Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Воронежская государственная лесотехническая академия»

Е.А. Аникеев

Методические указания

для выполнения лабораторных работ по дисциплине

«Технологии обработки информации»

для студентов специальности

09.03.02 – Информационные системы и технологии

**Воронеж 2019**

Рецензент: начальник лаборатории ОАО НИИЭТ

к-т. техн. наук А.И. Яньков

**Аникеев, Е.А.**

Методические указания по выполнению лабораторных работ по дисциплине «Технологии обработки информации» для студентов специальности 09.03.02 – «Информационные системы и технологии» / Е.А. Аникеев – Воронеж: Воронежская государственная лесотехническая академия, 2019. – 67с.

**Аникеев** Евгений Александрович

Методические указания по выполнению лабораторных работ по дисциплине «Технологии обработки информации» для студентов специальности 09.03.02 – «Информационные системы и технологии»

СОДЕРЖАНИЕ

[Правила оформления отчёта 5](#_Toc53486250)

[Лабораторная работа №1. Основы работы в среде SciLab 6](#_Toc53486251)

[Теоретическая часть 6](#_Toc53486252)

[Порядок выполнения работы 16](#_Toc53486253)

[Варианты заданий 17](#_Toc53486254)

[Контрольные вопросы 17](#_Toc53486255)

[Лабораторная работа №2. Описательная статистика 17](#_Toc53486256)

[Теоретическая часть 17](#_Toc53486257)

[Порядок выполнения работы 19](#_Toc53486258)

[Варианты заданий 20](#_Toc53486259)

[Контрольные вопросы 20](#_Toc53486260)

[Лабораторная работа №3. Комбинаторика 21](#_Toc53486261)

[Теоретическая часть 21](#_Toc53486262)

[Порядок выполнения работы 22](#_Toc53486263)

[Варианты заданий 22](#_Toc53486264)

[Контрольные вопросы 22](#_Toc53486265)

[Лабораторная работа №4. Параметры генеральной совокупности и выборки 23](#_Toc53486266)

[Теоретическая часть 23](#_Toc53486267)

[Порядок выполнения работы 25](#_Toc53486268)

[Варианты заданий 27](#_Toc53486269)

[Контрольные вопросы 27](#_Toc53486270)

[Лабораторная работа №5.Построение нечётких множеств и операции над ними 28](#_Toc53486271)

[Теоретическая часть 28](#_Toc53486272)

[Порядок выполнения работы 28](#_Toc53486273)

[Варианты заданий 34](#_Toc53486274)

[Контрольные вопросы 34](#_Toc53486275)

[Лабораторная работа №6.Вычисление нечётких выражений 35](#_Toc53486276)

[Теоретическая часть 35](#_Toc53486277)

[Порядок выполнения работы 36](#_Toc53486278)

[Варианты заданий 38](#_Toc53486279)

[Контрольные вопросы 38](#_Toc53486280)

[Лабораторная работа №7. Расчёт параметров выборки данных 39](#_Toc53486281)

[Теоретическая часть 39](#_Toc53486282)

[Порядок выполнения работы 42](#_Toc53486283)

[Лабораторная работа №8. Использование Python для обработки данных 43](#_Toc53486284)

[Теоретическая часть 43](#_Toc53486285)

[Порядок выполнения работы 46](#_Toc53486286)

[Лабораторная работа №9. Анализ выборки данных при помощи Python 48](#_Toc53486287)

[Теоретическая часть 48](#_Toc53486288)

[Порядок выполнения работы 51](#_Toc53486289)

[Лабораторная работа №10. Измерение центральной тенденции 54](#_Toc53486290)

[Теоретическая часть 54](#_Toc53486291)

[Порядок выполнения работы 55](#_Toc53486292)

[Варианты заданий 58](#_Toc53486293)

[Лабораторная работа №11. Измерение вариации 58](#_Toc53486294)

[Теоретическая часть 58](#_Toc53486295)

[Порядок выполнения работы 60](#_Toc53486296)

[Варианты заданий 61](#_Toc53486297)

[Лабораторная работа №12. Визуализация данных 62](#_Toc53486298)

[Теоретическая часть 62](#_Toc53486299)

[Порядок выполнения работы 63](#_Toc53486300)

[Варианты заданий 66](#_Toc53486301)

[Список литературы 67](#_Toc53486302)

## Правила оформления отчёта

Отчёт по лабораторной работе оформляется студентом в отдельной от лекций рабочей тетради, не использующей сменные блоки листов.

Содержание отчёта:

1. Титульный лист.
2. Цель лабораторной работы.
3. Теоретическое введение.
4. Выполнение работы.
5. Вывод по лабораторной работе.

Титульный лист содержит: наименования вышестоящей организации, учебного заведения, кафедры, на которой выполняется работа, а также номер и название самой работы. Также указывается, кто выполнил работу (фамилия, инициалы и номер группы студента) и кто проверил работу (учёные степень, звание, фамилия и инициалы). В нижней части титульного листа указывается город и годы выполнения работы.

Цель выполнения приведена в описании лабораторной работы. Она размещается на странице, следующей за титульным.

Теоретическое введение содержит основные положения, на которых основана выполняемая работа. Его объем должен составлять от половины до одной страницы.

В разделе «Выполнение работы» приводятся исходные данные, вычисления и схемы, необходимые для достижения цели работы. Подробное содержание этого раздела приводится в разделе «Порядок выполнения работы» для каждой лабораторной работы.

Вывод содержит изложение результатов работы, полученных в ходе её выполнения, с учётом цели работы.

## Лабораторная работа №1. Основы работы в среде SciLab

Цель работы: получение навыков выполнения основных арифметических операций, использования операторов для создания скриптов и построения графиков в пакете прикладных математических программ для технических и научных расчётов Scilab.

### Теоретическая часть

Scilab (читается Сайлэб) – пакет прикладных математических программ, предоставляющий открытое окружение для инженерных (технических) и научных расчётов. Это самая полная общедоступная альтернатива MATLAB.

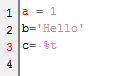
Пакет Scilab является кроссплатформенной системой компьютерной алгебры и обладает сходным с Matlab синтаксисом встроенного языка. Разработка системы Scilab ведется сотрудниками французского Национального института информатики и автоматизации (INRIA – Institut National de Recherche en Informatique et Automatique) с 80-х годов прошлого века. Изначально это был коммерческий проект под названием Blaise, а затем Basile. С 2003 года продукт получил новое имя Scilab и стал бесплатным. В настоящее время он распространяется по свободной лицензии CeCILL.

Сама система Scilab, как и Matlab, предназначена прежде всего для численных расчетов и работы с матрицами. Кроме того, она обладает развитыми средствами программирования (включая отладчик скриптов), так что ее в какой-то мере можно рассматривать как систему разработки высокотехнологичных приложений.Для системы имеется достаточно большое число пакетов расширений, которые можно найти на официальном сайте в разделе Toolbox center.

В Scilab существует два режима выполнения расчётов: консольный и скриптовый.

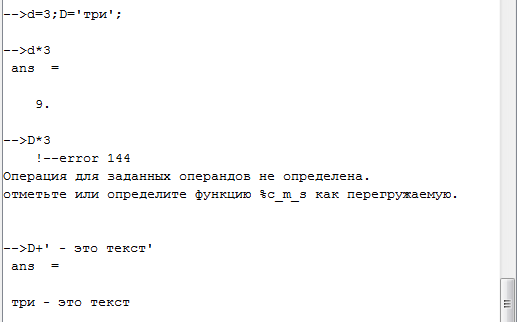
Встроенный язык Scilab – это язык структурного программирования. Весь выполняемый код размещается в функциях. В одном файле может быть несколько функций. Однако при разработке пакетов расширений принято хранить каждую функцию в отдельном файле.

Переменные не описываются, а создаются путем присвоения им начального значения, например так:

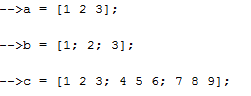


Переменные в Scilab не имеют строгой типизации, т. е. если в переменной хранился текст, то можно на следующем шаге записать в нее число, а затем логическое значение. Scilab следит за соответствием типов только при вычислении значений выражений.

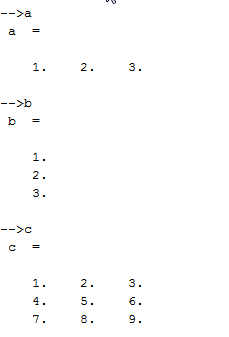
Вследствие Unix-корней системы, важен регистр букв в имени переменных, например:



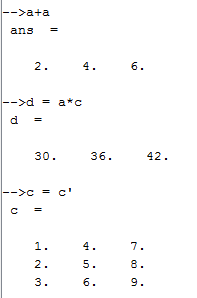
В Scilab можно производить различные математические операции с матрицами. Строки матриц отделяются друг от друга при помощи символа «;». Определим матрицу-строку , матрицу-столбец  и двумерную матрицу :



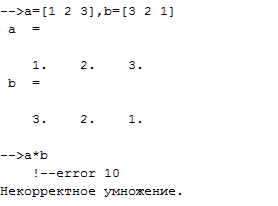
Символ «;» в конце строки запрещает немедленный вывод результата (в текущей версии программы не работает). Заданные матрицы выглядят так:



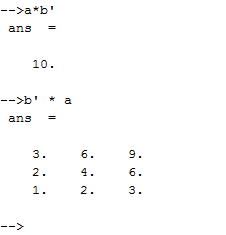
С матрицами можно производить различные действия, например складывать, умножать, транспонировать и т.д.



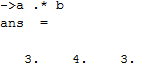
При этом используются следующие знаки действий: + (сложение), – (вычитание), \* (умножение), / (деление), ^ (возведение в степень), ' (транспонирование). Изначально эти операции служат для выполнения матричных действий по правилам матричной алгебры, но они используются и для операций с числами, которые представлены в системе как одномерные матрицы. При выполнении этих действий необходимо помнить о правилах матричной алгебры. Например:



Здесь сделана попытка перемножить две строки, но по правилам матричной алгебры это нельзя сделать. Одну из строк необходимо транспонировать, чтобы получился столбец. Кроме того, согласно правилам матричной алгебры, важен порядок множителей:



Для выполнения поэлементного умножения двух массивов необходимо использовать признак поэлементного действия, т. е. поставить перед знаком действия точку (точка и знак действия пишутся слитно, без пробела):

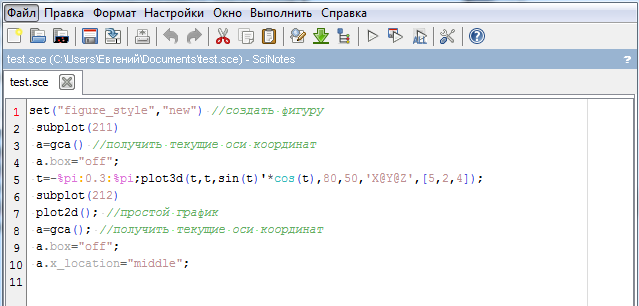


То же самое относится и ко всем остальным действиям, кроме операции транспонирования.

Для удобства написания функций в Scilab имеется встроенный редактор Scinotes. Он позволяет редактировать тексты функций, сохранять их в скриптах (файлах с расширением .sci) выполнять их в режиме отладки, содержит функцию автодополнения кода, а также средства непосредственной передачи текста программы в среду Scilab на выполнение.

Открыть редактор можно двумя способами:

* подать в консоли Scilab команду scinotes
* выбрать в главном окне последовательно пункты меню Инструменты/Редактор.



