

Лабораторная работа №3

Компьютерный практикум по статистическому анализу данных

Канева Екатерина, НФИбд-02-22

11 октября 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Канева Екатерина Павловна
- студент группы НФИбд-02-22
- Российский университет дружбы народов
- 1132222004@rudn.ru
- <https://nevseros.github.io/ru/>

Вводная часть

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

- Используя Jupyter Lab, повторить примеры.
- Выполнить задания для самостоятельной работы.

Выполнение работы

Выполнила примеры с циклами:

```
[1]: n = 8
while n > 10
    n += 1
    println(n)
end

1
2
3
4
5
6
7
8
9
10

[2]: myFriends = ["Ted", "Noby", "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 Noby, 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!

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

1
3
5
7
9

[4]: myFriends = ["Ted", "Noby", "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 Noby, 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. 1: Примеры с циклами.

Примеры с условными выражениями

Выполнила примеры с условными выражениями:

```
[17]: N = 12  
  
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  
Fizz
```

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

```
[22]: 10
```

Рис. 2: Примеры с условными выражениями.

Выполнила примеры с функциями:

```
[25]: function f(x)
      x^2
      end

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

[27]: f(42)

[27]: 1764

[28]: f2(x) = x^2

[28]: 1764

[29]: f3 = x -> x^2

[30]: g1 (generic function with 1 method)

[31]: v = [1, 2, 3]

[31]: 3-element Vector{Int64}:
       1
       2
       3

[32]: sort!(v)

[32]: 3-element Vector{Int64}:
       1
       2
       3

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

[33]: 3-element Vector{Int64}:
       1
       4
       9

[34]: f(x) = x^2
      @broadcast f, [1, 2, 3]

[34]: 3-element Vector{Int64}:
       1
       4
       9
```

Рис. 3: Примеры с функциями.

Примеры со сторонними библиотеками

Выполнила примеры со сторонними библиотекам:

```
[44]: Import Pig
[45]: Pig.addColors()
using Colors

Resolving package versions...
  Installed ColorTypes — v0.12.1
  Installed BasePlot — v1.2.2
  Installed PlotsMathNumbers — v0.8.0
  Installed Statistics — v1.12.1
  Installed Colors — v0.13.1
Updating 'C:\Users\alexeenw\Julia\environments\v0.11\Project.toml'
[Succ0895] + Colors v0.13.1
Updating 'C:\Users\alexeenw\Julia\environments\v0.11\Manifest.toml'
[Succ0897] + ColorTypes v0.12.1
[Succ0898] + Colors v0.13.1
[Succ0899] + PlotsMathNumbers v0.8.0
[Succ0900] + BasePlot v1.2.2
[Succ0901] + Statistics v1.12.1
[Succ0902] + LinearAlgebra v1.12.0
[Succ0903] + CompilerSupportLibraries_jll v0.12.1+0
[Succ0904] + OpenBLAS_jll v0.3.21+0
[Succ0905] + libmkl_fft v1.1.0+1.0.1
[46]: palette = distinguishable_colors(100)
[47]:
[48]:
[49]: rand(palette, 3, 3)
[50]:
```

Рис. 4: Примеры со сторонними библиотеками.

Задание для самостоятельной работы 1

Выполнила первое задание для самостоятельной работы:

```
[1]: print([i for i in 1:100], '\n')
print([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]

[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{[i => i**2 for i in 1:100]}
print(squares)

Dict{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, 23 => 529, 92 => 8464, 41 => 1681, 43 => 1849, 11 => 121, 36 => 1296, 68 => 4624, 69 => 4761, 98 => 9604, 82 => 6724, 85 => 7225, 39 => 1521, 84 => 7056, 77 => 5929, 7 => 49, 25 => 625, 95 => 9025, 71 => 5041, 66 => 4356, 76 => 5776, 34 => 1156, 50 => 2500, 59 => 3481, 93 => 8649, 2 => 4, 10 => 100, 18 => 324, 26 => 676, 27 => 729, 42 => 1764, 87 => 7569, 100 => 10000, 79 => 6241, 16 => 256, 20 => 400, 81 => 6561, 19 => 361, 49 => 2401, 44 => 1936, 9 => 81, 31 => 961, 74 => 5476, 61 => 3721, 29 => 841, 94 => 8836, 46 => 2116, 57 => 3249, 70 => 4900, 21 => 441, 38 => 1444, 88 => 7744, 78 => 6084, 72 => 5184, 24 => 576, 8 => 64, 17 => 289, 37 => 1369, 1 => 1, 53 => 2809, 22 => 484, 47 => 2209, 83 => 6889, 99 => 9801, 89 => 7921, 14 => 196, 3 => 9, 80 => 6400, 96 => 9216, 51 => 2601, 33 => 1089, 40 => 1600, 48 => 2304, 15 => 225, 65 => 4225, 97 => 9409}

[7]: squares_arr = [i**2 for i in 1:100]
print(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]
```

Рис. 5: Задание 1.

Задание для самостоятельной работы 2

Выполнила второе задание для самостоятельной работы:

```
[13]: a = parse(Int64, readline())  
  
      if a % 2 == 0  
        print(a)  
      else  
        print("нечётное")  
      end  
  
stdin> 5  
нечётное  
  
[15]: a = parse(Int64, readline())  
  
      a % 2 == 0 ? print(a) : print("нечётное")  
  
stdin> 5  
нечётное
```

Рис. 6: Задание 2.

Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы:

```
[17]: function add_one(x)
      x = x + 1
      end
      add_one(8.9)
[17]: 9.9
```

Рис. 7: Задание 3.

Задание для самостоятельной работы 4

Выполнила четвёртое задание для самостоятельной работы:

