = Gauss_Straight(_A)
```

```
[352]: 5x6 Matrix{Float64}:
        1.0  2.0  3.0  4.0  5.0  7.0
        0.0 -3.0 -4.0 -5.0 -6.0 -15.0
        0.0  0.0 -2.66667 -3.33333 -4.0 -4.0
        0.0  0.0  0.0 -2.5 -3.0  7.0
        0.0  0.0  0.0  0.0 -2.4  9.6
```

```
[353]: X = Gauss_Reverse(A__)
```

```
[353]: 5-element Vector{Float64}:
        -2.0000000000000044
         3.000000000000002
         4.9999999999999964
         2.0000000000000044
        -4.000000000000002
```

```
[354]: x = round.(X)
```

```
[354]: 5-element Vector{Float64}:
        -2.0
         3.0
         5.0
         2.0
        -4.0
```

```
[355]: x2 = round.(Gauss(A, b))
```

```
[355]: 5-element Vector{Float64}:
        -2.0
         3.0
         5.0
         2.0
        -4.0
```

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

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

```
[260]: 6x10 Matrix{Int64}:  
 7 10 1 1 7 4 1 6 2 1  
 7 4 6 6 6 8 5 5 3 3  
 6 8 9 8 7 4 2 8 4 2  
10 10 7 7 10 7 4 3 5 5  
 6 1 7 3 3 2 5 3 5 3  
 8 6 8 3 10 6 3 5 6 1
```

Рис. 38: Задание 10. Создание матрицы со случайными целыми числами

Задание 10. Пункт 1

– Найдите число элементов в каждой строке матрицы M , которые больше числа N_1 (например, $N_1 = 4$).

```
[261]: N1 = 4
Numbers_1 = Int.(zeros(6))
for i ∈ 1:6
    for j ∈ 1:10
        if (M[i, j] > N1)
            Numbers_1[i] += 1
        end
    end
end
Numbers_1
```

```
[261]: 6-element Vector{Int64}:
 4
 7
 6
 8
 4
 7
```

Рис. 39: Задание 10. Пункт 1.

Задание 10. Пункт 2

– Определите, в каких строках матрицы M число N_2 (например, $N_2 = 7$) встречается ровно 2 раза?

```
[264]: N2 = 7
Numbers_2 = Int.(zeros(6))
for i ∈ 1:6
    for j ∈ 1:10
        if (M[i, j] == N2)
            Numbers_2[i] += 1
        end
    end
end
print(Numbers_2)
findall(x -> x == 2, Numbers_2)
```

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

```
[264]: 1-element Vector{Int64}:
       1
```

Рис. 40: Задание 10. Пункт 2.

Задание 10. Пункт 3

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

```
[263]: K = 75
Numbers_3 = []
for i ∈ 1:9
    for j ∈ (i+1):10
        if (sum(M[:, i]) + sum(M[:, j]) > K)
            push!(Numbers_3, (i, j))
        end
    end
end
Numbers_3
```

```
[263]: 6-element Vector{Any}:
 (1, 2)
 (1, 3)
 (1, 5)
 (2, 3)
 (2, 5)
 (3, 5)
```

Рис. 41: Задание 10. Пункт 3.

11. Вычислите:

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

```
[265]: S_1 = 0
      for i ∈ 1:20
        for j ∈ 1:5
          S_1 += i^4 / (3 + j)
        end
      end
      S_1
```

```
[265]: 639215.2833333334
```

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

```
[266]: S_2 = 0
      for i ∈ 1:20
        for j ∈ 1:5
          S_2 += i^4 / (3 + i*j)
        end
      end
      S_2
```

```
[266]: 89912.02146097136
```

Результаты

В ходе работы я освоил работу с циклами и сторонние библиотеки Julia для решения задач линейной алгебры и работы с матрицами