В данном курсе используется версия Scilab 5.5.1, в которой присутствует редактор Scinotes. Внешний вид редактора показан на рисунке. Кратко рассмотрим пункты меню:

**Файл** – здесь находятся стандартные команды для работы с файлами.

**Правка** – содержит стандартные для пункта меню Правка операции: копировать, вставить и т. д.

**Формат** – содержит команды форматирования текста в редакторе.

**Настройки** – здесь находится довольно много пунктов, которые позволяют настроить внешний вид и поведение редактора от типа шрифта до горячих клавиш. Например, можно выбрать комбинацию клавиш, которая будет использоваться для вызова функции автодополнения кода.

**Окно** – команды управления рабочим окном. Позволяют разбить окно на части по вертикали и горизонтали, а также упорядочить размещение частей окна.

**Выполнить** – содержит пункты, позволяющие передать содержимое редактора в среду Scilab на выполнение или выполнить только выделенную часть.

**Справка –** это меню обеспечивает доступ к справочным материалам среды Scilab.

Сохранить и открыть созданный скрипт можно при помощи пунктов меню Файл.

Чтобы выполнить написанный скрипт, функцию, ее необходимо передать в основное окно Scilab (консоль). Для этого необходимо воспользоваться меню Выполнить, в котором присутствуют три пункта:

...**файл без отображения команд** – загрузить в Scilab текущий скрипт, команды скрипта не показываются в командной строке Scilab;

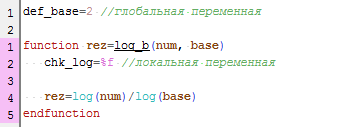
...**файл с отображением команд** – загрузить в Scilab текущий скрипт, команды скрипта показываются в командной строке Scilab;

...**до курсора с отображением команд** – загрузить в Scilab текущий скрипт, команды скрипта показываются в командной строке Scilab и выполняются до текущей позиции курсора в редакторе.;

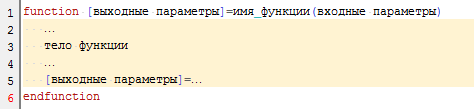
Чтобы выполнить скрипт полностью без отображения команд, надо на верхней панели редактора нажать кнопку «Выполнить».



Переменные, созданные внутри функции, являются локальными и действуют только в пределах этой функции. Переменные, созданные в пространстве до начала функции, являются глобальными и доступны во всех функциях данного файла или текущей рабочей сессии.

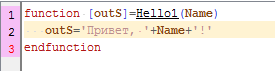


В общем случае описание функции выглядит следующим образом:

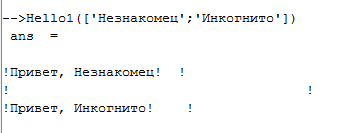


Если у функции всего один выходной параметр, то его можно не заключать в квадратные скобки, если же их больше одного, то они заключаются в скобки и перечисляются через запятую.

Создадим в редакторе Scinotes стандартную функцию, которая будет приветствовать всех по имени:



Теперь вызовем эту функцию в командной строке Scilab (не в редакторе!), указав в качестве входного параметра не одно значение, а массив значений:



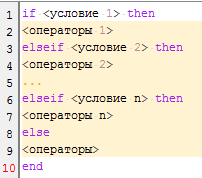
Scilab не выдал никакого предупреждения о том, что входной параметр является массивом. Поэтому, создавая функцию, нужно помнить, что входной параметр может быть массивом, и в критических случаях предусматривайте выполнение проверки на размер массива.

В пользовательских скриптах можно использовать любые сторонние функции, входящие в состав самой системы Scilab или ее пакетов-расширений.

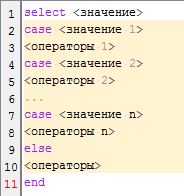
В языке Scilab две конструкции, организующие ветвление:

1. Конструкция if ... else;
2. Конструкция select ... case.

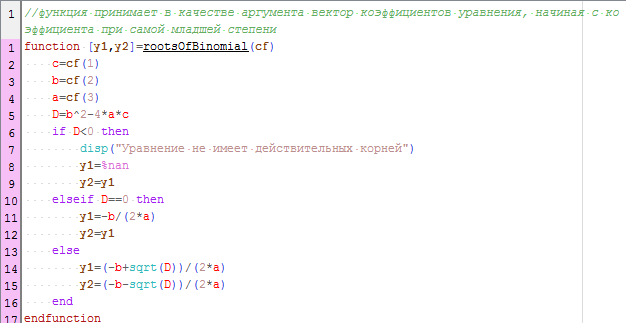
Общий синтаксис конструкции if ... else имеет следующий вид



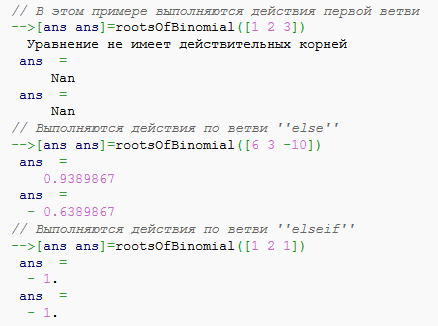
Общий синтаксис второй конструкции имеет вид



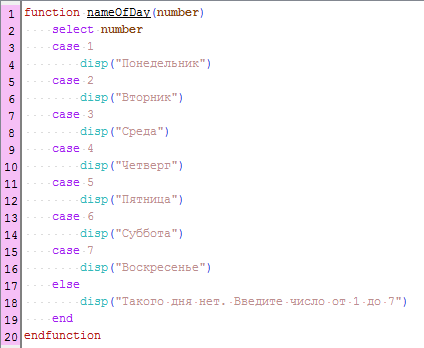
Для иллюстрации конструкций ветвления создадим функцию для расчета корней квадратного уравнения.



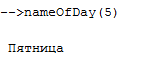
Результат вызова функции в командной строке Scilab:



Вторую конструкцию продемонстрируем на следующем примере. Пользователь передает функции число, которое означает номер дня недели. В ответ функция выводит название этого дня.



Результат выполнения этой функции:



Для выполнения повторяющихся операций в Scilab предусмотрены два вида циклов:

1. Цикл for;
2. Цикл while.

Общий синтаксис для цикла for имеет вид



Цикл for, как правило, используется для перебора элементов с операциями над ними на некотором ограниченном множестве, либо когда количество итераций нам известно. Цикл будет выполняться до тех пор, пока счетчик не станет больше введенного конечного значения. Шаг является не обязательным параметром для счетчика и по умолчанию он принимается равным единице.

Для цикла while синтаксис имеет следующий вид



Цикл while выполняется до тех пор, пока условие истинно. Обычно данный цикл используется, когда число итераций нам заранее неизвестно. С данным циклом необходимо быть особенно внимательным, так как зачастую можно задать условие, которое будет истинно всегда. В этом случае цикл становится бесконечным и имеет место так называемое зацикливание.

Иногда довольно сложно записать точное условие выхода из цикла явно. В этом случае нарочно создают бесконечный цикл, а условие выхода задают уже в теле с помощью специальных управляющих операторов.

Управление циклом в его теле возможно с помощью следующих специальных операторов:

**continue** — передача управления следующей итерации;

**break** — прекращение текущей итерации и выход из цикла.

Для примера напишем функцию, решающую задачу поиска минимального и максимального значения в массиве данных. Под массивом данных мы будем понимать числовую матрицу и числовой вектор. Будем использовать алгоритм простого перебора.

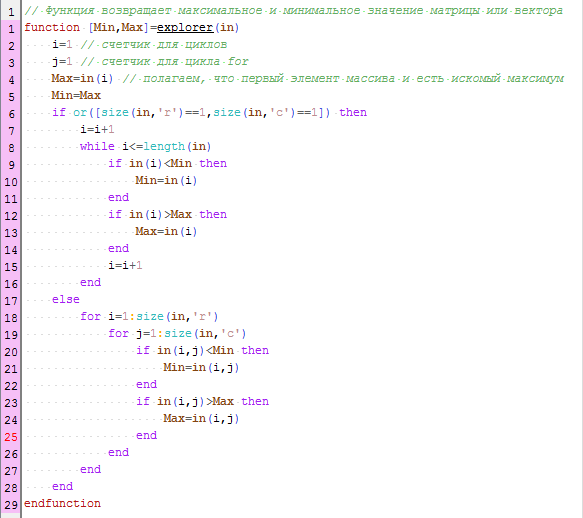
Положить, что первый элемент массива есть искомый минимум (максимум) и записать его во временную переменную;

Сравнить значение i-го элемента (начиная со второго) со значением во временной переменной: если i-ый элемент меньше (больше) него, то записать i-ый элемент во временную переменную, иначе перейти на следующий шаг;

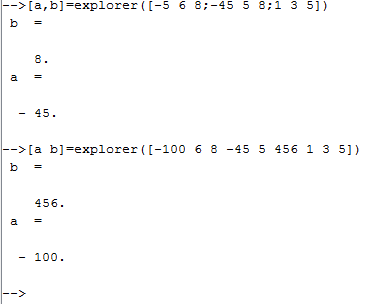
Увеличить значение счетчика i. Если счетчик вышел за размер массива, то закончить, иначе перейти на шаг 2.

Несложно заметить, что обработка матрицы и вектора по этому алгоритму будет не одинаковой, так как элементы в матрице адресуются двумя индексами. Для решения этой проблемы мы организуем ветвление, в котором по одной ветке будет вестись обработка вектора, а по другой – матрицы. Признаком вектора является наличие всего одного столбца или строки, так как мы помним, что адресация не зависит от представления вектора. Напомним, что для определения числа столбцов и строк используется функция size().

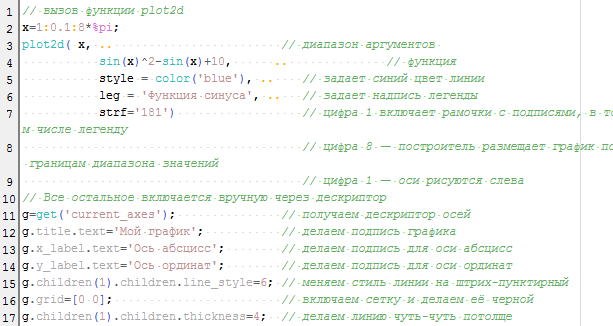
Вообще говоря, для решения этой задачи достаточно применения цикла for, но в качестве учебного примера использованы оба цикла: for и while. Итак, код такой функции будет выглядеть следующим образом:



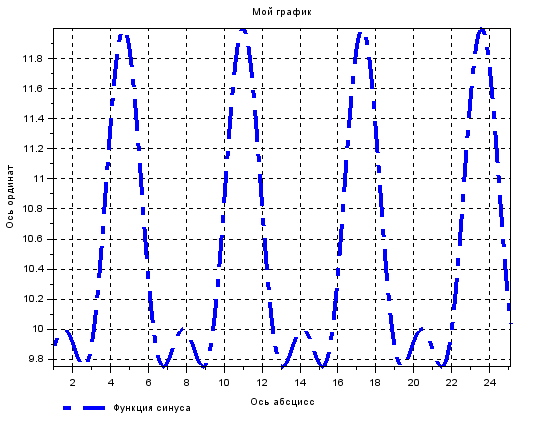
Результаты вызовов функции с различными данными:



Пакет Scilab также позволяет строить графики функций. В качестве примера приведём скрипт для построения двумерного графика функции :



Результат выполнения:



Окно графика является интерактивным, можно приближать и отдалять график, меняя его масштаб, при помощи колеса мыши.

