РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

ОТЧЕТ

по лабораторной работе № 1

дисциплина: Компьютерный практикум   
по статистическому анализ данных

Студент: Доре Стевенсон Эдгар

Группа: НКН-бд-01-19

**МОСКВА**

2022 г.

# Постановка задачи

Основная цель работы — подготовить рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомиться с основами синтаксиса Julia.

1. Установите под свою операционную систему Julia, Jupyter (разделы 1.3.1 и 1.3.2).

2. Используя Jupyter Lab, повторите примеры из раздела 1.3.3.

3. Выполните задания для самостоятельной работы (раздел 1.3.4).

**Выполнение работы**

1. Установите под свою операционную систему Julia, Jupyter (разделы 1.3.1 и 1.3.2).

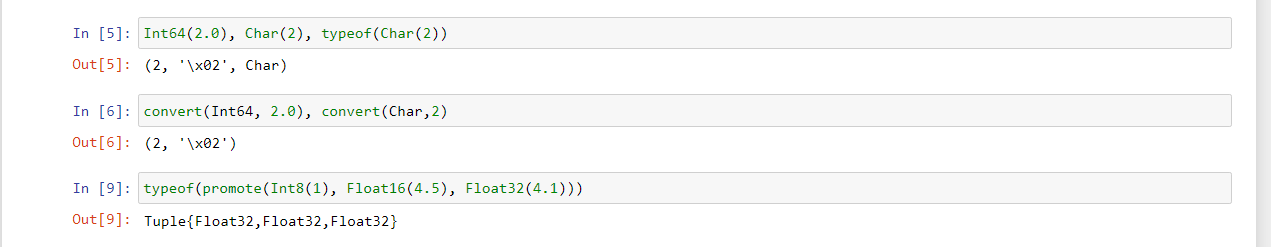
Посмотрели ознакомительное видео и выполнили установку требуемых приложений

1. Используя Jupyter Lab, повторите примеры из раздела 1.3.3.

Получим минимальные и максимальные значения целочисленных типов



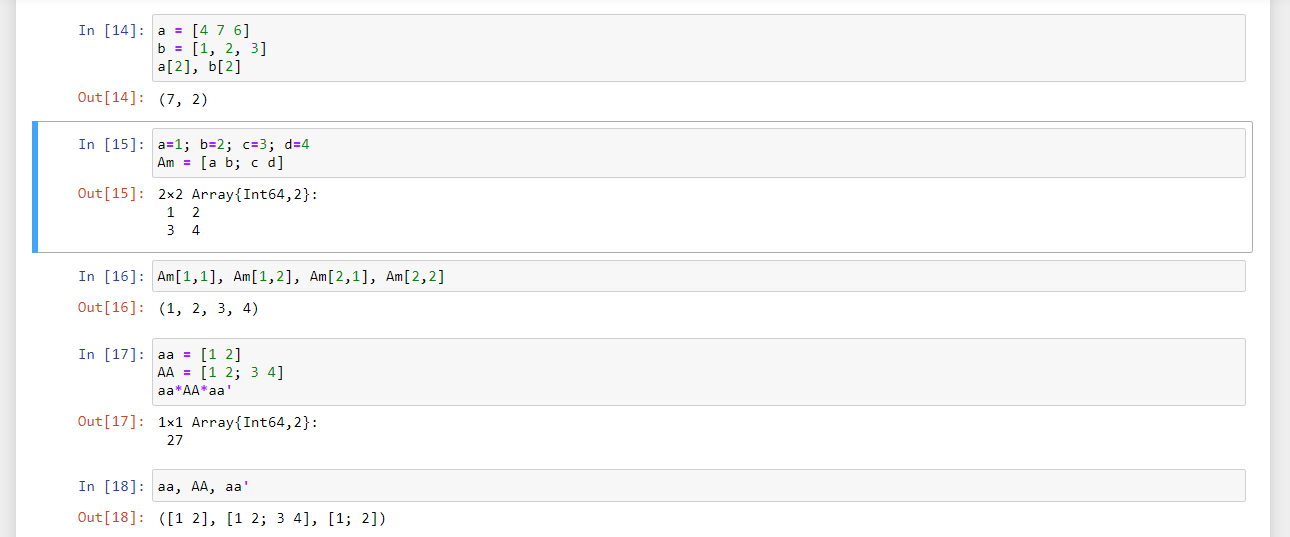
Примеры приведения типов



Примеры работы с функциями



Примеры работы с матрицами



1. Выполните задания для самостоятельной работы (раздел 1.3.4).
2. 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

