Лабораторная работа № 2. Julia. Структуры данных

Компьютерный практикум по статистическому анализу данных

Демидова Е. А.

16 ноября 2024

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Демидова Екатерина Алексеевна
- студентка группы НКНбд-01-21
- Российский университет дружбы народов
- https://github.com/eademidova

Введение

Цель работы

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

Задачи

1. Используя Jupyter Lab, повторите примеры из раздела 2.2.
2. Выполните задания для самостоятельной работы (раздел 2.4).

Выполнение лабораторной работы

Примеры

```
a = [1 2; 3 4]
b = [1 1]
```

```
1×2 Matrix{Int64}:
 1  1
```

```
a*b'*2
```

```
2×1 Matrix{Int64}:
  6
 14
```

```
x = (a = 2, b = "3")
println(x.a, x[1])
println(in("3", x))
```

```
22
true
```

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

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

```
pop!(phonebook, "Иванов И.И.")
```

```
("867-5309", "333-5544")
```

```
phonebook
```

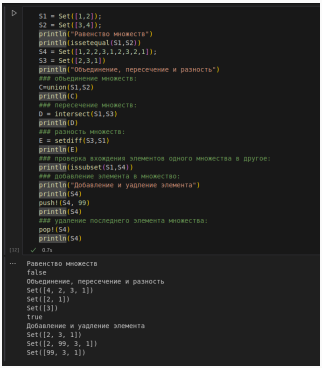
```
Dict{String, Any} with 1 entry:
  "Бухгалтерия" => "555-2368"
```

```
a = Dict("foo" => 0.0, "bar" => 42.0);
b = Dict("baz" => 17, "bar" => 13.0);
print(merge(a, b), merge(b,a))
```

```
Dict{String, Real}("bar" => 13.0, "baz" => 17, "foo" => 0.0)Dict{String, Real}("bar" => 42.0, "baz" => 17, "foo" => 0.0)
```

Рис. 1: Примеры. Кортежи

Примеры



```
S1 = Set([1,2]);
S2 = Set([3,4]);
println("Равенство множеств")
println(issetequal(S1,S2))
S4 = Set([1,2,2,3,1,2,3,2,1]);
S3 = Set([2,3,1])
println("Объединение, пересечение и разность")
### объединение множеств:
C=union(S1,S2)
println(C)
### пересечение множеств:
D = intersect(S1,S3)
println(D)
### разность множеств:
E = setdiff(S3,S1)
println(E)
### проверка вхождения элементов одного множества в другое:
println(issubset(S1,S4))
### добавление элемента в множество:
println("Добавление и удаление элемента")
println(S4)
push!(S4, 99)
println(S4)
### удаление последнего элемента множества:
pop!(S4)
println(S4)
```

```
[37]  ✓ 0.7s

Равенство множеств
false
Объединение, пересечение и разность
Set([4, 2, 3, 1])
Set([2, 1])
Set([3])
true
Добавление и удаление элемента
Set([2, 3, 1])
Set([2, 99, 3, 1])
Set([99, 3, 1])
```

Рис. 2: Примеры. Множества

Примеры

```
A = rand(4,3,2)
```

```
4×3×2 Array{Float64, 3}:
[:, :, 1] =
 0.53715   0.0253307  0.557304
 0.392261  0.613429   0.321627
 0.230467  0.612415   0.780613
 0.975484  0.393498   0.595916

[:, :, 2] =
 0.122025   0.264899  0.171144
 0.985735   0.846615  0.820727
 0.00105212 0.443124  0.609205
 0.560781   0.983995  0.443232
```

```
A[1, 2, :]
```

```
2-element Vector{Float64}:
 0.025330716121884866
 0.26408877227643766
```

```
ar_2=[i^2 for i=1:10 if (i^2%5!=0 && i^2%4!=0)]
```

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

```
ar = rand(10:20, 10, 5)
```

```
10×5 Matrix{Int64}:
 17  17  10  20  14
 12  13  10  18  20
 18  10  18  17  13
 11  11  15  13  10
 16  10  19  13  15
```

Рис. 3: Примеры. Массивы

Примеры

```
ar[[2, 4, 6], [1, 5]]
```

```
3×2 Matrix{Int64}:
 12  20
 11  10
 18  14
```

```
ar[1, 3:end]
```

```
3-element Vector{Int64}:
 10
 20
 14
```

```
sort(ar,dims=1)
```

```
10×5 Matrix{Int64}:
 11  10  10  13  10
 13  10  10  13  10
 11  11  10  13  13
 12  11  13  14  14
 12  13  15  17  14
 16  14  18  17  15
 17  15  19  18  18
 18  17  19  19  18
 18  17  20  20  19
 20  20  20  20  20
```

```
sort(ar,dims=2)
```

```
10×5 Matrix{Int64}:
 10  14  17  17  20
 10  12  13  18  20
 10  13  17  18  18
 10  11  11  13  15
 10  13  15  16  19
 13  14  14  18  20
```

