

Компьютерный практикум по статистическому анализу данных

презентация к лабораторной работе №2

Структуры данных

Ким И. В. НФИбд-01-21

Российский университет дружбы народов, Москва, Россия

Цель работы

Основная цель работы — изучить несколько структур данных, реализованных в Julia, научиться применять их и операции над ними для решения задач.

Задание для самостоятельной работы

Задание для самостоятельной работы

2.4. Задания для самостоятельного выполнения

1. Даны множества: $A = \{0, 3, 4, 9\}$, $B = \{1, 3, 4, 7\}$, $C = \{0, 1, 2, 4, 7, 8, 9\}$. Найти $P = A \cap B \cup A \cap B \cup A \cap C \cup B \cap C$.
2. Приведите свои примеры с выполнением операций над множествами элементов разных типов.
3. Создайте разными способами:
 - 3.1) массив $(1, 2, 3, \dots, N-1, N)$, N выберите больше 20;
 - 3.2) массив $(N, N-1, \dots, 2, 1)$, N выберите больше 20;
 - 3.3) массив $(1, 2, 3, \dots, N-1, N, N-1, \dots, 2, 1)$, N выберите больше 20;
 - 3.4) массив с именем tmp вида $(4, 6, 3)$;
 - 3.5) массив, в котором первый элемент массива tmp повторяется 10 раз;
 - 3.6) массив, в котором все элементы массива tmp повторяются 10 раз;
 - 3.7) массив, в котором первый элемент массива tmp встречается 11 раз, второй элемент — 10 раз, третий элемент — 10 раз;
 - 3.8) массив, в котором первый элемент массива tmp встречается 10 раз подряд, второй элемент — 20 раз подряд, третий элемент — 30 раз подряд;
 - 3.9) массив из элементов вида $2^{tmp[i]}$, $i = 1, 2, 3$, где элемент $2^{tmp[3]}$ встречается 4 раза; посчитайте в полученном векторе, сколько раз встречается цифра 6, и выведите это значение на экран;
 - 3.10) вектор значений $y = e^x \cos(x)$ в точках $x = 3, 3.1, 3.2, \dots, 6$, найдите среднее значение y ;
 - 3.11) вектор вида (x^i, y^j) , $x = 0.1, i = 3, 6, 9, \dots, 36, y = 0.2, j = 1, 4, 7, \dots, 34$;
 - 3.12) вектор с элементами $\frac{2^i}{i}$, $i = 1, 2, \dots, M, M = 25$;
 - 3.13) вектор вида $(\text{"fn1"}, \text{"fn2"}, \dots, \text{"fnN"})$, $N = 30$;
 - 3.14) векторы $x = (x_1, x_2, \dots, x_n)$ и $y = (y_1, y_2, \dots, y_n)$ целочисленного типа длины $n = 250$ как случайные выборки из совокупности $0, 1, \dots, 999$; на его основе:
 - сформируйте вектор $(y_2 - x_1, \dots, y_n - x_{n-1})$;
 - сформируйте вектор $(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \dots, x_{n-2} + 2x_{n-1} - x_n)$;
 - сформируйте вектор $\left(\frac{\sin(y_1)}{\cos(x_2)}, \frac{\sin(y_2)}{\cos(x_3)}, \dots, \frac{\sin(y_{n-1})}{\cos(x_n)} \right)$;
 - вычислите $\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i + 10}$;

Задание для самостоятельной работы

Задание для самостоятельной работы

- выберите элементы вектора y , значения которых больше 600, и выведите на экран; определите индексы этих элементов;
 - определите значения вектора x , соответствующие значениям вектора y , значения которых больше 600 (под соответствием понимается расположение на аналогичных индексных позициях);
 - сформируйте вектор $(|x_1 - \bar{x}|^{\frac{1}{2}}, |x_2 - \bar{x}|^{\frac{1}{2}}, \dots, |x_n - \bar{x}|^{\frac{1}{2}})$, где \bar{x} обозначает среднее значение вектора $x = (x_1, x_2, \dots, x_n)$;
 - определите, сколько элементов вектора y отстоят от максимального значения не более, чем на 200;
 - определите, сколько чётных и нечётных элементов вектора x ;
 - определите, сколько элементов вектора x кратны 7;
 - отсортируйте элементы вектора x в порядке возрастания элементов вектора y ;
 - выведите элементы вектора x , которые входят в десятку наибольших (top-10)?
 - сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора x .
4. Создайте массив `squares`, в котором будут храниться квадраты всех целых чисел от 1 до 100.
5. Подключите пакет `Primes` (функции для вычисления простых чисел). Сгенерируйте массив `myprimes`, в котором будут храниться первые 168 простых чисел. Определите 89-е наименьшее простое число. Получите срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа.
6. Вычислите следующие выражения:
- 6.1) $\sum_{i=10}^{100} (i^3 + 4i^2)$;
- 6.2) $\sum_{i=1}^M \left(\frac{2^i}{i} + \frac{3^i}{i^2} \right)$, $M = 25$;
- 6.3) $1 + \frac{2}{3} + \left(\frac{2}{3} \frac{4}{5} \right) + \left(\frac{2}{3} \frac{4}{5} \frac{6}{7} \right) + \dots + \left(\frac{2}{3} \frac{4}{5} \dots \frac{38}{39} \right)$.

Выполнение лабораторной работы

1. Используя Jupyter Lab, повторил примеры из раздела 2.2.

