

# Лабораторная работа №3

Компьютерный практикум по статистическому анализу данных

---

Амуничников Антон Игоревич

2025-10-08

1. Информация
2. Вводная часть
3. Выполнение лабораторной работы
4. Задания самостоятельной работы

# 1. Информация

---

## 1.1 Докладчик

- Амуничников Антон Игоревич
- Группа: НПИбд-01-22
- Российский университет дружбы народов им. П. Лумумбы
- 1132227133@pfur.ru

## **2. Вводная часть**

---

## 2.1 Цель работы

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

## 2.2 Задание

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

### **3. Выполнение лабораторной работы**

---



## 3.1 Примеры

```
25) n = 0
    while n < 10
        n+=1
        println(n)
    end
1
2
3
4
5
6
7
8
9
10

26) for n in 1:2:10
    println(n)
end
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
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!
```

**Рисунок 1:** Выполнение примеров с циклами

## 3.2 Примеры

```

# initialize, it's great to see you.

[6]: # 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
print(A, "\n")
# 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
print(B, "\n")

C = [i + j for i in 1:m, j in 1:n]
print(C, "\n")

[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]
[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]
[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]
```

```

[7]: # используем '&&' для реализации операции "AND"
# операция % вычисляет остаток от деления
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
```

## 3.3 Примеры

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

[8]: 10

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

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

[10]: sayhi("C-3P0")
      f(42)

      Hi C-3P0, it's great to see you!

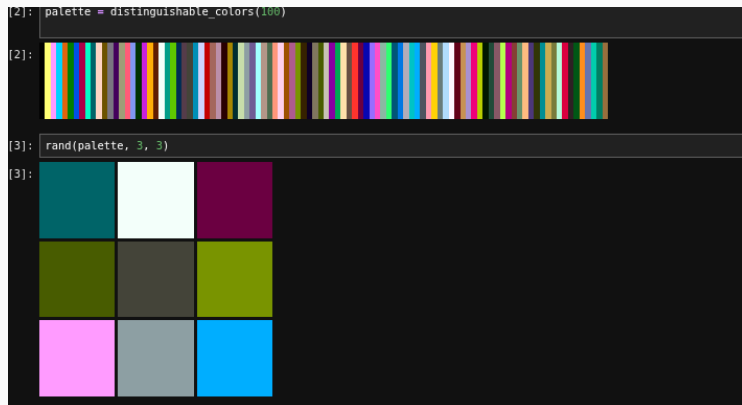
[10]: 1764

[1]: import Pkg
      Pkg.add("Colors")
      using Colors

      Resolving package versions...
      No Changes to `~/julia/environments/v1.11/Project.toml`
      No Changes to `~/julia/environments/v1.11/Manifest.toml`
```

Рисунок 3: Выполнение примеров с функциями

## 3.4 Примеры



**Рисунок 4:** Выполнение примеров со сторонними библиотеками

## **4. Задания самостоятельной работы**

---

## 4.1 Задание №1

```
[4]: print([i for i in 1:100])
print("\n", "Квадраты")
print("\n", [i**2 for i in 1:100])

[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]
Квадраты
[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, 6889, 7056, 7225, 7396, 7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9604, 9801, 10000]

[5]: squares = Dict{<
for i in 1:10
    squares[i] = i^2
end
print(squares)

Dict{Any, Any}[5 => 25, 4 => 16, 6 => 36, 7 => 49, 2 => 4, 10 => 100, 9 => 81, 8 => 64, 3 => 9, 1 => 1]

[6]: squares_arr = [i^2 for i in 1:100]
show(squares_arr)

[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, 6889, 7056, 7225, 7396, 7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9604, 9801, 10000]
```

Рисунок 5: Задание №1

## 4.2 Задание №2

- Напишем условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишем код, используя тернарный оператор.

```
18): a = readline()
    a = parse(Int64, a)

    if a%2 == 0
        print(a)
    else
        print("нечетное")
    end

stdIn> 3
нечетное

19): a = readline()
    a = parse(Int64, a)
    (a % 2 == 0) ? a : "нечетное"

stdIn> 4
4
```

**Рисунок 6:** Задание №2

## 4.3 Задание №3

- Напишем функцию `add_one`, которая добавляет 1 к своему входу.