Рис. 4: Примеры. Массивы

Примеры

```
[21]  findall(ar .> 14)
```

```
26-element Vector{CartesianIndex{2}}:
 CartesianIndex(1, 1)
 CartesianIndex(3, 1)
 CartesianIndex(5, 1)
 CartesianIndex(6, 1)
 CartesianIndex(10, 1)
 CartesianIndex(1, 2)
 CartesianIndex(7, 2)
 CartesianIndex(8, 2)
 CartesianIndex(10, 2)
 CartesianIndex(3, 3)
 CartesianIndex(4, 3)
 CartesianIndex(5, 3)
 CartesianIndex(7, 3)
 CartesianIndex(8, 3)
 CartesianIndex(9, 3)
 CartesianIndex(1, 4)
 CartesianIndex(2, 4)
 CartesianIndex(3, 4)
 CartesianIndex(6, 4)
 CartesianIndex(7, 4)
 CartesianIndex(9, 4)
 CartesianIndex(2, 5)
 CartesianIndex(5, 5)
 CartesianIndex(8, 5)
 CartesianIndex(9, 5)
 CartesianIndex(10, 5)
```

Рис. 5: Примеры. Массивы

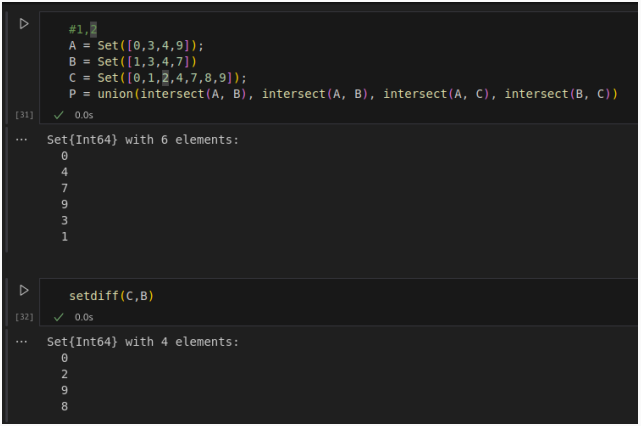


Рис. 6: Задание 1,2

Самостоятельные задания

Рис. 7: Задание 3

Самостоятельные задания

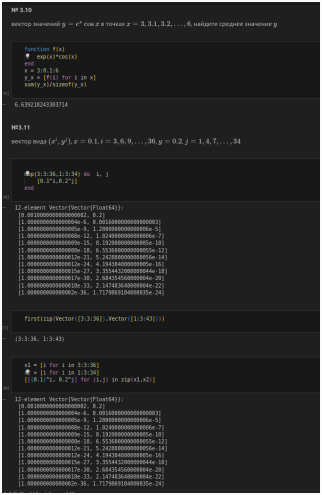


Рис. 8: Задание 3

Самостоятельные задания

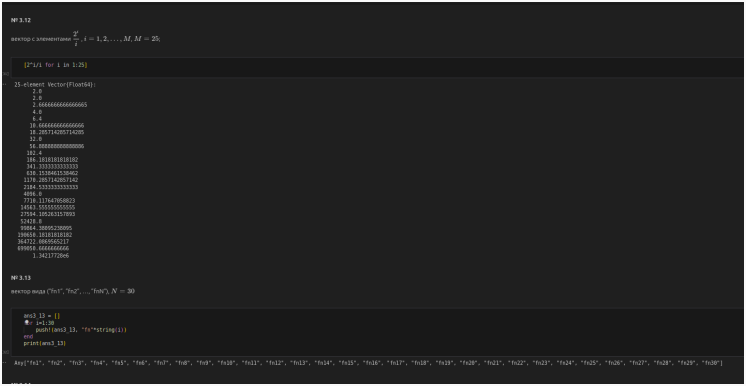


Рис. 9: Задание 3

Самостоятельные задания

№ 3.14

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

- сформируйте вектор $(y_2 - x_1, \ldots, y_n - x_{n-1})$;
- сформируйте вектор $(x_1 + 2x_2 - x_3, x_2 + 2x_3 - x_4, \ldots, x_n - 2 + 2x_{n-1} - x_n)$;
- сформируйте вектор $\left(\frac{\sin y_1}{\cos x_2}, \frac{\sin y_2}{\cos x_3}, \ldots, \frac{\sin y_{n-1}}{\cos x_n}\right)$;
- вычислите $\sum_{i=1}^{n-1} \frac{e^{-y_i - i}}{x_i + 10}$
- 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 $(|x_1 - \overline{x}|^{\frac{1}{4}}, |x_2 - \overline{x}|^{\frac{1}{4}}, \ldots, |x_n - \overline{x}|^{\frac{1}{4}})$, где $\overline{x}$ обозначает среднее значение вектора $x = (x_1, x_2, \ldots, x_n)$;