▼ Кортежи

```
[2]: ()
```

```
[2]: ()
```

```
[4]: favoritelang = ("Python", "Julia", "R")
```

```
[4]: ("Python", "Julia", "R")
```

```
[6]: x1 = (1, 2, 3)
```

```
[6]: (1, 2, 3)
```

```
[8]: x2 = (1, 2.0, "tmp")
```

```
[8]: (1, 2.0, "tmp")
```

```
[10]: x3 = (a=2, b=1+2)
```

```
[10]: (a = 2, b = 3)
```

```
[12]: length(x2)
```

```
[12]: 3
```

```
[21]: c = x1[2] + x1[3]
```

```
[21]: 5
```

```
[23]: x3.a, x3.b, x3[2]
```

```
[23]: (2, 3, 3)
```

```
[25]: in("tmp", x2), 0 in x2
```

```
[25]: (true, false)
```

Словари

```
[27]: phonebook = Dict("Иванов И.И." => ("867-5309", "333-5544"), "Бухгалтерия" => "555-2368")
```

```
[27]: Dict{String, Any} with 2 entries:  
      "Бухгалтерия" => "555-2368"  
      "Иванов И.И." => ("867-5309", "333-5544")
```

```
[29]: keys(phonebook)
```

```
[29]: KeySet for a Dict{String, Any} with 2 entries. Keys:  
      "Бухгалтерия"  
      "Иванов И.И."
```

```
[31]: values(phonebook)
```

```
[31]: ValueIterator for a Dict{String, Any} with 2 entries. Values:  
      "555-2368"  
      ("867-5309", "333-5544")
```

```
[33]: pairs(phonebook)
```

```
[33]: Dict{String, Any} with 2 entries:  
      "Бухгалтерия" => "555-2368"  
      "Иванов И.И." => ("867-5309", "333-5544")
```

```
[35]: haskey(phonebook, "Иванов И.И.")
```

```
[35]: true
```

```
[37]: phonebook["Сидоров П.С."] = "555-3344"
```

```
[37]: "555-3344"
```

```
[39]: pairs(phonebook)
```

```
[39]: Dict{String, Any} with 3 entries:  
      "Сидоров П.С." => "555-3344"  
      "Бухгалтерия"  => "555-2368"  
      "Иванов И.И."  => ("867-5309", "333-5544")
```

```
[43]: pop!(phonebook, "Иванов И.И.")
```

```
[43]: ("867-5309", "333-5544")
```

```
[45]: pairs(phonebook)
```

```
[45]: Dict{String, Any} with 2 entries:  
      "Сидоров П.С." => "555-3344"  
      "Бухгалтерия"  => "555-2368"
```

```
[47]: a = Dict{"foo" => 0.0, "bar" => 42.0};  
      b = Dict{"baz" => 17, "bar" => 13.0};
```

```
[51]: merge(a, b)
```

```
[51]: Dict{String, Real} with 3 entries:  
      "bar" => 13.0  
      "baz" => 17  
      "foo" => 0.0
```

```
[53]: merge(b,a)
```

```
[53]: Dict{String, Real} with 3 entries:  
      "bar" => 42.0  
      "baz" => 17  
      "foo" => 0.0
```

Множества

```
[55]: A = Set([1, 3, 4, 5])
```

```
[55]: Set{Int64} with 4 elements:  
      5  
      4  
      3  
      1
```

```
[68]: B = Set("abracadabra")
```

```
[68]: Set{Char} with 5 elements:  
      'a'  
      'd'  
      'r'  
      'k'  
      'b'
```

```
[76]: S1 = Set([1,2]);  
      S2 = Set([3,4]);  
      issetequal(S1,S2)
```

```
[76]: false
```

```
[78]: S3 = Set([1,2,2,3,1,2,3,2,1]);  
      S4 = Set([2,3,1]);  
      issetequal(S3,S4)
```

```
[78]: true
```

```
[80]: C=union(S1,S2)
```

```
[80]: Set{Int64} with 4 elements:  
      4  
      2  
      3  
      1
```

```
[82]: D = intersect(S1,S3)
```

```
[82]: Set{Int64} with 2 elements:  
      2  
      1
```

```
[84]: E = setdiff(S3,S1)
```

```
[84]: Set{Int64} with 1 element:  
      3
```



```
[86]: issubset(S1,S4)
```

```
[86]: true
```

```
[88]: push!(S4, 99)
```

```
[88]: Set{Int64} with 4 elements:
```

```
2
```

```
99
```

```
3
```

```
1
```

```
[90]: pop!(S4)
```

```
[90]: 2
```

```
[92]: S4
```

```
[92]: Set{Int64} with 3 elements:
```

```
99
```

```
3
```

```
1
```