**function** [y1,y2]=rootsOfBinomial(cf)

c=cf(1)

b=cf(2)

a=cf(3)

D=b^2-4\*a\*c

**if** D<0 then

disp("Уравнение не имеет действительных корней")

y1=%nan

y2=y1

**elseif** D==0 then

y1=-b/(2\*a)

y2=y1

**else**

y1=(-b+sqrt(D))/(2\*a)

y2=(-b-sqrt(D))/(2\*a)

**end**

**endfunction**

*// В этом примере выполняются действия первой ветви*

-->[ans ans]=rootsOfBinomial([1 2 3])

Уравнение не имеет действительных корней

*// Выполняются действия по ветви ''else''*

-->[ans ans]=rootsOfBinomial([6 3 -10])

*// Выполняются действия по ветви ''elseif''*

-->[ans ans]=rootsOfBinomial([1 2 1])

*// В этом примере выполняются действия первой ветви*

-->[ans ans]=rootsOfBinomial([1 2 3])

Уравнение не имеет действительных корней

ans =

Nan

ans =

Nan

*// Выполняются действия по ветви ''else''*

-->[ans ans]=rootsOfBinomial([6 3 -10])

ans =

0.9389867

ans =

- 0.6389867

*// Выполняются действия по ветви ''elseif''*

-->[ans ans]=rootsOfBinomial([1 2 1])

ans =

- 1.

ans =

- 1.

function [**Min**, **Max**]=explorer(**in**)

i=1 *// счетчик для циклов*

j=1 *// счетчик для цикла for*

**Max**=**in**(i) *// полагаем, что первый элемент массива и есть искомый максимум*

**Min**=**Max**

if or([size(**in**,'r')==1,size(**in**,'c')==1]) then

i=i+1

while i<=length(**in**)

if **in**(i)<**Min** then

**Min**=**in**(i)

end

if **in**(i)>**Max** then

**Max**=**in**(i)

end

i=i+1

end

else

for i=1:size(**in**,'r')

for j=1:size(**in**,'c')

if **in**(i,j)<**Min** then

**Min**=**in**(i,j)

end

if **in**(i,j)>**Max** then

**Max**=**in**(i,j)

end

end

end

end

endfunction

[a,b]=explorer([-5 6 8;-45 5 8;1 3 5])

-->x=1:0.1:4\*%pi;

-->plot2d( x, .. *// диапазон аргументов*

sin(x), .. *// функция*

style = color('blue'), .. *// задает синий цвет линии*

leg = 'Функция синуса', .. *// задает надпись легенды*

strf='181') *// цифра 1 включает рамочки с подписями, в том числе легенду*

*// цифра 8 — построитель размещает график по границам диапазона значений*

*// цифра 1 — оси рисуются слева*

*// Все остальное включается вручную через дескриптор*

-->g=**get**('current\_axes'); *// получаем дескриптор осей*

-->g.title.text='Мой график'; *// делаем подпись графика*

-->g.x\_label.text='Ось абсцисс'; *// делаем подпись для оси абсцисс*

-->g.y\_label.text='Ось ординат'; *// делаем подпись для оси ординат*

-->g.children(1).children.line\_style=6; *// меняем стиль линии на штрих-пунктирный*

-->g.grid=[0 0]; *// включаем сетку и делаем её черной*

-->g.children(1).children.thickness=4; *// делаем линию чуть-чуть потолще*

### Порядок выполнения работы

1. В командной строке произвести присвоение целой переменной. Над этой переменной выполнить действия: сложение, умножение, возведение в степень.
2. В командной строке задать две матрицы размером 3х3. Над этими матрицами произвести следующие действия: транспонирование, сложение, поэлементное сложение, умножение, возведение в степень.
3. В редакторе Scinotes набрать функцию, выполняющую поиск корней квадратного уравнения. Коэффициенты определяются вариантом задания.
4. В редакторе Scinotes набрать функцию поиска минимального и максимального значения в массиве данных. Продемонстрировать работу функции на векторе и квадратной матрице произвольного размера с любыми вещественными числами.
5. Построить график функции, соответствующей квадратному уравнению согласно своему варианту задания.
6. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:

* текст варианта задания на лабораторную работу;
* запись результатов действий согласно заданию (текст скриптов не включать).

### Варианты заданий

|  |  |  |  |
| --- | --- | --- | --- |
| **Вариант** | **Коэффициенты** | | |
|  |  |  |
| 1 | 10 | 9 | 2,4 |
| 2 | 12,4 | 4,2 | 5,7 |
| 3 | 10,4 | 6,4 | 4 |
| 4 | 2,4 | 8 | 9 |
| 5 | 5,7 | 12 | 4,2 |
| 6 | 4 | 19,2 | 6,4 |
| 7 | 9 | 12 | 8 |
| 8 | 4,2 | 15,2 | 12 |
| 9 | 6,4 | 16 | 19,2 |
| 10 | 8 | 10 | 12 |
| 11 | 12 | 12,4 | 15,2 |
| 12 | 19,2 | 10,4 | 16 |
| 13 | 12 | 2,4 | 10 |
| 14 | 15,2 | 5,7 | 12,4 |
| 15 | 16 | 4 | 10,4 |

### Контрольные вопросы

1. Что такое Scilab?
2. Для чего предназначен пакет Scilab?
3. Какие режимы выполнения расчётов существуют в Scilab, чем они отличаются?
4. Как задать переменную и матрицу в командной строке, какие действия с ними можно произвести?
5. Как создать скрипт в редакторе?
6. Как написать функцию в скрипте, как вызвать её в командной строке?
7. Какие конструкции в Scilab организуют ветвление, каков их синтаксис?
8. Какие конструкции в Scilab организуют циклы, каков их синтаксис?
9. Как построить двумерный график, какие существуют параметры у этой функции?

## Лабораторная работа №2. Описательная статистика

Цель работы: получение навыков расчета в среде Scilab параметров центральной тенденции и вариации.

### Теоретическая часть

Основным материалом при проведении исследований являются распределения переменных. Существуют способы численного описания этих переменных, наиболее часто используемыми являются измерение параметров центральной тенденции и вариации.

Параметры центральной тенденции: мода, медиана, среднее и взвешенное среднее.

Параметры вариации: размах, квартили, дисперсия, стандартное отклонение и коэффициент вариации.

Измерение центральной тенденции состоит в выборе одного числа, которое наилучшим образом описывает все значения признака из набора данных. Такое число называют центром набора данных, мерой центральной тенденции. Получив такое число, мы получаем информацию о распределении признака «в сжатой форме». При этом мы можем сравнивать при помощи этого числа два и более различных распределений.

**Мода** – наиболее часто встречающееся значение в выборке, наборе данных. В случае, если данные сгруппированы и построено распределение частот, модой является значение, имеющее наибольшую частоту.

Моду будем обозначать . Мода вполне пригодна для измерения центральной тенденции хотя бы потому, что это единственный способ описывать номинальное распределение не хуже порядкового или интервального. Ограничения в применении связаны с тем, что мода рассматривает лишь одну особенность распределения, а именно, расположение наиболее частого значения. Другие важные особенности, такие как число наблюдений выше или ниже моды, расстояние между модальными значениями и другие характеристики, остаются вне поля зрения.

Мода — значение во множестве наблюдений, которое встречается наиболее часто. (Мода = типичность.) Иногда в совокупности встречается более чем одна мода (например: 6, 2, 6, 6, 8, 9, 9, 9, 10; мода = 6 и 9). В этом случае можно сказать, что совокупность мультимодальна. Из структурных средних величин только мода обладает таким уникальным свойством. Как правило мультимодальность указывает на то, что набор данных не подчиняется нормальному распределению. Мода как средняя величина употребляется чаще для данных, имеющих нечисловую природу. Среди перечисленных цветов автомобилей (белый, черный, синий металлик, белый, синий металлик, белый) мода будет равна белому цвету.

**Медиана** определяется как серединное значение выборки, или значение, выше и ниже которого располагается одинаковое число наблюдений. Для нахождения медианы обязательно упорядочить данные. Медиана является точной серединой выборки. Обозначается  и определяется по-разному для выборок с четным и нечетным числом элементов. Для нечетного количества наблюдений медиана есть наблюдение с номером . Для четного количества наблюдений медиана вычисляется как среднее значение наблюдений с номерами  и . В случае нечетного количества наблюдений медиана есть просто середина выборки, выше и ниже которой располагается одинаковое количество наблюдений.

**Среднее** определяется как среднее арифметическое выборки, то есть как сумма всех значений выборки, деленная на ее объем. Следуя определению, будем находить среднее значение по формуле:



где  - сумма всех значений выборки,

 - объем выборки.

**Взвешенное среднее** – это среднее значение, получаемое при объединении нескольких групп наблюдений.

Если группы имеют одинаковый объем, то групповое среднее можно вычислить как среднее арифметическое имеющихся средних значений по каждой группе. Если же группы имеют различный объем, то групповое среднее можно найти по следующей формуле:

,

где  - сумма произведений средних в группе на количество элементов в этой группе, а  - общее число наблюдений.

Измерение вариации состоит в нахождении чисел, которые характеризуют степень разброса данных относительно центра распределения.

**Размах** – это разница между наибольшим и наименьшим значениями. Для нахождения размаха прежде рекомендуется упорядочить данные в порядке возрастания. Можно записать размах с помощью формулы:

.

**Квартили** – это значения, которые делят вариационный ряд на четыре равные по объему части. **Квартильный размах** (Inter Quartile Range - IRQ) – это разница между третьим и первым квартилями. Таких значений должно быть три: первая, вторая и третья квартиль соответственно. Для начала данные следует упорядочить. Затем отыскивается медиана, которая является вторым квартилем по определению. После этого находятся первый и третий квартили. Существует несколько вариантов формального определения квартилей.

Квартильный размах находится по формуле:



Если при вычислении размаха используются только наибольшее и наименьшее значения признака, а распределение данных между ними полностью игнорируется, то при вычислении квартильного размаха игнорируются «крайние» данные, расположенные за пределами первого и третьего квартилей. Между  и  расположено 50% всех данных.

**Дисперсия** для набора данных или выборки – это среднее арифметическое квадратов отклонений значений от их среднего.

Дисперсия обозначается . Основная формула (по определению) для нахождения дисперсии:



Формула означает, что нам следует вычитать среднее из каждого значения выборки, суммировать квадраты разности, а затем разделить полученную сумму на количество наблюдений минус 1.

**Стандартное отклонение** – квадратный корень из дисперсии выборки. Обозначается s и вычисляется по формуле:

.

**Коэффициент вариации** вычисляется как отношение стандартного отклонения к среднему значению выборки.

Формула для коэффициента вариации:

.

