

# Компьютерный практикум по статистическому анализу данных

## презентация к лабораторной работе №3

Управляющие структуры

---

Ким И. В. НФИбд-01-21

Российский университет дружбы народов, Москва, Россия

## Цель работы

---

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

## Задание

---

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

Выполнение работы. Повтор примеров  
из раздела 3.2

---

### 3.2.1. Циклы While и for

---

Цикл, который пока  $n < 10$  прибавляет к  $n$  единицу и печатает значение

### 3.2.1 Циклы While и for

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

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

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

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

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



Задали массив со строковыми элементами. Далее в цикле подставляется имя из массива в заданную строку и выводит на экран.

```
6]: for n in 1:2:10
    println(n)
end
myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]

for friend in myfriends
    println("Hi $friend, it's grear to see you!")
end

1
3
5
7
9
Hi Ted, it's grear to see you!
Hi Robyn, it's grear to see you!
Hi Barney, it's grear to see you!
Hi Lily, it's grear to see you!
Hi Marshall, it's grear to see you!

8]: m,n = 5, 5
A=fill{0,(m,n)}

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
```

Использование цикла for для создание двумерного массива, в котором значение каждой записи является суммой индексов строки и столбца

```
4]: B = fill(0,(m,n))  
  
for i in 1:m, j in 1:n  
    B[i,j] = i+j  
end  
B
```

```
4]: 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
```

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

```
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
```

### 3.2.2. Условные выражения

---

Пример, в котором используется условие: если N делится на 3 выводится “Fizz”, если на 5 “Buzz”, если на 3 и на 5 “FizzBuzz”

### 3.2.2 Условные выражения

```
[48]: N=15
      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
```

FizzBuzz

### 3.2.3. Функции

---

Написал функцию, при вызове которой подставляет принятое имя в строку.  
Вторая функция - функция возведения в квадрат

### 3.2.3. Функции

```
[61]: function sayhi(name)
      println("Hi $name, it's great to see you!")
      end

[61]: sayhi (generic function with 1 method)|

[63]: sayhi("Ilya")
      hi Ilya, it's great to see you!

[65]: function f(x)
      x^2
      end

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

[67]: f(5)
[67]: 25

[71]: sayhi2(name) = println("Hi $name, it's great to see you!")
[71]: sayhi2 (generic function with 1 method)

[75]: sayhi2("Ilya")
      Hi Ilya, it's great to see you!
```

Написал функцию, при вызове которой подставляет принятое имя в строку.  
Вторая функция - функция возведения в квадрат

```
[77]: f2(x) = x^2
[77]: f2 (generic function with 1 method)
[79]: f2(2)
[79]: 4
[81]: sayhi3 = name -> println("Hi $name, it's great to see you!")
      f3 = x -> x^2
[81]: #9 (generic function with 1 method)
[83]: sayhi3("Ilya")
      Hi Ilya, it's great to see you!
[89]: f3(6)
[89]: 36
```

Задал массив  $v$  и отсортировал его функцией  $\text{sort}(v)$  и  $\text{sort}!(v)$ . Разница в них была в том, что  $\text{sort}!(v)$  меняли значения в нашем массиве, а  $\text{sort}(v)$  - нет. Функция  $\text{map}()$  применила функцию  $f$  ко всем элементам массива

```
[105]: v = [3,5,2]
      sort(v)
      v

[105]: 3-element Vector{Int64}:
       3
       5
       2

[101]: sort!(v)
      v

[101]: 3-element Vector{Int64}:
       2
       3
       5

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

[107]: 3-element Vector{Int64}:
       1
       4
       9

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

[111]: 3-element Vector{Int64}:
       1
       8
      27
```



Применил функцию broadcast, которая привела все объекты к общему измерению, в отличие от функции map, которая напрямую применяет функцию поэлементно. Задал матрицу A и применили к ней функцию  $f(A)$  возведения в квадрат, получил  $A^*A$ . Далее применил функцию  $f.(A)$ , получил матрицу, где каждый элемент возведен в квадрат.

```
[119]: broadcast(f, [1,2,3])
[119]: 3-element Vector{Int64}:
       1
       4
       9

[121]: f.([1,2,3])
[121]: 3-element Vector{Int64}:
       1
       4
       9

[125]: A = [i+3*j for j in 0:2, i in 1:3]
[125]: 3x3 Matrix{Int64}:
       1  2  3
       4  5  6
       7  8  9

[127]: f(A)
[127]: 3x3 Matrix{Int64}:
      30  36  42
      66  81  96
     102 126 150

[129]: B = f.(A)
[129]: 3x3 Matrix{Int64}:
       1  4  9
      16 25 36
      49 64 81
```

Попробовал пример записей с синтаксисом broadcast.