▼ Массивы ¶

```
[95]: # создание пустого массива с абстрактным типом:  
empty_array_1 = []
```

```
[95]: Any[]
```

```
[97]: # создание пустого массива с конкретным типом:  
empty_array_2 = (Int64[])  
empty_array_3 = (Float64[])
```

```
[97]: Float64[]
```

```
[99]: # вектор-столбец:  
a = [1, 2, 3]
```

```
[99]: 3-element Vector{Int64}:  
 1  
 2  
 3
```

```
[101]: # вектор-строка:
```

```
b = [1 2 3]
```

```
[101]: 1x3 Matrix{Int64}:
```

```
1 2 3
```

```
[103]: # многомерные массивы (матрицы):
```

```
A = [[1, 2, 3] [4, 5, 6] [7, 8, 9]]
```

```
B = [[1 2 3]; [4 5 6]; [7 8 9]]
```

```
[103]: 3x3 Matrix{Int64}:
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
[105]: # одномерный массив из 8 элементов (массив  $1 \times 8$ )
```

```
# со значениями, случайно распределёнными на интервале [0, 1):
```

```
c = rand(1,8)
```

```
[105]: 1x8 Matrix{Float64}:
```

```
0.500108 0.692296 0.19341 0.48241 0.629914 0.168494 0.464049 0.816967
```

```
[111]: # многомерный массив $2 \times 3$ (2 строки, 3 столбца) элементов  
# со значениями, случайно распределёнными на интервале [0, 1):  
C = rand(2,3)
```

```
[111]: 2x3 Matrix{Float64}:  
 0.781188  0.614363  0.118252  
 0.748905  0.969616  0.932795
```

```
[117]: # трёхмерный массив:  
D = rand(4, 3, 2)
```

```
[117]: 4x3x2 Array{Float64, 3}:  
[:, :, 1] =  
 0.529021  0.154325  0.8865  
 0.892431  0.980441  0.706131  
 0.559755  0.0613123 0.243594  
 0.86616  0.863667  0.678796  
  
[:, :, 2] =  
 0.858315  0.637923  0.855021  
 0.0288136 0.763771  0.745951  
 0.150178  0.0504343 0.680842  
 0.0546455 0.795185  0.586381
```

```
[119]: # массив из квадратных корней всех целых чисел от 1 до 10:  
roots = [sqrt(i) for i in 1:10]
```

```
[119]: 10-element Vector{Float64}:  
 1.0  
 1.4142135623730951  
 1.7320508075688772  
 2.0  
 2.23606797749979  
 2.449489742783178  
 2.6457513110645907  
 2.8284271247461903  
 3.0  
 3.1622776601683795
```

```
[121]: # массив с элементами вида 3*x^2,  
# где x - нечётное число от 1 до 9 (включительно)  
ar_1 = [3*i^2 for i in 1:2:9]
```

```
[121]: 5-element Vector{Int64}:  
 3  
 27  
 75  
 147  
 243
```

```
[123]: # массив квадратов элементов, если квадрат не делится на 5 или 4:  
ar_2=[i^2 for i=1:10 if (i^2%5!=0 && i^2%4!=0)]
```

```
[123]: 4-element Vector{Int64}:  
      1  
      9  
     49  
     81
```

```
[125]: # одномерный массив из пяти единиц:  
ones(5)
```

```
[125]: 5-element Vector{Float64}:  
      1.0  
      1.0  
      1.0  
      1.0  
      1.0
```

```
[127]: # двумерный массив 2x3 из единиц:  
ones(2,3)
```

```
[127]: 2x3 Matrix{Float64}:  
      1.0  1.0  1.0  
      1.0  1.0  1.0
```

```
[129]: # одномерный массив из 4 нулей:  
zeros(4)
```

```
[129]: 4-element Vector{Float64}:  
 0.0  
 0.0  
 0.0  
 0.0
```

```
[131]: # заполнить массив 3x2 цифрами 3.5  
fill(3.5,(3,2))
```

```
[131]: 3x2 Matrix{Float64}:  
 3.5  3.5  
 3.5  3.5  
 3.5  3.5
```

```
[162]: # заполнение массива посредством функции repeat():  
repeat([1,2],3,3)  
repeat([1 2],3,3)
```

```
[162]: 3x6 Matrix{Int64}:  
 1  2  1  2  1  2  
 1  2  1  2  1  2  
 1  2  1  2  1  2
```

```
[180]: # преобразование одномерного массива из целых чисел от 1 до 12
# в двумерный массив 2x6
a = collect(1:12)
b = reshape(a,(2,6))
```

```
[180]: 2x6 Matrix{Int64}:
 1  3  5  7  9 11
 2  4  6  8 10 12
```

```
[182]: # транспонирование
b'
```

```
[182]: 6x2 adjoint(::Matrix{Int64}) with eltype Int64:
 1  2
 3  4
 5  6
 7  8
 9 10
11 12
```

```
[184]: # транспонирование
c = transpose(b)
```

```
[184]: 6x2 transpose(::Matrix{Int64}) with eltype Int64:
 1  2
 3  4
 5  6
 7  8
 9 10
11 12
```



```
[186]: # массив 10x5 целых чисел в диапазоне [10, 20]:  
ar = rand(10:20, 10, 5)
```

```
[186]: 10x5 Matrix{Int64}:  
 17 14 17 15 14  
 18 14 19 18 13  
 20 18 17 17 11  
 14 17 15 13 19  
 15 13 17 11 20  
 13 16 16 16 18  
 20 14 10 13 20  
 20 20 18 18 13  
 14 10 18 13 15  
 14 11 13 15 12
```

```
[188]: # выбор всех значений строки в столбце 2:  
ar[:, 2]
```

```
[188]: 10-element Vector{Int64}:  
 14  
 14  
 18  
 17  
 13  
 16  
 14  
 20  
 10  
 11
```

```
[190]: # выбор всех значений в столбцах 2 и 5:
```

```
ar[:, [2, 5]]
```

```
[190]: 10x2 Matrix{Int64}:
```

```
14 14
14 13
18 11
17 19
13 20
16 18
14 20
20 13
10 15
11 12
```

```
[192]: # все значения строк в столбцах 2, 3 и 4:
```

```
ar[:, 2:4]
```

```
[192]: 10x3 Matrix{Int64}:
```

```
14 17 15
14 19 18
18 17 17
17 15 13
13 17 11
16 16 16
14 10 13
20 18 18
10 18 13
11 13 15
```

```
[194]: # значения в строках 2, 4, 6 и в столбцах 1 и 5:  
ar[[2, 4, 6], [1, 5]]
```

```
[194]: 3x2 Matrix{Int64}:  
 18 13  
 14 19  
 13 18
```

```
[196]: # значения в строке 1 от столбца 3 до последнего столбца:  
ar[1, 3:end]
```

```
[196]: 3-element Vector{Int64}:  
 17  
 15  
 14
```

```
[198]: # сортировка по столбцам:  
sort(ar,dims=1)
```

```
[198]: 10x5 Matrix{Int64}:  
 13 10 10 11 11  
 14 11 13 13 12  
 14 13 15 13 13  
 14 14 16 13 13  
 15 14 17 15 14  
 17 14 17 15 15  
 18 16 17 16 18  
 20 17 18 17 19  
 20 18 18 18 20  
 20 20 19 18 20
```

```
[200]: sort(ar,dims=2)
```

```
[200]: 10x5 Matrix{Int64}:
```

```
 14  14  15  17  17
 13  14  18  18  19
 11  17  17  18  20
 13  14  15  17  19
 11  13  15  17  20
 13  16  16  16  18
 10  13  14  20  20
 13  18  18  20  20
 10  13  14  15  18
 11  12  13  14  15
```

```
[215]: # поэлементное сравнение с числом
# (результат - массив логических значений):
ar .> 14
```

```
[215]: 10x5 BitMatrix:
```

```
 1  0  1  1  0
 1  0  1  1  0
 1  1  1  1  0
 0  1  1  0  1
 1  0  1  0  1
 0  1  1  1  1
 1  0  0  0  1
 1  1  1  1  0
 0  0  1  0  1
 0  0  0  1  0
```

```
[220]: # возврат индексов элементов массива, удовлетворяющих условию:  
findall(ar .> 14)
```

```
[220]: 29-element Vector{CartesianIndex{2}}:
```

```
 CartesianIndex(1, 1)  
 CartesianIndex(2, 1)  
 CartesianIndex(3, 1)  
 CartesianIndex(5, 1)  
 CartesianIndex(7, 1)  
 CartesianIndex(8, 1)  
 CartesianIndex(3, 2)  
 CartesianIndex(4, 2)  
 CartesianIndex(6, 2)  
 CartesianIndex(8, 2)  
 CartesianIndex(1, 3)  
 CartesianIndex(2, 3)  
 CartesianIndex(3, 3)  
      :  
 CartesianIndex(9, 3)  
 CartesianIndex(1, 4)  
 CartesianIndex(2, 4)  
 CartesianIndex(3, 4)  
 CartesianIndex(6, 4)  
 CartesianIndex(8, 4)  
 CartesianIndex(10, 4)  
 CartesianIndex(4, 5)  
 CartesianIndex(5, 5)  
 CartesianIndex(6, 5)  
 CartesianIndex(7, 5)  
 CartesianIndex(9, 5)
```