### Порядок выполнения работы

1. Создать скрипт для выполнения работы и сохранить его. В первой строке написать комментарий, содержащий название работы:



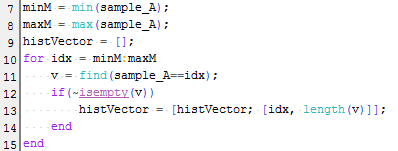
1. Сгенерировать последовательность случайных чисел размером 40:



1. Преобразовать выборку в целые числа, умножив каждый элемент на 100 и округлив до целого значения:



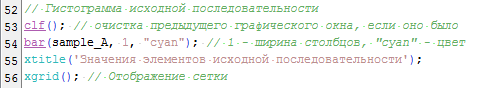
1. Найти моду распределения. для этого нужно определить, какие значения встречаются в последовательности, и сколько раз встречается каждое из этих значений:



1. Найти медиану и среднее распределения:



1. Сформировать четыре дополнительные последовательности произвольного объема от 30 до 60: sample\_В, sample\_С, sample\_D и sample\_E.
2. Вычислить среднее и размер получившихся последовательностей, воспользовавшись для этого функциями  и . Найти произведения этих величин для каждой последовательности (А-Е).
3. Найти общее число наблюдений, сложив размеры всех последовательностей.
4. Найти взвешенное среднее получившихся последовательностей.
5. Вычислить размах выборки А. Использовать функции нахождения максимального и минимального элементов  и .
6. Найти квартильный размах (функция ) выборки А.
7. Вычислить дисперсию (функция ), стандартное отклонение и коэффициент вариации выборки A.
8. Построить гистограмму исходной последовательности А:



1. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит запись результатов выполнения предыдущих пунктов.

### Варианты заданий

Варианты заданий в этой работе отсутствуют.

### Контрольные вопросы

1. Какие существуют способы численного описания распределений переменных, каковы их параметры?
2. В чем состоит измерение центральной тенденции?
3. Что такое мода, каковы ограничения её применения?
4. Приведите примеры нахождения моды.
5. Что такое медиана, как она вычисляется?
6. Что такое среднее и взвешенное среднее, как они вычисляются?
7. Размах и квартильный размах, формулы для их вычисления.
8. Что такое дисперсия, стандартное отклонение и коэффициент вариации, как они вычисляются?

## Лабораторная работа №3. Комбинаторика

Цель работы: получение навыков расчета в среде Scilab формул комбинаторики: перестановок, размещений, сочетаний и выбора с повторением.

### Теоретическая часть

Для вычисления вероятности иногда приходится использовать несколько важных формул из комбинаторики. Комбинаторика – это раздел математики, который содержит методы решения задач, связанных с перечислением и подсчетом. В комбинаторике имеются формулы для определения числа подмножеств заданного множества, подсчета числа перестановок, размещений и сочетаний.

**Принцип суммы.** Если некоторый объект  может быть выбран из совокупности объектов  способами, а другой объект  может быть выбран  способами, то выбрать либо , либо  можно  способами.

**Принцип произведения**. Если одно множество состоит из  различных элементов, другое из  различных элементов, и эти множества не пересекаются, то сколько различных пар можно образовать из элементов этих множеств, если первый элемент берется из первого множества, а второй – из второго? Согласно принципу произведения количество пар будет равно.

**Перестановки**. Сколькими способами  разных объектов могут быть расположены на одной линии?

Например, сколькими способами 6 человек могут сесть на шесть стульев? Чтобы подсчитать, можно размышлять следующим образом. Для первого существует 6 возможностей, для второго, после того как первый уже выбрал, останется всего 5, для следующего – 4 и так далее. Последний, шестой, после пятерых будет иметь только одну возможность. Итак, . Будем использовать обозначение 6! для записи таких произведений (произносится: шесть факториал). В общем виде количество перестановок из n элементов обозначается  и вычисляется по формуле:



**Размещения**. Сколькими способами из  разных объектов можно выбрать упорядоченное подмножество из  объектов? Упорядоченным считается множество, в котором задан порядок элементов. Объекты после выбора не возвращаются и повторно не могут быть выбраны. Например, сколькими способами из 6 человек можно выбрать четверых и рассадить на четыре стула? Способ подсчета аналогичен предыдущему. На первый стул сядет любой из шести, на следующий – уже из пяти. Всего четыре стула, поэтому: . В общем виде, количество возможных размещений из  элементов по m обозначается  и рассчитывается по формуле:



**Сочетания**. Сколькими способами из n разных объектов можно выбрать  объектов? Выбор не упорядочен. Объекты после выбора не возвращаются.

Например, сколькими способами из шестерых человек можно выбрать четверых? Несложные размышления приведут к тому, что следует модифицировать формулу для размещений, а именно, отказаться от упорядоченности выбранных элементов, что будет стоить нам  в знаменателе. Количество сочетаний для множества из  элементов по  элементов определяется по формуле:



**Выбор с повторением.** Сколькими способами из  разных объектов можно выбрать объектов с повторением? Объекты после выбора возвращаются.

Например, сколькими способами из шестерых человек можно выбрать четверых для дежурства, если можно выбирать с повторением и, потенциально, один из шестерых может быть выбран все четыре раза? В этом случае для каждого выбора у нас имеются все шесть кандидатов. Получим . В общем виде, количество способов для множества из  элементов по  элементов определяется по формуле: .

### Порядок выполнения работы

1. Создать скрипт для выполнения работы и сохранить его. В первой строке написать комментарий, содержащий название работы:



1. Сгенерировать два целых случайных числа  и , использовав функцию , в пределах согласно своему варианту. При этом должно соблюдаться условие .
2. Написать скрипт, рассчитывающий число перестановок, размещений, сочетаний и выборов с повторением. Для вычисления факториала используйте функцию .
3. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:

* текст варианта задания на лабораторную работу;
* формулы комбинаторики;
* результаты расчётов.

### Варианты заданий

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Минимальное и максимальное значения** | |
| 1 | 10 | 16 |
| 2 | 8 | 13 |
| 3 | 11 | 17 |
| 4 | 6 | 11 |
| 5 | 7 | 13 |
| 6 | 10 | 16 |
| 7 | 9 | 12 |
| 8 | 16 | 21 |
| 9 | 7 | 11 |
| 10 | 14 | 20 |
| 11 | 11 | 15 |
| 12 | 14 | 19 |
| 13 | 15 | 22 |
| 14 | 7 | 12 |
| 15 | 11 | 16 |

### Контрольные вопросы

1. Что такое комбинаторика?
2. В чём состоят принципы суммы и произведения?
3. Как решаются задачи при помощи формулы перестановок?
4. Как решаются задачи при помощи формулы размещений?
5. Как решаются задачи при помощи формулы сочетаний?
6. Как решаются задачи при помощи формулы выбора с повторением?

## Лабораторная работа №4. Параметры генеральной совокупности и выборки

Цель работы: определить параметры выборки, пригодные для оценки генеральной совокупности.

### Теоретическая часть

Характеристики генеральной совокупности называются параметрами. Параметр генеральной совокупности есть фиксированное число, которое нам не известно, при его вычислении случайность отсутствует. Тем самым, параметр есть неизвестная и фиксированная величина.

С другой стороны, статистикой мы назвали числовую характеристику выборки. Статистика является случайной величиной, так как в ее основе лежат данные, полученные в результате случайного отбора. Тем самым, статистика является известной и случайной величиной. Статистики являются оценочными функциями параметров генеральной совокупности. Фактическое значение статистики, рассчитанное по данным выборки, назовем оценкой параметра генеральной совокупности.

**Среднее** определяется как среднее арифметическое выборки, то есть как сумма всех значений выборки, деленная на ее объем. Следуя определению, будем находить среднее значение по формуле:



где  - сумма всех значений выборки,

 - объем выборки.

**Медиана** является точной серединой заранее упорядоченной выборки. Обозначается  и определяется по-разному для выборок с четным и нечетным числом элементов. Для нечетного количества наблюдений медиана есть наблюдение с номером . Для четного количества наблюдений медиана вычисляется как среднее значение наблюдений с номерами  и .

**Размах** – это разница между наибольшим и наименьшим значениями. Для нахождения размаха прежде рекомендуется упорядочить данные в порядке возрастания. Можно записать размах с помощью формулы:

.

**Дисперсия** для набора данных или выборки – это среднее арифметическое квадратов отклонений значений от их среднего. Дисперсия обозначается . Основная формула (по определению) для нахождения дисперсии:



**Стандартное отклонение** – квадратный корень из дисперсии выборки. Обозначается s и вычисляется по формуле:

.

**Доля** – это отношение некоторого подмножества частот к общей сумме частот:

, где

 – одна из частот в распределении,

 – общее число наблюдений.

Рассмотрим пример. Предположим, мы имеем генеральную совокупность, состоящую из чисел 1, 2 и 5.

Параметры такой генеральной совокупности:

Среднее значение



Медиана (число элементов – нечётное)



Размах



Генеральная дисперсия



Стандартное отклонение = 1,7



Доля нечетных чисел = 0,67

.

Теперь найдём все возможные выборки по два элемента из генеральной совокупности (таблица 1). Их число (выбор с повторениями) составит:

.

В этом учебном примере генеральная совокупность известна полностью. В реальности так почти никогда не случается. Как правило, невозможно, например, опросить всех жителей страны по некоторому вопросу. Поэтому из всей генеральной совокупности делаются выборки приемлемых размеров и оцениваются их параметры, по которым и делают статистические выводы о параметрах генеральной совокупности в целом. Не все параметры выборок совпадают с параметрами генеральной совокупности.

Поэтому найдём указанные выше параметры для каждой выборки, а затем посчитаем средние значения этих параметров для всех выборок. Совпадение этих средних значений с соответствующими параметрами генеральной совокупности позволит нам сделать вывод о том, какие параметры выборки могут характеризовать всю генеральную совокупность. Выполнение этих расчётов вручную – трудоёмкая задача, поэтому используем среду Scilab. Результаты занесём в таблицу 1.

Дисперсия и стандартное отклонение каждой из девяти выборок вычисляется по формулам:

, .

Например, для третьей выборки:

, .

Из таблицы видно, что совпадение среднего значения статистики со значением параметра генеральной совокупности происходит только для трех характеристик: среднего, дисперсии и доли. Эти три статистики могут служить оценками параметров генеральной совокупности. Три другие: медиана, размах и стандартное отклонение, как показал расчет, не могут служить оценками соответствующих параметров генеральной совокупности.

Таблица 1 Параметры генеральной совокупности и выборки

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Все возможные выборки объема n=2 | | Среднее | Медиана | Размах | Дисперсия | Стандартное отклонение | Доля нечётных чисел |
| 1 | 1 | 1,0 | 1,0 | 0 | 0,0 | 0,0 | 1,0 |
| 1 | 2 | 1,5 | 1,5 | 1 | 0,5 | 0,7 | 0,5 |
| 1 | 5 | 3,0 | 3,0 | 4 | 8,0 | 2,8 | 1,0 |
| 2 | 1 | 1,5 | 1,5 | 1 | 0,5 | 0,7 | 0,5 |
| 2 | 2 | 2,0 | 2,0 | 0 | 0,0 | 0,0 | 0,0 |
| 2 | 5 | 3,5 | 3,5 | 3 | 4,5 | 2,1 | 0,5 |
| 5 | 1 | 3,0 | 3,0 | 4 | 8,0 | 2,8 | 1,0 |
| 5 | 2 | 3,5 | 3,5 | 3 | 4,5 | 2,1 | 0,5 |
| 5 | 5 | 5,0 | 5,0 | 0 | 0,0 | 0,0 | 1,0 |
| Среднее значение статистики | | **2,7** | 2,67 | 1,78 | **2,9** | 1,3 | **0,7** |
| Параметр генеральной совокупности | | **2,7** | 2 | 4 | **2,9** | 1,7 | **0,7** |

### Порядок выполнения работы

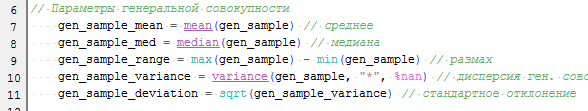
1. Рассчитать параметры генеральной совокупности согласно своему варианту по соответствующим формулам.
2. Создать скрипт для выполнения работы и сохранить его. В первой строке написать комментарий, содержащий название работы.