* read() – считывает значение указанного типа в бинарном представлении

read(stream, type)



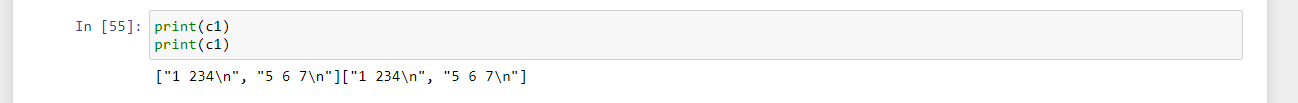
* readline() – считывает строку до символа завершения строки, включая этот символ



* readlines() – считывает все строки, поданные на вход в виде массива



* print() – распечатывает на экран содержимое переменной



* println() – распечатывает на экран с новой строки содержимое переменной



* show() – распечатывает на экран содержимое переменной, для сложных выражений выполняет форматирование, можно указать поток вывода

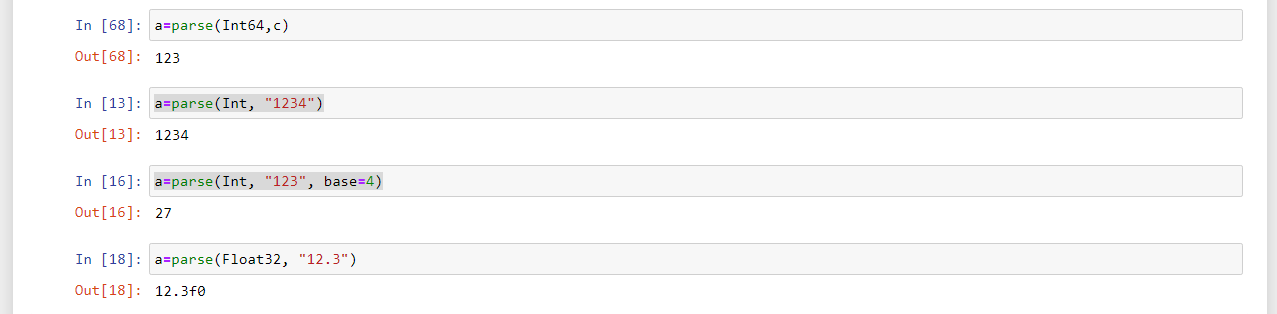


* write() – выполняет запись содержимого переменной в указанный поток, возвращает число записанных байт



1. Изучите документацию по функции parse(). Приведите свои примеры её использования, поясняя особенности её применения.

Функция parse() преобразовывает строку в численный тип, можно указать основание системы счисления и поток ввода.



1. 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.

Синтаксис предполагает использование стандартный символов в качестве обозначения операций

+ Сложение

- Вычитание, унарный минус

\* умножение

/ правостороннее деление

\ левостороннее деление

^ степень, работает также для матриц

fma() вычисление без округления

inv() обратное число

div() целочисленное деление

mod() остаток от деления

== сравнение на равенство

!= сравнение на неравество

< сравнение на меньшее

> сравнение на большее

<= меньшее либо равное

>= большее либо равное

~ побитовое НЕ

& побитовое И

| побитовое ИЛИ

! логическое НЕ

xor() побитовое ИСКЛ ИЛИ

&& логическое И

|| логическое ИЛИ

sin(), cos(), tan() тригонометрические в радианах

sind(), cosd(), tand() тригонометрические в градусах

sinh(), cosh(), tanh() гиперболические тригонометрические функции

Для всех тригонометрических функций реализованы обратные функции

log(х) натуральный логарифм

log(b, x) логарифм по основанию b

exp() экспонента e^x

exp2() 2^x

exp10() 10^x

round() математическое округление

abs() модуль

abs2() квадрат модуля

sqrt() квадратный корень

cbrt() кубический корень

factorial() факториал



1. Приведите несколько своих примеров с пояснениями с операциями над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр.