Задания для самостоятельной работы

▼ Выполните задания для самостоятельной работы (раздел 2.4). ¶

1. Даны множества: $A = \{0, 3, 4, 9\}$, $B = \{1, 3, 4, 7\}$, $C = \{0, 1, 2, 4, 7, 8, 9\}$. Найти

$$P = A \cap B \cup A \cap B \cup A \cap C \cup B \cap C.$$

```
[237]: A = Set([0,3,4,9])  
      B = Set([1,3,4,7])  
      C = Set([0,1,2,4,7,8,9])
```

```
[237]: Set{Int64} with 7 elements:  
      0  
      4  
      7  
      2  
      9  
      8  
      1
```

```
[285]: D = intersect(A,B)
```

```
[285]: Set{Int64} with 2 elements:  
      4  
      3
```

```
[291]: E = intersect(A,C)
```

```
[291]: Set{Int64} with 2 elements:  
      4  
      3
```

```
[293]: F = intersect(B,C)
```

```
[293]: Set{Int64} with 2 elements:  
      4  
      3
```

```
[297]: P = union(D,E)
```

```
[297]: Set{Int64} with 2 elements:  
      4  
      3
```


2. Приведите свои примеры с выполнением операций над множествами элементов разных типов.

```
[304]: A = Set("abrakadabra")
```

```
[304]: Set(Char) with 5 elements:
```

```
'a'  
'd'  
'p'  
'k'  
'b'
```

```
[306]: B = Set("ghedfuawfjdaa")
```

```
[306]: Set(Char) with 9 elements:
```

```
'f'  
'u'  
'd'  
'a'  
'o'  
'g'  
'h'  
'u'  
'j'
```

```
[308]: C = intersect(A,B)
```

```
[308]: Set{Char} with 2 elements:  
      'a'  
      'd'
```

```
[310]: P = union(A,B)
```

```
[310]: Set{Char} with 12 elements:  
      'f'  
      'w'  
      'a'  
      'd'  
      'e'  
      'g'  
      'h'  
      'u'  
      'r'  
      'k'  
      'j'  
      'b'
```

```
[316]: E = setdiff(A,B)
```

```
[316]: Set{Char} with 3 elements:  
      'r'  
      'k'  
      'b'
```

```
[318]: E = setdiff(B,A)
```

```
[318]: Set{Char} with 7 elements:  
      'f'  
      'w'  
      'e'  
      'g'  
      'h'  
      'u'  
      'j'
```

3. Создайте разными способами:

