Презентация по лабораторной работе №1 по предмету Компьютерный практикум по статистическому анализу данных

СТУДЕНТ ГРУППЫ НФИБД-01-20 ЕВДОКИМОВ МАКСИМ МИХАЙЛОВИЧ (1032203019)

Цель работы

- Изучите документацию по основным функциям Julia для чтения / записи / вывода информации на экран: read(), readline(), readlines(), readdlm(), print(), println(), show(), write(). Приведите свои примеры их использования, поясняя особенности их применения.
- Изучите документацию по функции parse(). Приведите свои примеры её использования, поясняя особенности её применения.
- Изучите синтаксис Julia для базовых математических операций с разным типом переменных: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, сравнение, логические операции. Приведите свои примеры с пояснениями по особенностям их применения.
- Приведите несколько своих примеров с пояснениями с операциями над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр.

```
using Random
using DelimitedFiles

print("specify the number of columns and rows (separated by a space):\n")
a = split(readline(), ' ')
rows, columns = parse(Int, a[1]), parse(Int, a[2])
show(rows)
file = open("data//lab1_data.txt", "w")
for r in 1:rows
    for c in 1:columns
        if c == columns
            write(file, string(rand(0:20))*"\n")
        else
            write(file, string(rand(0:20))*" ")
        end
    end
end
close(file)
```

```
specify the number of columns and rows (separated by a space):
stdin>  3 3
3
```

Выполнение работы: Начало и первый файл

Начнём с того что внесём в программу библиотеки Random для случайного заполнения матриц в файлах и DelimitedFiles для использования методов таких как «readdlm». Также создаем первый файл и указываем размер двумерного массива, который будет записан в него.

```
file = open("data//lab1_data.txt", "r")
println("Выберите способ чтения массива (1->read, 2->readlines, 3->readline):")
input = readline()
if input == "1"
    m = [[parse(Float64, el) for el in split(line, ' ')] for line in split(read(file, String), "\n")[1:rows]]
elseif input == "2"
    m = [[parse(Float64, el) for el in split(line, ' ')] for line in readlines(file)]
else
    m = [[parse(Float64, el) for el in split(readline(file), ' ')] for _ in 1:rows]
end
show(file)
close(file)
show(m)
massive1 = hcat(m...)
```

```
Выберите способ чтения массива (1->read, 2->readlines, 3->readline):
stdin>  2
IOStream(<file data//lab1_data.txt>)[[1.0, 10.0, 19.0], [18.0, 14.0, 2.0], [15.0, 9.0, 11.0]]
3×3 Matrix{Float64}:
  1.0  18.0  15.0
 10.0  14.0   9.0
 19.0   2.0  11.0
```

Выполнение работы: чтение файла и создание матрицы

Проверяем файл открывая его, а также все доступные команды для считывания данных в строковом виде. И сохраняем полученные данные в виде Matrix из float64 элементов.

```
x = [rand(1:9, columns) for _ in 1:rows]
open("data//lab1_dlm.txt", "w") do file2
    writedlm(file2, [row for row in x])
end
massive2 = readdlm("data//lab1_dlm.txt", Float64)
```

```
3×3 Matrix{Float64}:
 7.0  3.0  9.0
 8.0  8.0  7.0
 4.0  5.0  5.0
```

Выполнение работы: использование DelimitedFiles

Используя библиотеку DelimitedFiles и его методы создаём второй файл с равным по размеру матрицей из первого файла и сразу выводим эти данные другой командой.

```
function ArithmeticOperators(a, b)
    println("a = $a и b = $b")
    println("Сумма: ", a + b)
    println("Вычитание: ", a - b)
    println("Умножение: ", a * b)
    println("Деление a на b: ", a / b)
    println("Деление b на a: ", a \ b)
    println("Целочисленное деление: ", a ÷ b)
    println("Возведение a в b: ", a ^ b)
    println("Остаток от деления a на b: ", a % b)
end

function NumericBitwiseComparisons(a, b)
    println("a = $a и b = $b")
    println("Ровны ли? ", a == b)
    println("Не ровны ли? ", a != b)
    println("a больше b? ", a > b)
    println("a больше или ровно b? ", a >= b)
    println("a меньше b? ", a < b)
    println("a меньше или ровно b? ", a <= b)
    println("логическое и: ", a & b)
    println("логическое или: ", a | b)
    println("логическое хор: ", a ⊻ b)
    println("логическое нет и: ", a ⊼ b)
    println("логическое нет или: ", a ⊽ b)
end
```

```
function MatrixVectorOperations(a, b)
    println("a = $a\nb = $b")
    println("Сумма:\n", a .+ b)
    println("Вычитание:\n", a .- b)
    println("Умножение:\n", a .* b)
    println("Деление a на b:\n", a ./ b)
    println("Деление b на a:\n", a .\ b)
    println("Целочисленное деление:\n", a .÷ b)
    println("Возведение a в b:\n", a .^ b)
    println("Остаток от деления a на b:\n", a .% b)
end
```