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

```
[135]: 3×3 Matrix{Float64}:  
 3.0  6.0  9.0  
12.0 15.0 18.0  
21.0 24.0 27.0
```

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

```
[137]: 3×3 Matrix{Float64}:  
 3.0  6.0  9.0  
12.0 15.0 18.0  
21.0 24.0 27.0
```

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

```
[139]: 3×3 Matrix{Float64}:  
 3.0  6.0  9.0  
12.0 15.0 18.0  
21.0 24.0 27.0
```

### 3.2.4 Сторонние библиотеки (пакеты) в Julia

---

## Загрузил пакеты и попробовал их использовать

### 3.2.4. Сторонние библиотеки (пакеты) в Julia

```
[142]: import Pkg
```

```
[144]: Pkg.add("Example")
```

```
Updating registry at `C:\Users\ksudz\.julia\registries\General.toml`  
Resolving package versions...  
Installed Example - v0.5.5  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Project.toml`  
[70/00007] + Example v0.5.5  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Manifest.toml`  
[70/00007] + Example v0.5.5  
Precompiling project...  
✓ Example  
1 dependency successfully precompiled in 1 seconds. 17 already precompiled.
```

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

```
Resolving package versions...  
Installed FixedPointNumbers - v0.8.5  
Installed ColorTypes - v0.12.0  
Installed Colors - v0.13.0  
Installed Reexport - v1.2.2  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Project.toml`  
[50000000] + Colors v0.13.0  
Updating `C:\Users\ksudz\.julia\environments\v1.10\Manifest.toml`  
[50000077] + ColorTypes v0.12.0
```

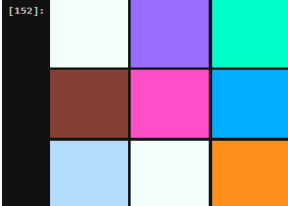
## Загрузил пакеты и попробовал их использовать

.

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



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



### 3.4 Задания для самостоятельного выполнения

---

## Задание 1

---

# Использовал цикл for для печати чисел от 1 до 100 и их квадратов.

## 1. Используя циклы while и for:

### 1.1. Выведите на экран целые числа от 1 до 100 и напечатайте их квадраты

```
1) for i in 1:100
    print(i, " ")
end
println("\n\n")
for i in 1:100
    print(i^2, " ")
end

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

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 324
9 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

2) i=1
j=1
while i<=101
    print(i, " ")
    i+=1
end
println("\n\n")
while j<=101
    print(j^2, " ")
    j+=1
end

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83
84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

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 324
9 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
```



Объявил, что `squares` - это словарь. Использовал цикл `for` от 1 до 100, где присвоил целые числа к ключам, а их квадраты в качестве пар-значений.

## 1.2. Создайте словарь `squares`, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений

```
[5]: squares = Dict{  
    for i in 1:100  
        squares[i] = i^2  
    end  
    squares
```

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

```
 5 => 25  
36 => 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  
[ ] => [ ]
```

Сделал тоже самое, используя цикл while

```
[23]: squares1 = Dict()
      i=1
      while i!=101
          squares1[i]=i^2
          i+=1
      end
      squares1

[23]: 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
      ⋮ => ⋮
```

Задал массив нулей `squares_arr` размерностью 100, использовал цикл `for` для возведения `i` в квадрат, записи в массив и вывода на экран. Сделал тоже самое, используя цикл `while`.

### 1.3. Создайте массив `squares_arr`, содержащий квадраты всех чисел от 1 до 100

```
[7]: squares_arr = zeros(Int64,100)
    for i in 1:100
        squares_arr[i] = i^2
        print(squares_arr[i], " ")
    end
```

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 6891 7056 7225 7396 7569 7744 7921 8100 8281 8464 8649 8836 9025 9216 9409 9604 9801 10000

```
[25]: squares_arr1 = zeros(Int64,100)
    i=1
    while i!=101
        squares_arr1[i] = i^2
        print(squares_arr1[i], " ")
        i+=1
    end
```

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 6891 7056 7225 7396 7569 7744 7921 8100 8281 8464 8649 8836 9025 9216 9409 9604 9801 10000