▼ 3.1 массив (1, 2, 3, ... $N - 1$, N), N выберите больше 20

```
[357]: N=50  
A = zeros(Int64,N)  
for i in 1:N  
    A[i] = i  
end  
A
```

```
[357]: 50-element Vector{Int64}:
```

```
 1  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
10  
11  
12  
13  
⋮  
39  
40  
41
```

```
[360]: a = collect(1:N)
```

```
[360]: 50-element Vector{Int64}:
```

```
 1
```

```
 2
```

```
 3
```

```
 4
```

```
 5
```

```
 6
```

```
 7
```

```
 8
```

```
 9
```

```
10
```

```
11
```

```
12
```

```
13
```

```
⋮
```

```
39
```

```
40
```

```
41
```

```
42
```

```
43
```

```
44
```

```
45
```

```
46
```

```
47
```

```
48
```

```
49
```

```
50
```

3.2 массив ($N, N - 1 \dots, 2, 1$), N выберите больше 20;

```
366]: N=50  
A = zeros(Int64,N)  
for i in 1:N  
    A[i] = N  
    N-=1  
end  
A
```

```
366]: 50-element Vector{Int64}:
```

```
50  
49  
48  
47  
46  
45  
44  
43  
42  
41  
40  
39  
38  
:  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2
```

```
[376]: N=50  
a = collect(N:-1:1)
```

```
[376]: 50-element Vector{Int64}:
```

```
 50
```

```
 49
```

```
 48
```

```
 47
```

```
 46
```

```
 45
```

```
 44
```

```
 43
```

```
 42
```

```
 41
```

```
 40
```

```
 39
```

```
 38
```

```
  ⋮
```

```
 12
```

```
 11
```

```
 10
```

```
  9
```

```
  8
```

```
  7
```

```
  6
```

```
  5
```

```
  4
```

```
  3
```

```
  2
```

```
  1
```

3.3 массив (1, 2, 3, ..., $N - 1$, N , $N - 1$, ..., 2, 1), N выберите больше 20;

```
[457]: N=50
k=1
A = zeros(Int64,N*2+1)
for i in 1:N
    A[i] = i
    k+=1
end
for i in N+1:N*2+1
    A[i] = k
    k-=1
end
A
```

```
[457]: 101-element Vector{Int64}:
```

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
 ⋮
12
11
10
```



```
[488]: a = collect(1:N)
      b = collect(N:-1:1)
      c=vcat(collect(1:N),collect(N:-1:1))
```

```
[488]: 100-element Vector{Int64}:
```

```
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
 ⋮
12
11
10
 9
 8
 7
 6
 5
 4
 3
 2
 1
```

3.4 массив с именем tmp вида (4, 6, 3)

```
[495]: tmp = [4,6,3]
```

```
[495]: 3-element Vector{Int64}:  
      4  
      6  
      3
```

```
[497]: tmp1=4  
      tmp2=6  
      tmp3=3  
      tmp = [tmp1,tmp2,tmp3]
```

```
[497]: 3-element Vector{Int64}:  
      4  
      6  
      3
```

3.5 массив, в котором первый элемент массива tmp повторяется 10 раз;

```
[504]: TMP = zeros(Int64,10)
      for i in 1:10
          TMP[i] = tmp1
      end
      TMP
```

```
[504]: 10-element Vector{Int64}:
         4
         4
         4
         4
         4
         4
         4
         4
         4
         4
```

```
[506]: fill(tmp1,(1,10))
```

```
[506]: 1×10 Matrix{Int64}:
      4  4  4  4  4  4  4  4  4  4
```

3.6 массив, в котором все элементы массива tmp повторяются 10 раз

```
[510]: TMP = zeros(Int64,30)
      for i in 0:9
          TMP[3*i+1] = tmp1
          TMP[3*i+2] = tmp2
          TMP[3*i+3] = tmp3
      end
      TMP
```

```
[510]: 30-element Vector{Int64}:
         4
         6
         3
         4
         6
         3
         4
         6
         3
         4
         6
         3
         4
         6
         3
         4
         6
         3
         4
         6
         3
         4
         6
         3
         4
         6
         3
         4
         6
         3
```

```
[516]: repeat([tmp1,tmp2,tmp3],10,1)
```

```
[516]: 30x1 Matrix{Int64}:
```

```
4
```

```
6
```

```
3
```

```
4
```

```
6
```

```
3
```

```
4
```

```
6
```

```
3
```

```
4
```

```
6
```

```
3
```

```
4
```

```
:
```

```
4
```

```
6
```

```
3
```

```
4
```

```
6
```

```
3
```

```
4
```

```
6
```

```
3
```

```
4
```

```
6
```

```
3
```

3.7 массив, в котором первый элемент массива tmp встречается 11 раз, второй элемент — 10 раз, третий элемент — 10 раз

```
[529]: N=11
tmp = zeros(Int64,N)
for i in 0:N
    tmp[i+1:] = tmp1
    tmp[i+2:] = tmp2
    tmp[i+3:] = tmp3
end
tmp[N] = tmp1
tmp
```

```
[529]: 31-element Vector{Int64}:
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
 6
 3
 4
```

3.8 массив, в котором первый элемент массива tmp встречается 10 раз подряд, второй элемент — 20 раз подряд, третий элемент — 30 раз подряд;

```
[556]: N=60
      k=1
      A = zeros(Int64,N)
      for i in 1:10
          A[i] = tmp1
      end
      for i in 11:30
          A[i] = tmp2
      end
      for i in 31:60
          A[i] = tmp3
      end
      A
```

```
[556]: 60-element Vector{Int64}:
```

4
4
4
4
4
4
4
4
4
4
6
6
6
:
3
3
3
3
3
3
3
3
3
3
3

3.9 массив из элементов вида

$$2^{\text{tmp}[i]}$$

, $i = 1, 2, 3$, где элемент

$$2^{\text{tmp}[3]}$$

встречается 4 раза; посчитайте в полученном векторе, сколько раз встречается цифра 6, и выведите это значение на экран