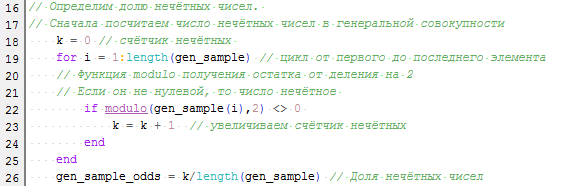
1. Задать генеральную совокупность согласно своему варианту в виде матрицы.



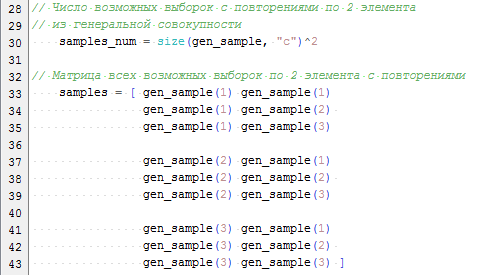
1. Рассчитать параметры генеральной совокупности, используя функции Scilab. Проверить совпадение с результатами расчёта в п.1.



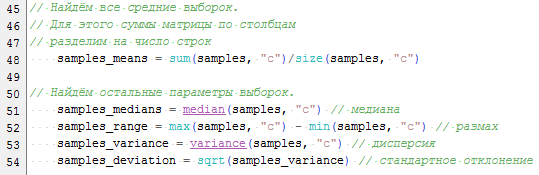
1. Определить долю нечётных чисел генеральной совокупности, проверить результат расчёта в п.1



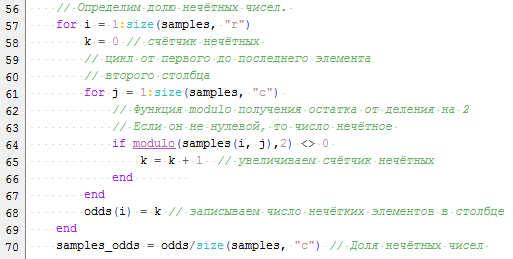
1. Определить число возможных выборок из генеральной совокупности и составить все эти выборки:



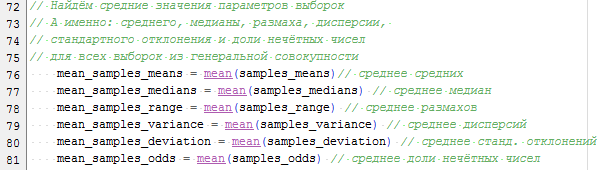
1. Найти параметры всех выборок:



1. Найти доли нечётных чисел всех выборок:



1. Найти средние значения всех параметров всех выборок:



1. Организовать вывод данных одним из способов: в командной строке основного окна Scilab, при помощи функции  или функции *printf*. Указанные функции изучить самостоятельно.
2. Оформить отчёт по лабораторной работе. Раздел «Выполнение работы» содержит:

* текст варианта задания на лабораторную работу;
* Полный расчёт параметров генеральной совокупности;
* таблицу с результатами расчётов в среде Scilab (таблица 1).

### Варианты заданий

|  |  |
| --- | --- |
| Вариант | Генеральная совокупность |
| 1 | 1 6 7 |
| 2 | 2 4 8 |
| 3 | 11 12 16 |
| 4 | 12 13 17 |
| 5 | 7 10 13 |
| 6 | 9 15 17 |
| 7 | 6 8 11 |
| 8 | 8 12 14 |
| 9 | 9 14 16 |
| 10 | 7 9 12 |
| 11 | 4 8 9 |
| 12 | 5 7 11 |
| 13 | 9 13 15 |
| 14 | 13 16 18 |
| 15 | 9 10 13 |

### Контрольные вопросы

1. Что такое параметры генеральной совокупности?
2. Что такое статистики выборки, как они соотносятся с генеральной совокупностью?
3. Что такое среднее?
4. Что такое медиана?
5. Что такое размах?
6. Что такое дисперсия?
7. Что такое стандартное отклонение?
8. Что такое доля?
9. Какие параметры выборок совпадают с параметрами генеральной совокупности?

## Лабораторная работа №5.Построение нечётких множеств и операции над ними

Цель работы: освоить основные операции, выполняемые над нечёткими множествами.

### Теоретическая часть

Пусть  – произвольное непустое множество элементов . Тогда нечетким множеством  множества  называется совокупность пар:

, , ,

Где  – функция принадлежности нечеткого множества , которая может принимать значения на некотором множестве . Как правило,

.

Значение  показывает, в какой мере элемент  принадлежит множеству .

Пусть заданы нечеткие множества:

 и , , .

Тогда для них справедливы отношения и операции:

**Равенство**:

, когда 

**Включение**:

, то есть включено в .

**Объединение (нечёткое ИЛИ)**:

Для нечетких множеств:



**Пересечение (нечёткое И)**



**Дополнение**

Для нечетких множеств:



**Алгебраическое произведение**



**Алгебраическая сумма**



**Алгебраическая разность**



**Концентрация**



**Растяжение**

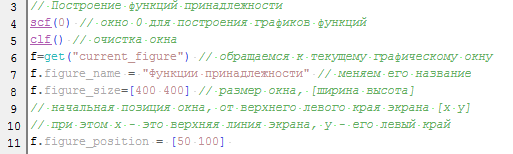
.

### Порядок выполнения работы

1. Создать скрипт для выполнения работы и сохранить его. В первой строке написать комментарий, содержащий название работы.



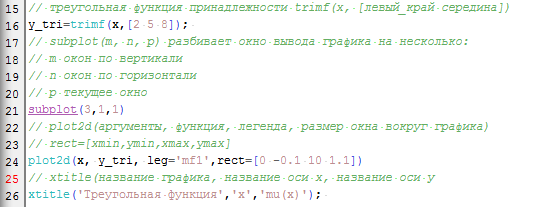
1. Создать окно для построения функций принадлежности:



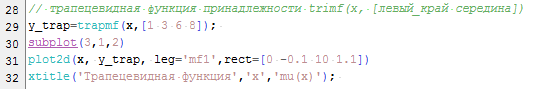
1. Задать набор значений аргумента множества:



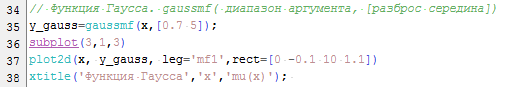
1. Построить в графическом окне нечёткое множество треугольной формы:



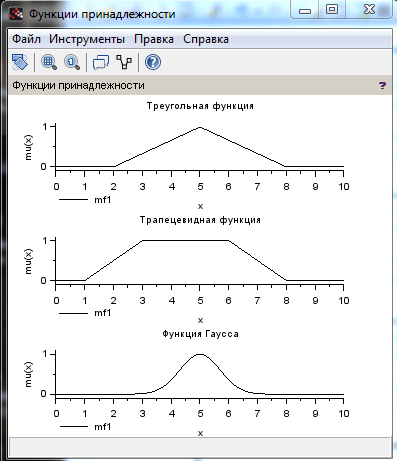
1. Построить в графическом окне нечёткое множество трапецевидной формы:



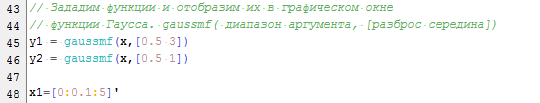
1. Построить в графическом окне нечёткое множество функции Гаусса:



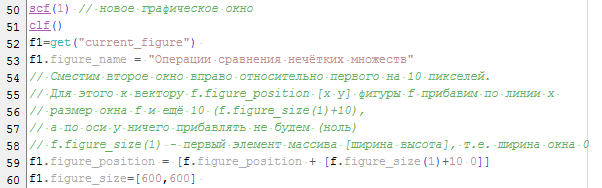
1. Проверить работу программы:



1. Задать две функции Гаусса  и  для проведения логических операций и диапазон изменения их аргументов:



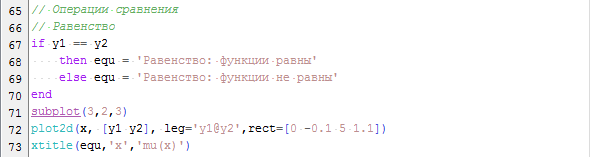
1. Создать второе окно для отображения результатов операций:



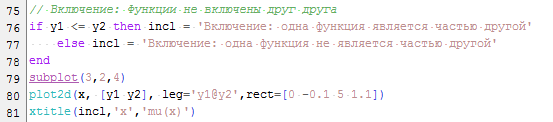
1. Отобразить в нём нечёткие множества (функции принадлежности)  и :



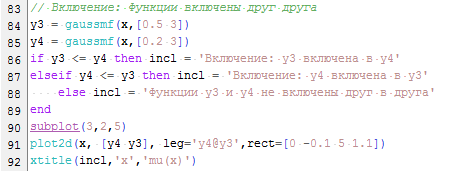
1. Отобразить в том же окне результат операции равенства функций  и :



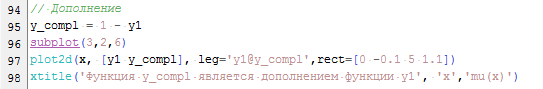
1. Отобразить в том же окне результат операции включения функций  и :



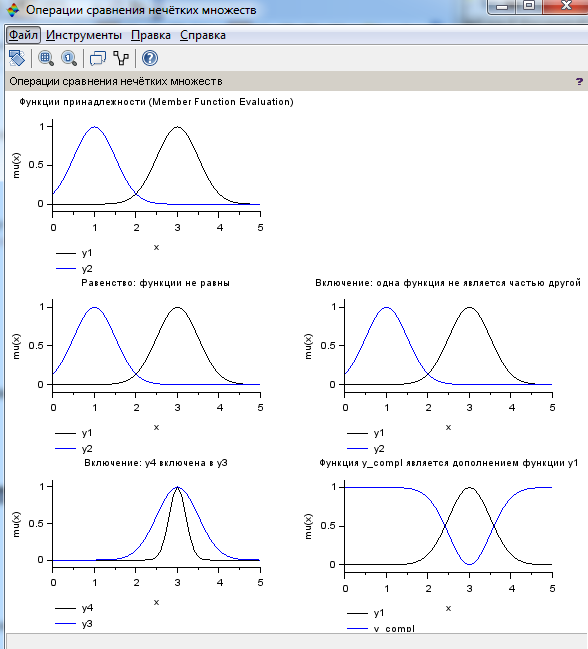
1. Задать две функции Гаусса  и  для проведения операции включения, провести над ними операцию, результат отобразить в том же графическом окне:



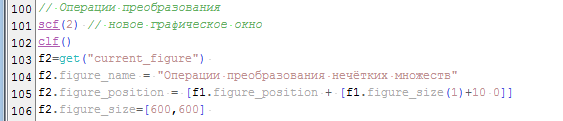
1. Провести над функцией  операцию дополнения, вывести результат в том же графическом окне:



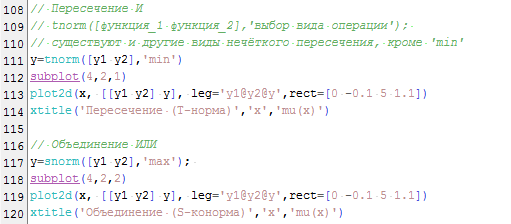
1. Проверить работу программы:



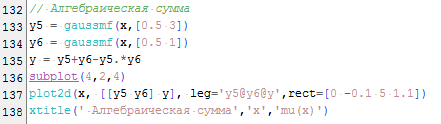
1. Создать новое графическое окно для отображения результатов операций преобразования:



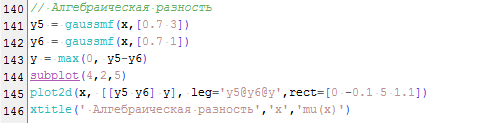
1. Провести над функциями  и  операции пересечения и объединения:



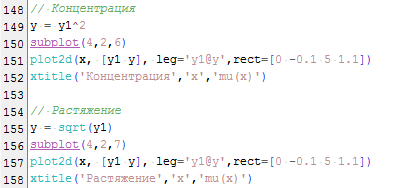
1. Задать функции Гаусса  и , провести над ними операцию алгебраической суммы:



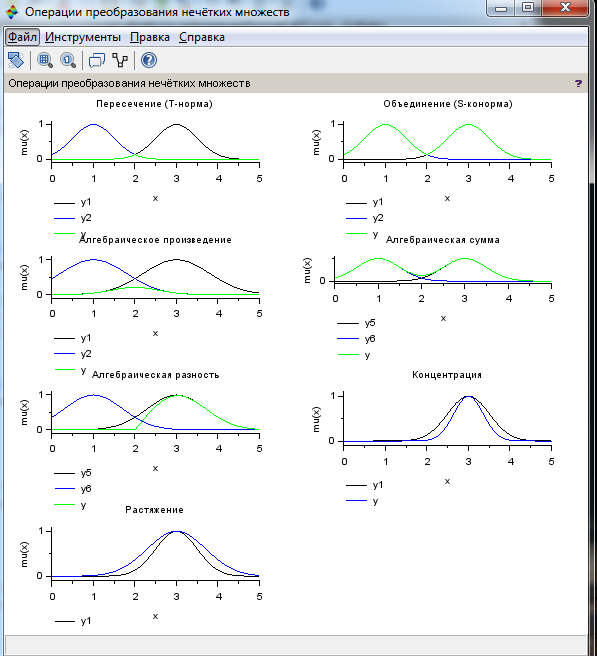
1. Переопределить функции Гаусса  и , провести над ними операцию алгебраической разности:



1. Провести над функцией  операции концентрации и растяжения:



1. Проверить работу программы:



### Варианты заданий

Варианты заданий в данной работе отсутствуют.

### Контрольные вопросы

1. Что такое нечёткое множество?
2. Дайте определение операций равенства и включения.
3. Дайте определение операций объединения, пересечения и дополнения.
4. Дайте определение операций алгебраических произведения, суммы и разности.
5. Дайте определение операций концентрации и растяжения.

## Лабораторная работа №6.Вычисление нечётких выражений

Цель работы: освоить основные операции, выполняемые над нечёткими переменными.

### Теоретическая часть

Для смыслового различения нечетких множеств существует понятие нечеткой переменной.

Нечеткой переменной называют тройку

, в которой

 – имя переменной,

 – область определения переменной,

 – нечеткой подмножество  переменных , описывающее ограничения на возможные значения переменной . Это подмножество определяет правило выбора из  тех , которые соответствуют понятию .

Значениями нечетких переменных являются числа, т.е. область определения – числовая. В том случае, когда значениями нечеткой переменной являются не числа, а слова или предложения на естественном или формальном языке, эта нечеткая переменная называется лингвистической переменной.

Нечеткое высказывание – это предложение, построенное по логическим или синтаксическим правилам из нечетких и лингвистических переменных.

Операции над нечеткими высказываниями заключаются в следующем.

1. Определить степень истинности нечеткого высказывания.
2. Сравнить, насколько смысл одного высказывания совпадает со смыслом других.

На основании одних нечетких высказываний получить другие нечеткие высказывания, вытекающие из первых (нечеткий вывод).

Все эти операции представляют собой попытку моделирования человеческих рассуждений в процессе принятия решений.

Рассмотрим **операции** над нечеткими логическими высказываниями.

Пусть  и  –нечеткие высказывания со степенями истинности:





Тогда справедливы операции:

**Отрицание**



Для множества:





**Конъюнкция**

Степень истинности конъюнкции равна степени истинности менее истинного высказывания:



**Дизъюнкция**



**Исключающее или**



**Импликация**



Степень истинности  не меньше степени истинности  и тем выше, чем выше степень истинности .

**Эквивалентность**

Эквивалентность высказываний  и  есть минимум импликаций.

Высказывания  и  эквивалентны



Если , то высказывания обычно считают эквивалентными.

Пусть  и  – нечеткие переменные, соответствующие нечетким высказываниям  и .

Используются следующие меры сравнения заданных высказываний.

**Степень включения**

Когда говорят, что одна нечеткая переменная включена в другую, то под этим понимают, что .





Если включения  в  неполное, то говорят о степени включения  в .





**Степень равенства**



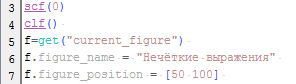


### Порядок выполнения работы

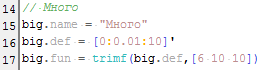
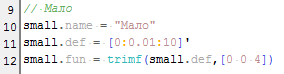
1. Создать скрипт для выполнения работы и сохранить его. В первой строке написать комментарий, содержащий название работы.



1. Создать окно для построения результатов работы



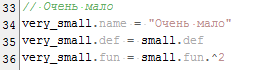
1. Создать нечёткие переменные «Мало» и «Много»



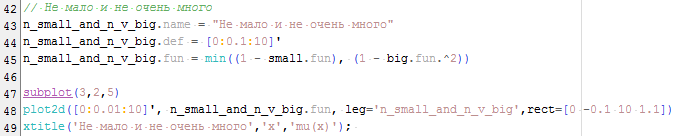
1. Построить их в графическом окне



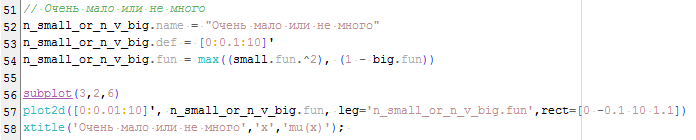
1. Вычислить и построить выражение «Очень мало»



1. Вычислить и построить выражение «Не мало и не очень много»



1. Вычислить и построить выражение «Очень мало или немного»



1. Вычислить и построить два выражения согласно своему варианту.

### Варианты заданий

|  |  |  |
| --- | --- | --- |
| № | Тип функции | Операции |
| 1 | Треугольник  Трапеция | ИЛИ, НЕ, Импликация |
| 2 | Треугольник  Треугольник | НЕ, Очень, ИЛИ |
| 3 | Трапеция  Трапеция | ИЛИ, И, Импликация |
| 4 | Треугольник  Гаусса | НЕ, Очень, И |
| 5 | Трапеция  Трапеция | ИЛИ, И, НЕ |
| 6 | Гаусса  Гаусса | НЕ, Импликация, Очень |
| 7 | Треугольник  Трапеция | ИЛИ, И., НЕ |
| 8 | Гаусса  Треугольник | НЕ, Очень, И |
| 9 | Трапеция  Трапеция | ИЛИ, НЕ, Импликация |
| 10 | Треугольник  Гаусса | НЕ, ИЛИ, Очень |
| 11 | Трапеция  Треугольник | ИЛИ, НЕ, И |
| 12 | Гаусса  Гаусса | Очень, ИЛИ, И |
| 13 | Треугольник  Гаусса | НЕ, Импликация, И |
| 14 | Трапеция  Гаусса | Очень, ИЛИ, И |
| 15 | Треугольник  Гаусса | НЕ, ИЛИ, Очень |

### Контрольные вопросы

1. Что такое нечёткая переменная?
2. Что является значениями нечётких переменных?
3. Что такое нечёткое высказывание, какие операции можно производить с ними?
4. Что такое отрицание, конъюнкция и дизъюнкция?
5. Что такое исключающее или, импликация и эквивалентность?
6. Что такое степень включения и степень равенства?

## Лабораторная работа №7. Расчёт параметров выборки данных

Цель работы: научиться определять основные параметры выборки.

### Теоретическая часть

В данной работе будет произведено упорядочивание выборки чисел, вычислены частоты наблюдений и построено их интервальное распределение.

**Переменная** (variable) – это общая для всех изучаемых объектов характеристика или свойство, проявления которого могут меняться у различных объектов. Эти проявления называют значениями, альтернативами и градациями. Предметом исследования являются переменные и их значения, которые могут быть распределены на некоторой области.

**Распределение переменной** – это совокупность различных значений, которые принимает переменная для различных изучаемых объектов.

**Генеральная совокупность** – это вся интересующая исследователя совокупность изучаемых объектов.

**Выборка** – это некоторая часть генеральной совокупности, отбираемая специальным образом и исследуемая с целью получения выводов о свойствах генеральной совокупности.

Результаты некоторых измерений могут быть записаны в виде последовательности чисел, которые являются статистиками некоторого объекта. Эти данные могут быть упорядоченными и неупорядоченными. **Упорядоченные** данные содержат результаты наблюдений, записанные по возрастанию или убыванию. **Неупорядоченные** данные состоят из результатов, записанных в произвольном порядке.

**Частота** – это количество наблюдений, в которых признак принимает определённое значение или находится в определённом интервале. Распределение частот показывает частоты во взаимосвязи с результатами наблюдений.

Рассмотрим пример.

Предположим, что наблюдения веса 24 респондентов находятся в интервале от 46 до 70 (таблица 1).

Таблица 1 Вес тела (упорядоченные данные)

|  |  |  |  |
| --- | --- | --- | --- |
| 46 | 59 | 65 | 69 |
| 49 | 60 | 65 | 69 |
| 50 | 60 | 65 | 69 |
| 50 | 60 | 66 | 70 |
| 52 | 61 | 67 | 70 |
| 53 | 62 | 67 | 70 |

Недостатком такого представления является его слабая пригодность для последующего изучения выборки. Построим другое, более удобное представление. Все наблюдения находятся в интервале от наименьшего 46 до наибольшего 70.

**Интервальное** распределение частот состоит из некоторого количества интервалов равной длины, на которые делится весь диапазон изменения признака, и соответствующих этим интервалам частот. При таком преобразовании утрачивается часть информации, поскольку наблюдения перестают быть точными. Но этими данными можно пожертвовать для получения знания о виде распределения.

Сгруппируем наблюдения (таблица 1) и построим таблицу интервального распределения частот (таблица 2).

Таблица 2 Интервальное распределение частот

|  |  |
| --- | --- |
| Интервалы |  |
| 45-49 | 2 |
| 50-54 | 4 |
| 55-59 | 1 |
| 60-64 | 5 |
| 65-69 | 9 |
| 70-74 | 3 |
| **Итого** | **24** |

**Интервальное** распределение частот состоит из некоторого количества интервалов равной длины, на которые делится весь диапазон изменения признака, и соответствующих этим интервалам частот.

При построении интервального распределения нужно соблюдать несколько условий.

1. Интервалы не должны пересекаться.
2. Все возможные значения измеряемого признака должны быть охвачены интервалами.
3. Интервалы должны иметь одинаковую длину. Но если в крайние интервалы попадает небольшое количество значений признака, то они могут быть объединены с соседними. В этом случае они будут иметь большую длину. Например, «менее 5», «5-10» и «более 10».
4. Интервалы не должны иметь пробелов. Те из них, в которые не попало ни одного значения, тоже включаются в рассмотрение. Исключение составляют первый и последний интервалы, их можно исключить.
5. Принято, что середины интервалов являются целыми числами. Кроме того, рекомендуется использовать от 5 до 20 интервалов.