## Задание 2

---

Использовал цикл for от 1 до 10 с условием: если остаток от деления  $i$  на 2 равен нулю, то выводит на экран, что число  $i$  чётное, в обратном случае - нечётное

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

```
for i in 1:10
    if i % 2 == 0
        println("Число ", i, " - чётное")
    else if i % 2 != 0
        println("Число ", i, " - нечётное")
    end
end
```

Число 1 - нечётное  
Число 2 - чётное  
Число 3 - нечётное  
Число 4 - чётное  
Число 5 - нечётное  
Число 6 - чётное  
Число 7 - нечётное  
Число 8 - чётное  
Число 9 - нечётное  
Число 10 - чётное

## Задание 3

---

Написал функцию `add_one`, которая прибавляет единицу к своему входному значению.

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

```
[47]: function add_one(x)
      x+=1
      end
      add_one(5)
```

```
[47]: 6
```

## Задание 4

---



Задал матрицу  $A$  размерностью  $3 \times 3$ , использовал функцию `broadcast`, в которой прибавил ко всем элементам матрицы единицу. Сделал тоже самое, используя функцию `map()` и `add_one()`.

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

```
{65}: A = [i+j for j in 0:2, i in 1:3]
```

```
{65}: 3x3 Matrix{Int64}:
```

```
 1  2  3
 6  7  8
11 12 13
```

```
{70}: broadcast(x=0x1,A)
```

```
{70}: 3x3 Matrix{Int64}:
```

```
 2  3  4
 7  8  9
12 13 14
```

```
{84}: map(add_one,A)
```

```
{84}: 3x3 Matrix{Int64}:
```

```
 2  3  4
 7  8  9
12 13 14
```

## Задание 5

---

Задал матрицу  $A$  и функцию возведения в куб  $f$ , применил функцию  $f(A)$  и получил матрицу нулей. Во втором пункте использовал цикл `for` от 3 до 9 с шагом 3, в котором присваивал к 3, 6 и 9 элементам сумму первого и второго.

5. Задайте матрицу  $A$  следующего вида:

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

- Найдите  $A^3$

```
106]: A = [[1, 5, -2] [1, 2, -1] [3, 6, -3]]
      f(x)=x^3
      f(A)
```

```
106]: 3x3 Matrix(Int64):
      0  0  0
      0  0  0
      0  0  0
```

- Замените третий столбец матрицы  $A$  на сумму второго и третьего столбцов

```
123]: for i in 3:3:9
      A[i]=A[i-2]+A[i-1]
      end
      A
```

```
123]: 3x3 Matrix(Int64):
      1  1  3
      5  2  6
      6  3  9
```

## Задание 6

---

Создал матрицу  $B$  используя функцию `repeat`, и нашел произведение транспонированной  $B$  на  $B$

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

```
[179]: B = repeat([10, -10, 10], 5, 3)
```

```
[179]: 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
 10  10  10
```

```
[181]: B'
```

```
[181]: 3x15 adjoint(::Matrix{Int64}) with eltype 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
```

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

```
[183]: 3x3 Matrix{Int64}:
```

```
1500 1500 1500
1500 1500 1500
1500 1500 1500
```

## Задание 7

---

Создал матрицу  $Z$  размерности  $6 \times 6$ , где все элементы равны нулю и матрицу  $E$ , где все элементы равны 1

7. Создайте матрицу  $Z$  размерности  $6 \times 6$ , все элементы которой равны нулю, и матрицу  $E$ , все элементы которой равны 1.

```
[194]:
```

```
Z = zeros(6,6)
```

```
display(Z)
```

```
E = ones(6,6)
```

```
6x6 Matrix{Float64}:
```

```
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
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
```

```
[194]:
```

```
6x6 Matrix{Float64}:
```

```
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0
```

Задал 4 матрицы заполненные нулями размерностью  $6 \times 6$ , далее нашел закономерности, по которым их нужно привести к нужным матрицам, и используя цикл `for` и условие `if` выполнил задание.