```
[594]: tmp = zeros(Int64, 4)
for i in 1:3
    tmp[i] = 2^tmp[i]
    tmp[i+1] = 2^tmp[i+1]
    tmp[i+2] = 2^tmp[i+2]
end
for i in 1:3
    tmp[i+1] = 2^tmp[i]
end
tmp
```

```
[594]: 6-element Vector{Int64}:
 16
 64
  8
 16
 16
 16
```



```
[618]: TMP = split(join(string.(TMP)), "")
```

```
[618]: 11-element Vector{SubString{String}}:  
  "1"  
  "6"  
  "6"  
  "4"  
  "8"  
  "1"  
  "6"  
  "1"  
  "6"  
  "1"  
  "6"
```

```
[614]: size(findall(TMP .=="6"))[1]
```

```
[614]: 5
```

3.10 вектор значений $y = e^x * \cos(x)$ в точках $x = 3, 3.1, 3.2, \dots, 6$, найдите среднее значение y

```
[638]: a = collect(3:0.1:6)
A = ones(size(a)[1])
for i in 1:size(a)[1]
    A[i] = exp(a[i])*cos(a[i])
end
A
```

```
[638]: 31-element Vector{Float64}:
-19.884530844146987
-22.178753389342127
-24.490696732801293
-26.77318244299338
-28.969237768093574
-31.011186439374516
-32.819774760338504
-34.30336011037369
-35.35719361853035
-35.86283371230767
-35.68773240011913
-34.68504225166807
-32.693695428321746
⋮
25.046704998273004
42.09920106253839
61.99663027669454
84.92906736250268
111.0615860420258
140.5250750527875
173.40577640857734
209.73349424783467
249.46844055885668
292.4867067371223
338.5643778585117
387.36034029093076
```

```
[645]: sum(A)/size(a)[1]
```

```
[645]: 53.11374594642971
```

```
[645]: sum(A)/size(a)[1]
```

```
[645]: 53.11374594642971
```

3.11 Вектор вида (x^i, y^j) , $x = 0.1$, $i = 3, 6, 9, \dots, 36$, $y = 0.2$, $j = 1, 4, 7, \dots, 34$

```
[698]: x=0.1
       y=0.2
       i1 = collect(3:3:36)
       i2 = collect(1:3:34)
       A = zeros(size(i1)[1],2)
       for i in 1:size(i1)[1]
           A[i,1] = x^i1[i]
           A[i,2] = y^i2[i]
       end
       A
```

```
[698]: 12x2 Matrix{Float64}:
 0.001  0.2
 1.0e-6  0.0016
 1.0e-9  1.28e-5
 1.0e-12 1.024e-7
 1.0e-15 8.192e-10
 1.0e-18 6.5536e-12
 1.0e-21 5.24288e-14
 1.0e-24 4.1943e-16
 1.0e-27 3.35544e-18
 1.0e-30 2.68435e-20
 1.0e-33 2.14748e-22
 1.0e-36 1.71799e-24
```

3.12 Вектор с элементами $2^i/i$, $i = 1, 2, \dots, M$, $M=25$

```
[708]: a = collect(1:25)
A = zeros(size(a)[1])
for i in 1:size(a)[1]
    A[i] = 2^i/i
end
A
```

```
[708]: 25-element Vector{Float64}:
```

```
 2.0
 2.0
 2.6666666666666665
 4.0
 6.4
10.666666666666666
18.285714285714285
32.0
56.888888888888886
102.4
186.1818181818182
341.3333333333333
630.1538461538462
1170.2857142857142
2184.5333333333333
4096.0
7710.117647058823
14563.555555555555
27594.105263157893
52428.8
99864.38095238095
190650.18181818182
364722.0869565217
699050.6666666666
 1.34217728e6
```

3.13 вектор вида ("fn1", "fn2", ..., "fnN"), $N = 30$

```
[713]: N=30  
A = [join(["fn", string(i)]) for i in 1:N]
```

```
[713]: 30-element Vector{String}:
```

```
"fn1"  
"fn2"  
"fn3"  
"fn4"  
"fn5"  
"fn6"  
"fn7"  
"fn8"  
"fn9"  
"fn10"  
"fn11"  
"fn12"  
"fn13"  
⋮  
"fn19"  
"fn20"  
"fn21"  
"fn22"  
"fn23"  
"fn24"  
"fn25"  
"fn26"  
"fn27"  
"fn28"  
"fn29"  
"fn30"
```

3.14 Векторы $x=(x_1, x_2, \dots, x_n)$ и $y=(y_1, y_2, \dots, y_n)$ целочисленного типа длины $n=250$ как случайные выборки из совокупности $0, 1, \dots, 999$ на его основе:

```
[756]: n=250  
x = rand([0:999, n])  
y = rand([0:999, n])
```

```
[756]: 250-element Vector{Int64}:
```

```
400  
325  
906  
589  
848  
361  
20  
205  
931  
364  
32  
947  
210  
⋮  
205  
226  
152  
400  
717  
876  
78  
662  
981  
778  
921  
815
```


Сформируйте вектор $(y_2 - x_1, \dots, y_n - x_n - 1)$

```
[761]: A = zeros(Int64,n-1)
      for i in 1:n-1
          A[i] = y[i+1]-x[i]
      end
      A
```

```
[761]: 249-element Vector{Int64}:
```

```
-14
657
-247
763
-480
-534
-254
576
213
-964
520
-510
127
⋮
-605
-64
96
-356
697
476
-224
383
887
304
208
184
```

Сформируйте вектор($x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \dots, x_{n-2} + 2x_{n-1} - x_n$)