```
[46]: function add_one(number)
      number + 1
      end

[46]: add_one (generic function with 1 method)

[47]: add_one(5)

[47]: 6
```

**Рисунок 7:** Задание №3



## 4.4 Задание №4

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

```
[49]: A
[49]: 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
[50]: map(x -> (x + 1), A)
[50]: 5x5 Matrix{Int64}:
      3  4  5  6  7
```

## 4.5 Задание №5

- Зададим матрицу  $A$ . Найдем  $A^3$ . Заменим третий столбец матрицы  $A$  на сумму второго и третьего столбцов.

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

[51]: 3x3 Matrix{Int64}:
 1  1  3
 5  2  6
-2 -1 -3

[55]: # Найдите A^3
g(x) = x^3
B = g.(A)

[55]: 3x3 Matrix{Int64}:
 1  1  27
125  8  216
-8 -1 -27

[56]: # Замените третий столбец матрицы A на сумму второго и третьего столбцов
for i in 1:3
    A[i, 3] = A[i, 2] + A[i, 3]
end
A

[56]: 3x3 Matrix{Int64}:
 1  1  4
 5  2  8
-2 -1 -4
```

Рисунок 9: Задание №5

## 4.6 Задание №6

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

```
-2 -1 -4  
5]: B = [j % 2 == 0 ? -10 : 10 for i in 1:15, j in 1:3]  
C = B' * B  
5]: 3x3 Matrix{Int64}:  
 1500  -1500  1500  
 -1500  1500  -1500  
 1500  -1500  1500
```

Рисунок 10: Задание №6

## 4.7 Задание №7

```
(81): Z = fill(0, 4, 4)
      F = fill(1, 4, 4)

      Z1 = copy(Z)
      for i in 1:4, j in 1:4
          if abs(i - j) == 1
              Z1[i, j] = 1
          end
      end
      Z1

(82): G=0 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

(83): Z2 = copy(Z)
      for i in 1:4, j in 1:4
          if i == j || abs(i - j) == 2
              Z2[i, j] = 1
          end
      end
      Z2

(84): G=0 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

(85): Z3 = copy(Z)
      for i in 1:4, j in 1:4
          if (i - j + 1) == 3 || abs(i - i + 1 - j) == 2
              Z3[i, j] = 1
          end
      end
      Z3

(86): G=0 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

(87): Z4 = copy(Z)
      for i in 1:4, j in 1:4
          if (i - j) % 3 == 0
              Z4[i, j] = 1
          end
      end
      Z4

(88): G=0 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
```

Рисунок 11: Задание №7

## 4.8 Задание №8

```
[92]: function outer(x,y,operation)
    if (operation == "+")
        res = x + y
    elseif (operation == "-")
        res = x - y
    elseif (operation == ".")
        res = x * y
    elseif (operation == "/")
        res = x / y
    end
end

[92]: outer (generic function with 1 method)

[93]: A1 = outer([1, 2, 3, 4], 5, "+")
A1
[93]: 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

[94]: A2 = outer([1, 2, 3], [1, 2, 3, 4], "**")
A2
[94]: 5x5 Matrix{Int64}:
 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

[95]: A3 = outer([1, 2, 3, 4], 5, "%")
A3.res = outer(A3, 5, "%")

[95]: 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
```

**Рисунок 12:** Задание №8. Реализация функции `outer()`. Проверка работы функции

## 4.9 Задание №8

```
[97]: A4 = outer(0:9, 0:9, "+")
      A4_res = outer(A4, 10, "%")

[97]: 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

[104]: A5 = outer(0:8, 0:8, "-")
      A5_res = mod.(A5, 9)

[104]: 9x9 Matrix{Int64}:
      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

[108]: 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
      println("Решение системы: ")
      println(x)

      Решение системы:
```

## 4.10 Задание №9

```
008]: 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  
println("Решение системы: ")  
println(x)  
  
Решение системы:  
[-2.0000000000000013, 3.0000000000000027, 4.9999999999999964, 2.0000000000000027, -4.000000000000001]
```

**Рисунок 14:** Задание №9. Решение систему линейных уравнений

## 4.11 Задание №10

```
109]: M = rand(1:10, 6, 10)
109]: 6×10 Matrix{Int64}:
      3  8  5  3  8  2  9  8  1  5
      9  1  1  1  1  4 10  6  1  3
      7  6  9  2  5 10  2  7  5  9
      2  9  6  6  1  9  2  9  5  9
      1  6  3  5  8 10 10  5 10  2
      6  1  6  4  9  6  1  7  3  6
112]: N = 4
      count_1 = sum(M.>N)
112]: 37
114]: count_2 = [i for i = 1:6 if sum(M[i,:].==7)==2]
114]: 1-element Vector{Int64}:
      3
118_: K = 70
      count_3 = [(i, j) for i = 1:6, j = 2:5 if (i != j && sum(M[:,i] + M[:,j]) > K)]
118]: 3-element Vector{Tuple{Int64, Int64}}:
      (6, 2)
      (6, 3)
      (6, 5)
119]: sum1 = sum(i^4 / (3 + j) for i in 1:20, j in 1:5)
      sum2 = sum(i^4 / (3 + i*j) for i in 1:20, j in 1:5)
      println("sum1 = ", sum1)
      println("sum2 = ", sum2)
      sum1 = 639215.2833333338
      sum2 = 89912.02146097131
```

Рисунок 15: Задание №10. Задание №11



## 4.12 Выводы

- В результате выполнения данной лабораторной работы я освоил применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.