# Матрица Z1 и Z2

- Используя цикл while или for и закономерности расположения элементов, создайте следующие матрицы размерности 6 × 6:

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

```
[255]: Z1 = zeros(Int64,6,6)
      Z2 = zeros(Int64,6,6)
      Z3 = zeros(Int64,6,6)
      Z4 = zeros(Int64,6,6)
      for i in 1:6, j in 1:6
          if i == j+1 || i == j-1
              Z1[i,j]=1
          end
      end
      Z1
```

```
[255]: 6x6 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
```

```
[261]: for i in 1:6, j in 1:6
        if i==j || i==j+2 || i==j-2
            Z2[i,j] = 1
        end
    end
    Z2
```

```
[261]: 6x6 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
```

```
[259]: for i in 1:6, j in 1:6
        if i==7-j || i==5-j || i== 9-j
            Z3[i,j] = 1
        end
    end
    Z3
```

```
[259]: 6x6 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
```

```
[263]: for i in 1:6, j in 1:6
        if i%2!=0 && j%2!=0
            Z4[i,j] = 1
        elseif i%2==0 && j%2==0
            Z4[i,j] = 1
        end
    end
    Z4
```

```
[263]: 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
```

## Задание 10

---

Используя функцию `rand`, создал матрицу  $M$  размерности  $6 \times 10$ , элементами которой являются целые числа, выбранные случайным образом из диапазона  $1, 2, \dots, 10$ . Используя цикл `for` перебрал всю матрицу и нашел количество элементов, которые больше 4

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,  $T = 4$ ).

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

```
[366]: 6x10 Matrix{Int64}:
```

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

```
[370]: k=0
N=4
for i in 1:size(M)[1], j in 1:size(M)[2]
    if M[i,j] > N
        k+=1
    end
end
k
```

```
[370]: 34
```

Задал  $P=7$ , использовал цикл `for` от 1 до размерности матрицу по строкам (6), ввел счетчик  $z$ , который будет обнуляться при движении  $i$ , ввел второй цикл  $j$  от 1 до размерности матрицы по столбцам (10), ввел условие, что если элемент матрицы равен  $P$ , то срабатывает счетчик. После того, как мы прошли один круг цикла  $j$  проверяется условие, что если  $z==2$ , то напечатать строку. И так проверяется каждая строка.

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

```
[374]: P=7
for i in 1:size(M)[1]
    z=0
    for j in 1:size(M)[2]
        if M[i,j] == P
            z+=1
        end
    end
    if z==2
        println("В строке №",i, " число ", P, " встречается ровно 2 раза")
    end
end
```

В строке №1 число 7 встречается ровно 2 раза

Использовал цикл for для нахождения суммы элементов каждого столбца, далее присваиваю это значение к новому массиву A и веду счетчик для порядка элементов в массиве A. После использую еще один цикл, в котором перебираю элементы массива A и ищу пары столбцов, сумма элементов которых будет больше 75, если она больше, то печатает на экран.

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

```
[475]: K = 75
z=1
A = zeros(Int64,size(M)[2])
display(M)
for i in 1:size(M)[2]
    G=0
    for j in 1:size(M)[1]
        G = G+M[j,i]
    end
    A[z]=G
    z+=1
end
for i in 1:size(A)[1]
    for j in 1:size(A)[1]
        if A[i]+A[j]>75
            println("Сумма столбцов M",i," и M",j," больше 75")
        end
    end
end
```

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

```
Сумма столбцов M2 и M9 больше 75
Сумма столбцов M4 и M9 больше 75
Сумма столбцов M5 и M9 больше 75
Сумма столбцов M6 и M6 больше 75
Сумма столбцов M6 и M7 больше 75
Сумма столбцов M6 и M8 больше 75
Сумма столбцов M6 и M9 больше 75
Сумма столбцов M7 и M6 больше 75
Сумма столбцов M7 и M9 больше 75
Сумма столбцов M8 и M6 больше 75
Сумма столбцов M8 и M9 больше 75
Сумма столбцов M9 и M2 больше 75
Сумма столбцов M9 и M4 больше 75
Сумма столбцов M9 и M5 больше 75
Сумма столбцов M9 и M6 больше 75
Сумма столбцов M9 и M7 больше 75
Сумма столбцов M9 и M8 больше 75
Сумма столбцов M9 и M9 больше 75
Сумма столбцов M9 и M10 больше 75
Сумма столбцов M10 и M6 больше 75
Сумма столбцов M10 и M9 больше 75
```

## Задание 11

---



## 11. Вычислите:

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

```
[471]: sum([sum([i^4/(3+j)] for j in 1:5) for i in 1:20])
```

```
[471]: 1-element Vector{Float64}:  
639215.2833333334
```

```
[473]: sum([sum([i^4/(3+i*j)] for j in 1:5) for i in 1:20])
```

```
[473]: 1-element Vector{Float64}:  
89912.02146097136
```

## Выводы

---

Используя Jupyter lab повторил примеры из раздела 3.2 и выполнил задания для самостоятельной работы.