```
[790]: A = zeros(Int64,n-2)
      for i in 1:n-2
          A[i] = x[i]+2*x[i+1]-x[i+2]
      end
      A
```

[790]: 248-element Vector{Int64}:

```
 1
1836
165
1213
1490
1117
1018
-339
1716
1130
1376
1596
659
⋮
1334
1334
-354
1548
396
518
725
766
-7
329
1269
1484
```

Сформируйте вектор

$$\left(\frac{\sin(y_1)}{\cos(x_2)}, \frac{\sin(y_2)}{\cos(x_3)}, \dots, \frac{\sin(y_{n-1})}{\cos(x_n)} \right)$$



```
815]: A = zeros(n-1)
      for i in 1:n-1
          A[i] = sin(y[i])/cos(x[i+1])
      end
      A
```

```
815]: 249-element Vector{Float64}:
 0.57671939880262639
-1.0466882417184395
-0.954462610452438
-1.7095505581993755
-0.4834477929301719
 0.29506690691539267
-0.9129452511424161
-0.7299679915863524
-0.891870948153518
-0.42604033021542276
-0.6572124289813911
-1.6031512363442237
 0.6813221871933406
 ⋮
 0.3901909335560866
-0.8378816923055991
 0.4475459420760665
 2.287090296499363
 1.6198844291169272
 0.7155100792776311
-0.5862069788245171
 0.5301701893427556
-0.8273888341142545
-0.7406576474653931
 1.0021243617254574
-0.8017645708456649
```

▼ Вычислите

$$\sum_{i=1}^{n-1} \frac{e^{-x_{i+1}}}{x_i + 10}$$



```
39]: summ=0
      for i in 1:n-1
          summ += exp(-x[i+1])/(x[i]+10)
      end
      summ
```

```
39]: 0.009316277642042724
```

выберите элементы вектора y, значения которых больше 600, и выведите на экран; определите индексы этих элементов;

```
[868]: A = []  
for i in 1:n  
    if y[i] > 600  
        println("y[" , i , "] = " , y[i])  
    end  
end
```

```
y[1] = 906  
y[2] = 848  
y[9] = 931  
y[12] = 947  
y[14] = 618  
y[15] = 673  
y[17] = 722  
y[21] = 612  
y[22] = 623  
y[25] = 885  
y[28] = 679  
y[36] = 814  
y[37] = 725  
y[38] = 836  
y[39] = 692  
y[44] = 783  
y[45] = 728  
y[47] = 680  
y[49] = 627  
y[51] = 653
```

определите значения вектора x, соответствующие значениям вектора y, значения которых больше 600 (под соответствием понимается расположение на аналогичных индексных позициях);

```
770] A = []  
    for i in 1:n  
        if x[i] > 600 && y[i] > 600  
            println("x[%d,%d] = ", x[i])  
        end  
    end
```

```
x[3] = 836  
x[5] = 841  
x[12] = 720  
x[25] = 680  
x[37] = 883  
x[39] = 877  
x[54] = 785  
x[59] = 751  
x[60] = 874  
x[62] = 764  
x[67] = 891  
x[69] = 887  
x[81] = 888  
x[83] = 645  
x[85] = 699  
x[88] = 713  
x[93] = 648  
x[94] = 902  
x[103] = 619  
x[110] = 982  
x[118] = 714  
x[124] = 953  
x[129] = 655
```

сформируйте вектор

$$(|x_1 - x^*|^{\frac{1}{2}}, |x_2 - x^*|^{\frac{1}{2}}, \dots, |x_n - x^*|^{\frac{1}{2}})$$

, где x^* обозначает среднее значение вектора

$$x = (x_1, x_2, \dots, x_n)$$

↑

```
[911]: xsum = sum(x)/n
      A = zeros(n)
      for i in 1:n
          A[i] = (abs(x[i] - xsum))^0.5
      end
      A
```

```
[911]: 250-element Vector{Float64}:
      11.786602563928248
      15.130234631359817
      18.922896184252558
      19.82231066248332
      19.054553261622274
      8.722155696844675
      4.350172410376396
      11.087109632361356
      18.081039793109245
      22.76128291639116
      7.136105380387819
      15.558791726866197
      3.6160752204565685
      ⋮
      13.70853748581518
```

определите, сколько элементов вектора y отсто.т от максимального значения не более, чем на 200; 1

```
[1284]: j=0  
for i in 1:n  
    if y[i] >= maximum(y) - 200  
        j=j+1  
    end  
end  
j
```

```
[1284]: 46
```


определите, сколько чётных и нечётных элементов вектора x ;

```
[966]: j=0
      k=0
      for i in 1:n
          if x[i] % 2 == 0
              j+=1
          end
          if x[i] % 2 != 0
              k+=1
          end
      end
      println("Вектор x состоит из ",j," четных элементов")
      println("Вектор x состоит из ",k," нечетных элементов")
```

Вектор x состоит из 127 четных элементов

Вектор x состоит из 123 нечетных элементов

определите, сколько элементов вектора x кратны 7

```
[973]: j=0
      for i in 1:n
          if x[i] % 7 == 0
              j+=1
          end
      end
      println("Вектор x состоит из ",j," элементов кратных 7")
```

Вектор x состоит из 30 элементов кратных 7

отсортируйте элементы вектора x в порядке возрастания элементов вектора y