Рассмотрим пример. Имеются данные о продолжительности 25 телефонных разговорах в минутах. Нужно получить распределение частот, используя 8 интервалов.

Таблица 3 Продолжительность телефонных разговоров

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 12 | 19 | 16 | 12 | 33 |
| 31 | 10 | 17 | 22 | 35 |
| 7 | 25 | 14 | 21 | 25 |
| 13 | 24 | 9 | 35 | 37 |
| 23 | 15 | 21 | 19 | 18 |

Построим интервальное распределение частот.

Найдем самое большое и самое маленькое значения: 37 и 7.

Затем определим диапазон значений выборки, вычтя наименьшее значение из наибольшего: 37 - 7 = 30. Теперь определим количество интервалов: задано 8. Найдем длину интервала, разделив диапазон данных на количество интервалов и округлив до ближайшего целого: .

Нижней границей первого интервала является наименьшее значение 7. Прибавим 4 и получим нижнюю границу второго интервала. Продолжим прибавлять до тех пор, пока не получим 8 значений: 7, 11, 15, 19, 23, 27, 31, 35 – нижние границы. Теперь вычтем единицу из нижних пределов, чтобы получить верхние пределы. Первый интервал 7-10, второй 11-13 и так далее.

Найдем точные границы интервалов, вычитая 0,5 от каждого нижнего предела и прибавляя 0,5 к верхнему пределу интервала. Подсчитаем количество данных, попадающих в каждый из интервалов, и запишем итоговые численные значения в столбце для частот.

Таблица 4 Вычисление интервального распределения частот

|  |  |  |
| --- | --- | --- |
| **Интервал** | **Границы** | **Частота** |
| 7-10 | 6,5-10,5 | 3 |
| 11-14 | 10,5-14,5 | 4 |
| 15-18 | 14,5-18,5 | 4 |
| 19-22 | 18,5-22,5 | 5 |
| 23-26 | 22,5-26,5 | 4 |
| 27-30 | 26,5-30,5 | 0 |
| 31-34 | 30,5-34,5 | 2 |
| 35-38 | 34,5-38,5 | 3 |

Рассмотрим доли частот. **Доля** – это отношение некоторого подмножества частот к общей сумме частот:

, где

 – одна из частот в распределении,

 – общее число наблюдений.

Вычисление долей частот позволяет приводить различные измерения к стандартному распределению значений от  до . Это позволяет сравнивать измерения, произведённые на выборках разного объема.

**Проценты** это доля, умноженная на 100. Учитывая формулу доли, получим:

.

Преобразование долей частот в проценты переводит первоначальные частоты в другую линейную шкалу от  до .

**Гистограмма частот** – это графическое представление, в котором по оси Х откладываются значения переменной, а по оси Y соответствующие им частоты. Гистограмма строится в виде прямоугольников, высота которых соответствует частоте значения переменной.

Таблица 5 Варианты задания

|  |  |
| --- | --- |
| **№ варианта** | **Выборка** |
| 1 | [2, 2, 1, 5, 9, 1, 10, 5, 9, 2, 1, 10, 2, 8, 1, 7, 4, 4, 5, 8] |
|  |  |
| 2 | [5, 8, 4, 11, 11, 10, 4, 2, 8, 3, 5, 2, 11, 3, 2, 7, 9, 9, 11, 10] |
|  |  |
| 3 | [10, 1, 4, 10, 9, 5, 11, 9, 11, 5, 11, 9, 7, 3, 5, 7, 2, 8, 7, 6] |
|  |  |
| 4 | [1, 6, 5, 7, 8, 1, 8, 5, 11, 7, 8, 8, 10, 10, 10, 5, 9, 4, 6, 2] |
|  |  |
| 5 | [5, 8, 4, 11, 11, 10, 4, 2, 8, 3, 5, 2, 11, 3, 2, 7, 9, 9, 11, 10] |
|  |  |
| 6 | [1, 6, 8, 9, 9, 5, 6, 10, 8, 7, 1, 2, 3, 2, 6, 5, 7, 2, 6, 9] |
|  |  |
| 7 | [7, 6, 7, 6, 11, 2, 4, 2, 2, 10, 11, 4, 2, 6, 10, 11, 2, 1, 9, 7] |
|  |  |
| 8 | [9, 4, 9, 8, 8, 8, 9, 11, 9, 11, 7, 7, 5, 3, 3, 10, 1, 4, 6, 9] |
|  |  |
| 9 | [10, 8, 9, 7, 9, 4, 3, 10, 11, 9, 2, 3, 9, 4, 6, 8, 11, 3, 5, 11] |
|  |  |
| 10 | [1, 7, 3, 5, 10, 3, 6, 11, 1, 2, 9, 2, 7, 7, 9, 8, 8, 4, 2, 11] |
|  |  |
| 11 | [11, 5, 4, 6, 10, 11, 10, 7, 8, 10, 7, 9, 2, 9, 3, 3, 11, 9, 11, 5] |
|  |  |
| 12 | [5, 4, 6, 1, 5, 9, 6, 5, 1, 8, 11, 3, 5, 6, 4, 10, 6, 10, 3, 6] |
|  |  |
| 13 | [4, 3, 3, 5, 6, 7, 2, 2, 9, 3, 1, 2, 2, 11, 11, 11, 5, 10, 4, 8] |
|  |  |
| 14 | [2, 10, 3, 11, 8, 4, 10, 4, 2, 3, 3, 5, 7, 3, 10, 1, 2, 5, 9, 11] |
|  |  |
| 15 | [8, 10, 7, 11, 3, 11, 2, 5, 8, 6, 11, 9, 11, 3, 11, 2, 5, 4, 9, 4] |

### Порядок выполнения работы

1. Ознакомиться с выборкой, занести её в отчёт в табл. 5 первой строкой.
2. Упорядочить выборку, занести её в табл. 5 второй строкой.
3. Подсчитать частоты значений выборки. Результаты занести в табл. 6

Таблица 6 Частоты значений выборки

|  |  |
| --- | --- |
| Значение | Частота |
|  |  |
|  |  |
|  |  |

1. Найти минимум и максимум и диапазон значений выборки.
2. Построить интервальное распределение частот для пяти интервалов: найти интервалы, их точные границы и частоты значений в выборке. Результаты занести в табл. 7.

Таблица 7 Интервальное распределение частот

|  |  |  |
| --- | --- | --- |
| Интервал | Границы | Частота |
|  |  |  |
|  |  |  |

1. Используя данные табл. 7, построить гистограмму частот.
2. Найти процент и долю чётных значений в выборке.
3. Оформить отчёт, который должен содержать:
   1. Неупорядоченную выборку (таблица 5).
   2. Упорядоченную выборку (таблица 5).
   3. Частоты значений выборки (таблица 6)
   4. Интервальное распределение частот (таблица 7)
   5. Минимум, максимум, диапазон, число интервалов, длину интервалов, процент и долю чётных значений выборки.
   6. Гистограмму частот.
   7. Коробковую диаграмму. На ней обозначить медиану, первый и третий квартили, минимальное и максимальное значения.

## Лабораторная работа №8. Использование Python для обработки данных

Цель работы: при помощи языка Python создать большую выборку случайных данных и записать её в файл.

### Теоретическая часть

Кроме Scilab, существует много других инструментов, применяемых в промышленной и научной разработке. Одним из наиболее часто используемых является язык Python (Пайтон). Это высокоуровневый интерпретируемый язык программирования общего назначения. Он позволяет создавать консольные и графические приложения, веб-сайты и серверы, мобильные приложения.

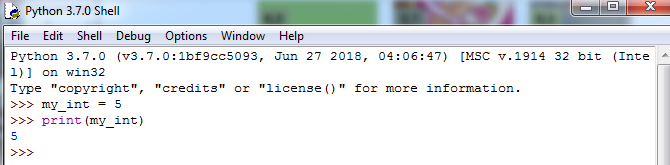
Python – активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Он поддерживается в большинстве существующих операционных систем. Этот язык программирования распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. Скачать интерпретатор Python 3.7.2 для установки можно по адресу <https://www.python.org/>.

Дистрибутив языка включает в себя среду разработки программ Integrated Development Environment (IDLE):

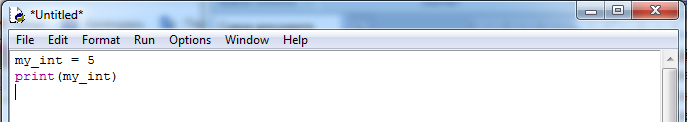
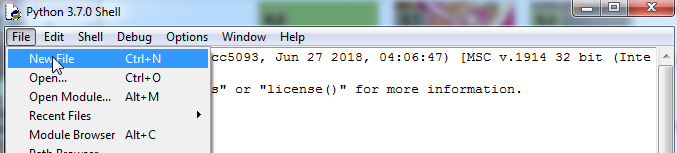
**Пуск\Все\_программы\Python\_3.7\IDLE**

Она будет использоваться в данной работе. Файлы, содержащие Python код, имеют расширение .py и называются модулями.

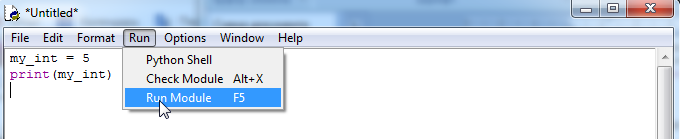
Код можно набирать в командной строке IDLE, например



Так можно проводить несложные расчёты. Для многострочных модулей удобнее пользоваться встроенным редактором



Модуль, написанный в редакторе, сохраняется и запускается из меню или при помощи клавиши F5



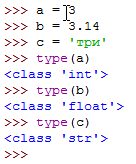
При этом модуль должен быть сохранён. Результат выполнения появится в командной строке IDLE.



В языке Python используются числа, переменные, операторы, модули и функции. Рассмотрим те из них, которые понадобятся в данной работе.

Одной из синтаксических особенностей языка является выделение блоков кода с помощью отступов (пробелов или табуляций), поэтому в Python отсутствуют операторные скобки **begin/end**, как в языке Паскаль, или фигурные скобки, как в Си. Число отступов кратно четырём: первый уровень – четыре пробела, второй уровень вложенности – восемь и т.д. Это позволяет сократить количество строк и символов в программе и делает код удобным для чтения. С другой стороны, поведение и даже корректность программы может зависеть от начальных пробелов в тексте.

В Python используются целый тип (**int**), вещественный (**float**), булевский (**boolean**) и другие. Тип переменной определятся в момент присвоения ей значения (динамическая типизация). Функция **type()** возвращает тип переменной или значения, переданного ей.

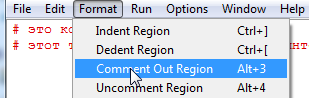


В Python используются различные операторы. В лабораторных работах будут использованы арифметические (сложение, вычитание, деление), условный оператор, а также операторы цикла и ветвления.

Комментарии в Python обозначаются знаком #



Если нужно закомментировать значительный блок кода, то можно выделить этот блок и воспользоваться сочетанием Alt+F3 или пунктом меню IDLE.



Снять комментарии можно выделив блок кода и применив комбинацию Alt+F4.