```
[23]: a = [1 2 3; 4 5 6; 7 8 9]
```

```
A = map(add_one, a)
```

```
[23]: 3x3 Matrix{Int64}:
```

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

Рис. 8: Задание 4.

Задание для самостоятельной работы 5

Выполнила пятое задание для самостоятельной работы:

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

```
[26]: 3x3 Matrix{Int64}:  
 1  1  3  
 5  2  6  
-2 -1 -3
```

```
[27]: A*3
```

```
[27]: 3x3 Matrix{Int64}:  
 3  3  9  
15  6 18  
-6 -3 -9
```

```
[28]: A[i, 3] = A[i, 3] + A[i, 2]
```

```
A
```

```
[29]: 3x3 Matrix{Int64}:  
 1  1  4  
 5  2  8  
-2 -1 -4
```

Рис. 9: Задание 5.

Задание для самостоятельной работы 6

Выполнила шестое задание для самостоятельной работы:

```
[39]: B = fill(10, (15, 3))  
  
      for i in 1:15  
        B[i, 2] = -10  
      end  
  
[40]: C = B' * B  
  
[40]: 3x3 Matrix{Int64}:  
      1500  -1500  1500  
      -1500  1500  -1500  
      1500  -1500  1500
```

Рис. 10: Задание 6.

Задание для самостоятельной работы 7

Выполнила седьмое задание для самостоятельной работы:

```
[100]: Z = fill(0, (6, 6))

[101]: end Matrix(Diagonal{
      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

[102]: K = fill(1, (6, 6))

[103]: end Matrix(Diagonal{
      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

[104]: Z1 = fill(0, (6, 6))

      for i in 1:6
        for j in 1:6
          if abs(i - j) == 1
            Z1[i, j] = 1
          end
        end
      end

      Z1

[105]: end Matrix(Diagonal{
      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

[106]: Z2 = fill(0, (6, 6))

      for i in 1:6
        for j in 1:6
          if abs(i - j) == 2 || i == j
            Z2[i, j] = 1
          end
        end
      end

      Z2
```

Рис. 11: Задание 7 (1).

Задание для самостоятельной работы 7

Выполнила седьмое задание для самостоятельной работы:

```
[92] 0x6 Matrix(2x664):  
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  
  
[93] Z3 = F111(0, (6, 6))  
  
for i in 1:6  
    for j in 1:6  
        if abs(i - 7 + 3) == 2 || i == 7 - 3  
            Z3[i, j] = 1  
        end  
    end  
end  
Z3  
  
[94] 0x6 Matrix(2x664):  
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  
  
[95] Z4 = F111(0, (6, 6))  
  
for i in 1:6  
    for j in 1:6  
        if abs(i - 3) == 2 || i == 3 || abs(i - 3) == 4  
            Z4[i, j] = 1  
        end  
    end  
end  
Z4  
  
[96] 0x6 Matrix(2x664):  
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. 12: Задание 7 (2).

Задание для самостоятельной работы 8

Выполнила восьмое задание для самостоятельной работы:

```
[70]: function outer(x, y, operation)
      res = fill(0, (length(x), length(y)))

      for i in 1:length(x)
        for j in 1:length(y)
          res[i, j] = operation(x[i], y[j])
        end
      end

      return res
    end
```

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

```
[72]: A1 = outer(0:4, 0:4, +)
```

```
[72]: S×S 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
```

```
[76]: A2 = outer(0:4, 1:5, ^)
```

```
[76]: S×S 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
```

```
[80]: A3 = outer(0:4, 0:4, (i, j) -> (i + j) % 5)
```

```
[80]: S×S 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
```

Рис. 13: Задание 8 (1).

Задание для самостоятельной работы 8

Выполнила восьмое задание для самостоятельной работы:

```
[04]: A4 = outer(0:9, 0:9, (i, j) -> (i + j) % 10)
```

```
[04]: 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
```

```
[05]: A5 = outer(0:8, 0:8, (i, j) -> (i - j + 9) % 9)
```

```
[05]: 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
```

Рис. 14: Задание 8 (2).

Задание для самостоятельной работы 9

Выполнила девятое задание для самостоятельной работы:

```
[110]: a = fill(0, (5, 5))  
  
      for i in 1:5  
        for j in 1:5  
          A[i, j] = abs(i - j) + 1  
        end  
      end  
  
      A  
  
[110]: 5x5 Matrix{Int64}:  
      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  
  
[112]: y = [7, -1, -3, 5, 17]:  
  
[116]: print(A \ y)  
      [-2.0000000000000036, 3.0000000000000058, 4.999999999999998, 1.9999999999999991, -3.999999999999999]
```

Рис. 15: Задание 9.

Задание для самостоятельной работы 10

Выполнила десятое задание для самостоятельной работы:

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

      N = 4
      Me = 7
      K = 75;

[125]: print(sum(M.>N))

      37

[133]: print([i for i in 1:6 if sum(M[i,:]>=7) == 2])

      [1]

[143]: print([(i, j) for i in 1:6, j in 2:5 if (i != j && sum(M[i,:]+M[:,j])>K)])

      [(1, 2), (3, 2), (4, 2), (5, 2), (6, 2), (1, 3), (2, 3), (4, 3), (5, 3), (6, 3), (2, 4), (3, 4), (2, 5), (3, 5)]
```

Рис. 16: Задание 10.

Задание для самостоятельной работы 11

Выполнила одиннадцатое задание для самостоятельной работы:

```
[145]: sum1 = sum(i^4 / (3 + j) for i in 1:20, j in 1:5)
[145]: 639215.2833333338

[147]: sum1 = sum(i^4 / (3 + j*i) for i in 1:20, j in 1:5)
[147]: 89912.02146097131
```

Рис. 17: Задание 11.

Заключение

Освоила применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.