- 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.

```
n = 250
x = [rand(0:999) for i=1:n];
y = [rand(0:999) for i=1:n];
using Statistics
```

(1)

```
tmp1 = [y[i+1]-x[i] for i=1:n-1];
tmp2 = [x[i]+2*x[i+1]-x[i+2] for i=1:n-2];
tmp3 = [sin(y[i])/cos(x[i+1]) for i=1:n-1];
tmp4 = sum([exp(-x[i+1])/(x[i]+10) for i=1:n-1]);
tmp5_ind = findall(y.>600);
tmp5_el = filter(e->e.>600, y);
tmp6_1 = [x[i] for i=1:250 if(y[i]>600)];
tmp6_2 = [x[i] for i in tmp5_ind];
mean_x = mean(x);
tmp7 = [abs(x[i]-mean(x)) for i=1:n];
tmp8 = sum(x.>maximum(x).-200);
tmp9_odd = sum(map(isodd, x));
tmp9_even = sum(map(iseven, x));
tmp10 = x[sortperm(y)];
tmp11 = last(sort(x),10);
tmp12 = collect(Set(x));
```

(6)

Рис. 10: Задание 3

Самостоятельные задания

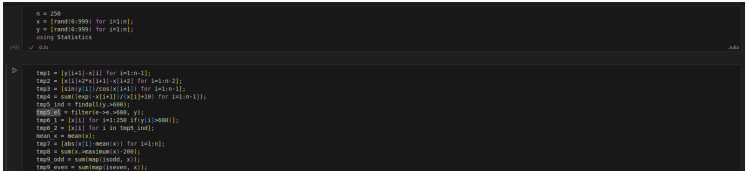


Рис. 11: Задание 3

Самостоятельные задания

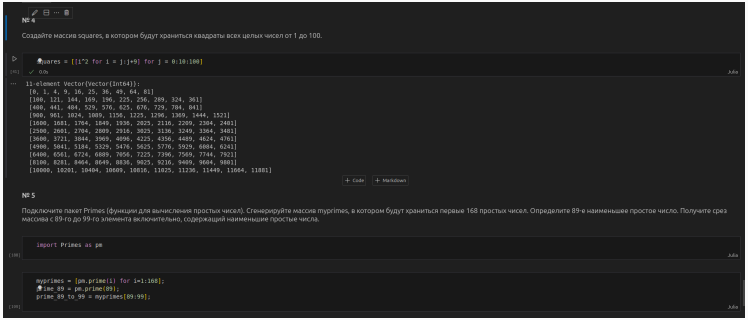


Рис. 12: Задание 4

Самостоятельные задания

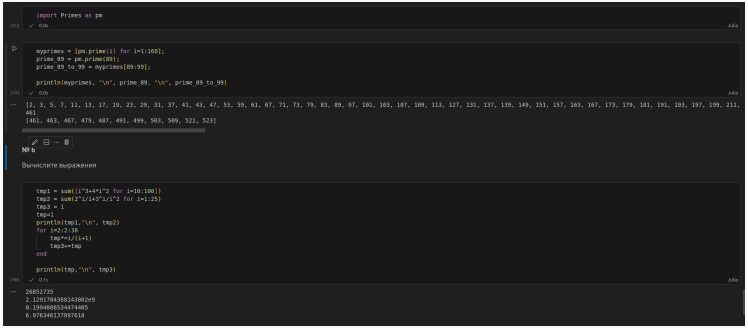


Рис. 13: Задание 5,6

Выводы

Выводы

В результате выполнения работы изучили несколько структур данных, реализованных в Julia, научились применять их и операции над ними для решения задач.

Список литературы

1. JuliaLang [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: https: //julialang.org/ (дата обращения: 11.10.2024).
2. Julia 1.11 Documentation [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: https://docs.julialang.org/en/v1/ (дата обращения: 11.10.2024).