Выполнение работы: функции с лог. и мат. операциями

Теперь чтобы проверить все перечисленные и не только математические операции я создал 3 функции: первая проводит обычные математические операции над двумя числами, вторая логические.

```
ArithmeticOperators(rand(1:9), rand(1:9))
NumericBitwiseComparisons(rand(1:9), rand(1:9))
```

```
a = 4 и b = 8
Сумма: 12
Вычитание: -4
Умножение: 32
Деление a на b: 0.5
Деление b на a: 2.0
Целочисленное деление: 0
Возведение a в b: 65536
Остаток от деления a на b: 4
a = 1 и b = 7
Ровны ли? false
Не ровны ли? true
a больше b? false
a больше или ровно b? false
a меньше b? true
a меньше или ровно b? true
логическое и: 1
логическое или: 7
логическое xor: 6
логическое нет и: -2
логическое нет или: -8
```

Выполнение работы: работа с числами

И вызываю данные функции задавая случайное значение от 1 до 9 для первых двух.

```
MatrixVectorOperations(massive1, massive2)
```

```
a = [1.0 18.0 15.0; 10.0 14.0 9.0; 19.0 2.0 11.0]
b = [7.0 3.0 9.0; 8.0 8.0 7.0; 4.0 5.0 5.0]
Сумма:
[8.0 21.0 24.0; 18.0 22.0 16.0; 23.0 7.0 16.0]
Вычитание:
[-6.0 15.0 6.0; 2.0 6.0 2.0; 15.0 -3.0 6.0]
Умножение:
[7.0 54.0 135.0; 80.0 112.0 63.0; 76.0 10.0 55.0]
Деление a на b:
[0.14285714285714285 6.0 1.6666666666666667; 1.25 1.75 1.2857142857142858; 4.75 0.4 2.2]
Деление b на a:
[7.0 0.16666666666666666 0.6; 0.8 0.5714285714285714 0.7777777777777778; 0.21052631578947367 2.5 0.45454545454545453]
Целочисленное деление:
[0.0 6.0 1.0; 1.0 1.0 1.0; 4.0 0.0 2.0]
Возведение a в b:
[1.0 5832.0 3.8443359375e10; 1.0e8 1.475789056e9 4.782969e6; 130321.0 32.0 161051.0]
Остаток от деления a на b:
[1.0 0.0 6.0; 2.0 6.0 2.0; 3.0 2.0 1.0]
```

```
MatrixVectorOperations(rand(1:9, 10), rand(1:9, 10))
```

```
a = [5, 5, 7, 9, 8, 4, 7, 9, 8, 9]
b = [9, 3, 9, 2, 6, 2, 4, 8, 1, 2]
Сумма:
[14, 8, 16, 11, 14, 6, 11, 17, 9, 11]
Вычитание:
[-4, 2, -2, 7, 2, 2, 3, 1, 7, 7]
Умножение:
[45, 15, 63, 18, 48, 8, 28, 72, 8, 18]
Деление a на b:
[0.5555555555555556, 1.6666666666666667, 0.7777777777777778, 4.5, 1.3333333333333333, 2.0, 1.75, 1.125, 8.0, 4.5]
Деление b на a:
[1.8, 0.6, 1.2857142857142858, 0.2222222222222222, 0.75, 0.5, 0.5714285714285714, 0.8888888888888888, 0.125, 0.2222222222222222]
Целочисленное деление:
[0, 1, 0, 4, 1, 2, 1, 1, 8, 4]
Возведение a в b:
[1953125, 125, 40353607, 81, 262144, 16, 2401, 43046721, 8, 81]
Остаток от деления a на b:
[5, 2, 7, 1, 2, 0, 3, 1, 0, 1]
```

Выполнение работы: работа с матрицами и векторами

И для последней функции смотрим как работают стандартные математические операции для матриц и векторов.

Заключение

- Повторил основы работы с числовыми и строковыми типами данных и их векторами, и матрицами. Ознакомился с работой на Julia в Jupyter, а также изучил и привёл примеры математических и логических операций.