```
[1104]: x = rand(0:999, n)
```

```
[1104]: 250-element Vector{Int64}:
```

```
 21  
377  
219  
376  
496  
732  
342  
729  
193  
43  
475  
843  
788  
:  
902  
271  
651  
224  
820  
105  
322  
296  
901  
985  
416  
537
```

```
[1095]: y = rand(0:999, n)
```

```
[1095]: 250-element Vector{Int64}:
```

```
557
```

```
872
```

```
526
```

```
921
```

```
344
```

```
676
```

```
153
```

```
538
```

```
777
```

```
73
```

```
484
```

```
2
```

```
145
```

```
⋮
```

```
311
```

```
973
```

```
97
```

```
712
```

```
94
```

```
208
```

```
480
```

```
934
```

```
877
```

```
846
```

```
317
```

```
417
```

```

97]: j=0
      k=0
      n=250
      a = zeros(n)
      A = zeros(n)
      B = zeros(n)

      for i in 1:n-1
          for j in 1:n-1
              if (y[j] > y[j+1])
                  a[j] = y[j]
                  y[j]=y[j+1]
                  y[j+1]=a[j]
                  A[j] = x[j]
                  x[j]=x[j+1]
                  x[j+1]=A[j]
              end
          end
      end
      y

```

```

97]: 250-element Vector{Int64}:

```

```

 2
 4
12
15
21
24
28
32
32
35
37
38
41
 ⋮
952

```

```
[1099]:
```

```
x
```

```
[1099]: 250-element Vector{Int64}:
```

```
103
```

```
413
```

```
103
```

```
557
```

```
125
```

```
852
```

```
238
```

```
521
```

```
484
```

```
85
```

```
977
```

```
904
```

```
675
```

```
⋮
```

```
834
```

```
18
```

```
210
```

```
164
```

```
345
```

```
267
```

```
829
```

```
776
```

```
320
```

```
29
```

```
49
```

```
911
```

выведите элементы вектора x , которые входят в десятку наибольших (top-10)?

```
[1129]: j=0
k=1
n=250
a = zeros(n)
A = zeros(n)
B = zeros(n)

for i in 1:n-1
    for j in 1:n-1
        if (x[j] > x[j+1])
            A[j] = x[j]
            x[j]=x[j+1]
            x[j+1]=A[j]

        end
    end
end
for i in n-9:n
    println("Наибольшее значение №", k, " - x[",i,"]", " = ", x[i])
    k+=1
end
```

```
Наибольшее значение №1 - x[241] = 954
Наибольшее значение №2 - x[242] = 962
Наибольшее значение №3 - x[243] = 964
Наибольшее значение №4 - x[244] = 970
Наибольшее значение №5 - x[245] = 981
Наибольшее значение №6 - x[246] = 985
Наибольшее значение №7 - x[247] = 987
Наибольшее значение №8 - x[248] = 988
Наибольшее значение №9 - x[249] = 993
Наибольшее значение №10 - x[250] = 995
```

сформируйте вектор, содержащий только уникальные (неповторяющиеся) элементы вектора x

```
1133]: unique(x)
```

```
1133]: 222-element Vector{Int64}:
```

```
 4  
 6  
16  
21  
25  
29  
36  
40  
41  
43  
50  
55  
62  
:  
936  
952  
954  
962  
964  
970  
981  
985  
987  
988  
993  
995
```


4. Создайте массив squares, в котором будут храниться квадраты всех целых чисел от 1 до 100.

```
[1232]: squares = zeros(100)
for i in 1:100
    squares[i] = i^2
end
squares
```

```
[1232]: 100-element Vector{Float64}:
 1.0
 4.0
 9.0
16.0
25.0
36.0
49.0
64.0
81.0
100.0
121.0
144.0
169.0
 ⋮
7921.0
8100.0
8281.0
8464.0
8649.0
8836.0
9025.0
9216.0
9409.0
9604.0
9801.0
10000.0
```

5. Подключите пакет Primes (функции для вычисления простых чисел). Сгенерируйте массив `myprimes`, в котором будут храниться первые 168 простых чисел. Определите 89-е наименьшее простое число. Получите срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа.

```
1244]: import Pkg; Pkg.add("Primes")
```

```
Resolving package versions...
No changes to 'C:\Users\ksudr\.julia\environments\v1.10\Project.toml'
No changes to 'C:\Users\ksudr\.julia\environments\v1.10\Manifest.toml'
```

```
1245]: using Primes
```

```
1246]: myprimes = [prime(i) for i in 1:168]
```

```
1248]: 168-element Vector{Int64}:
```

```
 2
 3
 5
 7
11
13
17
19
23
29
31
37
41
 |
919
929
937
941
947
953
967
971
977
983
991
997
```

```
[1252]: myprimes[89]
```

```
[1252]: 461
```

```
[1254]: myprimes[89:99]
```

```
[1254]: 11-element Vector{Int64}:
```

```
 461
```

```
 463
```

```
 467
```

```
 479
```

```
 487
```

```
 491
```

```
 499
```

```
 503
```

```
 509
```

```
 521
```

```
 523
```

Вычислите следующие выражения:

$$\sum_{i=10}^{100} (i^3 + 4i^2)$$

¶

```
[1264]: a = sum([i^3+4*i^2 for i in 10:100])
```

```
[1264]: 26852735
```

$$\sum_{i=1}^M \left(\frac{2^i}{i} + \frac{3^i}{i^2} \right)$$

M=25

```
[1268]: M=25  
a = sum([2^i/i+3^i/i^2 for i in 1:M])
```

```
[1268]: 2.1291704368143802e9
```

Выводы

Используя Jupyter lab повторил примеры из раздела 2.2, выполнил задания для самостоятельной работы