Отчет по лабораторной работе №3

Компьютерный практикум по статистическому анализу данных

Амуничников Антон Игоревич

Содержание

# 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. Задание

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

# 3. Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений [**julialang?**]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia [**juliadoc?**].

# 4. Выполнение лабораторной работы

Для начала выполним примеры из лабораторной работы, чтобы познакомиться с циклами, условными операторами, функциями и работой со сторонними библиотеками ([рис. 1](#fig-001) - [рис. 4](#fig-004)).

|  |
| --- |
| Рисунок 1: Выполнение примеров с циклами |

|  |
| --- |
| Рисунок 2: Выполнение примеров с условными выражениями |

|  |
| --- |
| Рисунок 3: Выполнение примеров с функциями |

|  |
| --- |
| Рисунок 4: Выполнение примеров со сторонними библиотеками |

Теперь перейдем к выполнению заданий для самостоятельной работы.

Используя циклы while и for ([рис. 5](#fig-005)):

* выведем на экран целые числа от 1 до 100 и напечатаем их квадраты;
* создадим словарь squares, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;
* создадим массив squares\_arr, содержащий квадраты всех чисел от 1 до 100.

|  |
| --- |
| Рисунок 5: Задание №1 |

Напишем условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Перепишем код, используя тернарный оператор ([рис. 6](#fig-006)).

|  |
| --- |
| Рисунок 6: Задание №2 |

Напишем функцию add\_one, которая добавляет 1 к своему входу ([рис. 7](#fig-007)).

|  |
| --- |
| Рисунок 7: Задание №3 |

Используем map() или broadcast() для задания матрицы , каждый элемент которой увеличивается на единицу по сравнению с предыдущим. ([рис. 8](#fig-008))

|  |
| --- |
| Рисунок 8: Задание №4 |

Зададим матрицу . Найдем . Заменим третий столбец матрицы на сумму второго и третьего столбцов ([рис. 9](#fig-009)).

|  |
| --- |
| Рисунок 9: Задание №5 |

Создадим матрицу с элементами , , , . Вычислим матрицу ([рис. 10](#fig-010)).

|  |
| --- |
| Рисунок 10: Задание №6 |

Создадим матрицу размерности 6×6, все элементы которой равны нулю, и матрицу , все элементы которой равны 1. Используя цикл while или for и закономерности расположения элементов, создадим следующие матрицы размерности 6×6 ([рис. 11](#fig-011)).

|  |
| --- |
| Рисунок 11: Задание №7 |

Напишем свою функцию, аналогичную функции outer() языка R. Функция должна иметь следующий интерфейс: outer(x,y,operation) ([рис. 12](#fig-012)), ([рис. 13](#fig-013)).

|  |
| --- |
| Рисунок 12: Задание №8. Реализация функции outer(). Проверка работы функции |

|  |
| --- |
| Рисунок 13: Задание №8. Проверка работы функции outer() |

Решим систему линейных уравнений с 5 неизвестными ([рис. 14](#fig-014)).

|  |
| --- |
| Рисунок 14: Задание №9. Решение систему линейных уравнений |

В 10 задании произведем анализ количества элементов матрицы, удовлетворяющих необходимым условиям. Вычислим выражения ([рис. 15](#fig-015)).

|  |
| --- |
| Рисунок 15: Задание №10. Задание №11 |

# 5. Выводы

В резульатте выполнения данной лабораторной работы я освоил применение циклов функций и сторонних для Julia пакетов для решение задач линейной алегбры и работы с матрицами.