Большие программы, как правило, состоят из стартового файла верхнего уровня, и набора файлов-модулей. Модулем в Python называется любой файл с программой (все программы, которые вы писали, можно назвать модулями). Множество стандартных функций в Python содержаться в модулях. Каждая программа может импортировать модуль и получить доступ к его классам, функциям и объектам. Нужно заметить, что модуль может быть написан не только на Python, а например, на C или C++. Подключить модуль можно с помощью команды **import**.

Например, модуль **random** содержит функцию **randint()**, которая генерирует случайное число, в примере ниже – от 1 до 10.

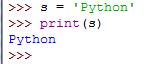


Также можно импортировать из модуля только одну или несколько функций.



Функция **print()** Python выводит заданные объекты на стандартное устройство вывода (экран) или отправляет их текстовым потоком в файл.

**print(список объектов)**



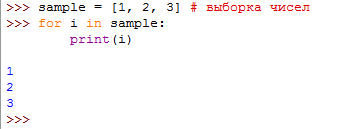
В данной работе будет использован цикл **for**

**for <элемент> in <итерируемый объект>:**

**выражение 1**

**выражение 2**

Например:



Конструкция **with ... as** используется для оборачивания выполнения блока инструкций менеджером контекста. Синтаксис конструкции **with ... as:**

**with** expression [**as** target] , expression **as** target**:**

**выражение 1**

**выражение 2**

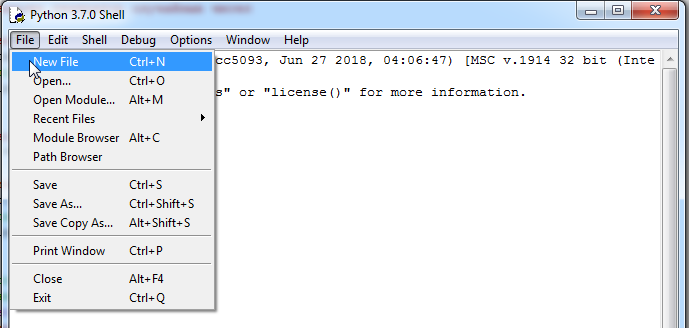
Эта конструкция часто применяется для записи данных в файл и чтения из него.

В процессе выполнения лабораторной работы будет обрабатываться случайно сгенерированная выборка данных. Хранить её будем в файле формата .csv (Comma Separated Values).

Формат CSV является наиболее часто используемым форматом импорта и экспорта для баз данных и таблиц. Файл CSV (значения, разделенные запятыми) позволяет сохранять данные в табличной структуре с расширением **.csv**. Такие файлы считаются очень простыми в обработке, поэтому используются в различных приложениях. Для работы с ними в Python создан специальный модуль, который нужно импортировать командой **import csv**. Этот модуль имеет несколько функций и классов, доступных для чтения и записывая файлов, в том числе **csv.reader csv.writer**.

### Порядок выполнения работы

1. Запустить среду создания программ IDLE (Пуск\Все\_программы\Python\_3.7\IDLE)
2. Создать файл



1. Сохранить файл с именем **lab8\_Ivanov** (номер лабораторной работы плюс фамилия студента)
2. Импортировать модули для работы со случайными числами и файлами CSV



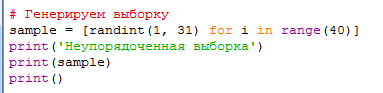
1. Инициализировать генератор случайных чисел



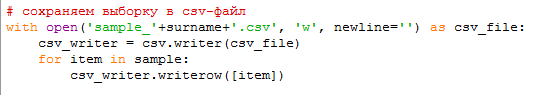
1. Создать переменную, в которой будет храниться фамилия студента. В качестве значения указать свою фамилию.



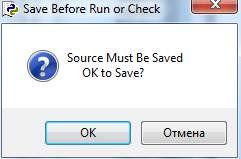
1. Сгенерировать выборку с именем **sample** из 40 случайных значений в диапазоне от 1 до 31 и вывести её на экран



1. Сохранить выборку в CSV-файл



1. Выполнить модуль клавишей F5, сохранив при необходимости



1. При этом в окне командной строки IDLE появится неупорядоченная выборка



1. В той же папке, в которой лежит выполняемый модуль, появится файл CSV, содержащий эту выборку (значок файла может отличаться). Он будет использоваться в дальнейших лабораторных работах.



Теперь модуль **gen\_sample\_Ivanov** можно закрыть.

1. Оформить отчёт, раздел «Выполнение работы» которого должен содержать:
   1. Имя и путь к файлу, содержащему выборку.
   2. Неупорядоченную выборку из 40 значений, сгенерированную модулем.

## Лабораторная работа №9. Анализ выборки данных при помощи Python

Цель работы: научиться анализировать большие выборки данных с применением языка программирования Python.

### Теоретическая часть

В данной работе будет использоваться условный оператор с двумя ветвями:

**if логическое\_выражение :**

**выражение 1**

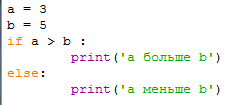
**выражение 2**

**else:**

**выражение 3**

**выражение 4**

Например:



Различные объекты в Python можно объединять в структуры. Одной из таких структур является список (**list**). Его можно создать напрямую, с именем или без него.

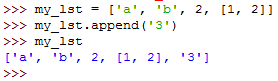


Также можно создать список при помощи функции **list()**.

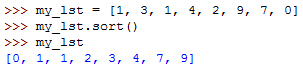


Списки имеют различные методы для работы с ними. В данной работе нам понадобятся методы добавления элементов в список и сортировки.

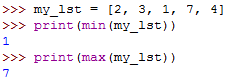
Добавить элемент к концу списка можно методом **.append()**.



Для сортировки применяется метод **.sort()**. При этом содержимое списка должно быть сортируемым. Например, нельзя отсортировать список, содержащий целые и строковые значения.



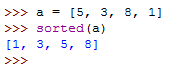
В списке также можно найти минимальное и максимальное значение.



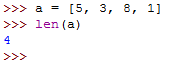
Множество задается перечислением всех его элементов в фигурных скобках. Исключением является пустое множество, которое можно создать при помощи функции **set()**. Если функции **set** передать в качестве параметра список, строку или кортеж, то она вернёт множество, составленное из элементов списка, строки, кортежа. При помощи этой функции можно убрать из списка дублирующиеся элементы и упорядочить его.



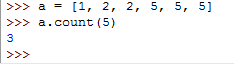
Для упорядочивания списка существуют различные способы. В данной работе будет использована функция **sorted()**.



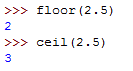
Найти длину списка можно функцией **len()**



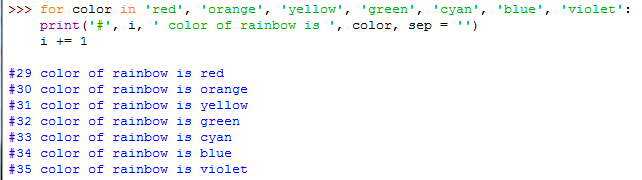
Метод списка **.count()** позволяет посчитать число уникальных элементов в списке



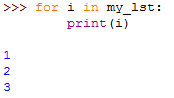
Функции **floor()** и **ceil()** используются для округления чисел в меньшую и большую сторону соответственно.



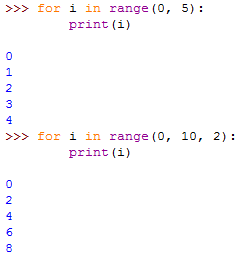
Цикл **for** в Python используется для перебора некоторого множества значений. Это множество может быть задано списком, кортежем, строкой или диапазоном.



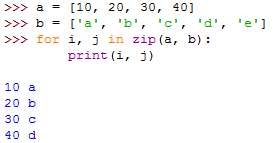
Вот перебор списка.



Для перебора значений заданного диапазона используется функция **range(begin, end, step)**. В качестве параметров она использует начальное значение (**begin**), конечное значение минус единица (**end - 1**) и шаг перебора значений (**step**).

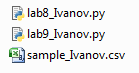


Кроме этого, В Python есть функция **zip()**, которая позволяет пройтись одновременно по нескольким итерируемым объектам (спискам и т.д.):

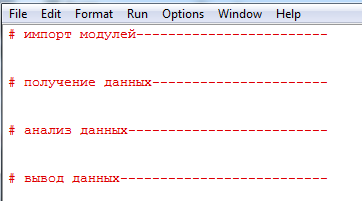


### Порядок выполнения работы

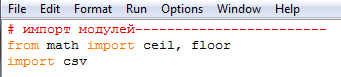
1. Запустить среду создания программ IDLE (Пуск\Все\_программы\Python\_3.7\IDLE)
2. Создать новый файл и cохранить его с именем **lab9\_Ivanov** (лабораторная работа №8 плюс фамилия студента). Этот файл должен находиться в той же папке, что и файлы **lab8\_Ivanov** и **sample\_Ivanov**



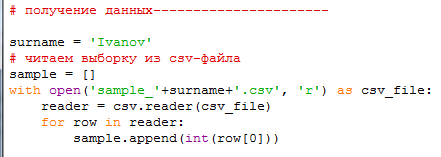
1. Разметить в **lab9\_Ivanov** секции модуля при помощи комментариев



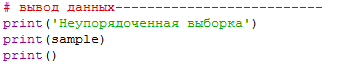
1. В секции **«импорт модулей»** импортировать из модулей функции для получения случайных чисел и работы с файлами CSV



1. В секции **«получение данных»** добавить код чтения выборки из файла, указав в переменной **surname** свою фамилию



1. В секции **«вывод данных»** добавить код вывода выборки на экран



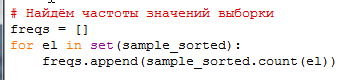
1. В секции **«анализ данных»** задать количество интервалов **ranges\_num**



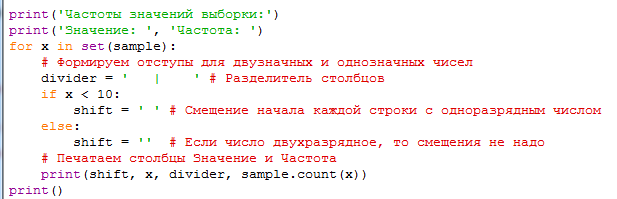
1. Упорядочить выборку. В дальнейшей работе будет использоваться только упорядоченная выборка.



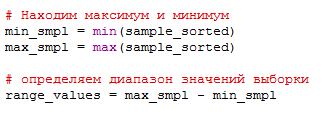
1. Найти частоты всех значений выборки (функция **count()**).



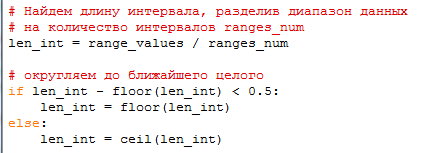
1. В секции **«вывод данных»** сформировать таблицу для вывода на экран в окне командной строки



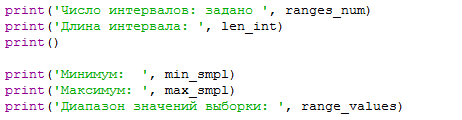
1. В секции **«анализ данных»** найти максимум, минимум и определить диапазон значений выборки



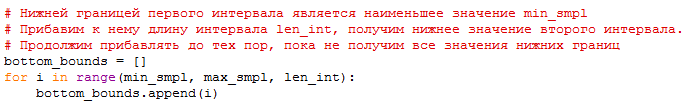
1. В секции **«анализ данных»** найти длину интервала и округлить до целых значений по математическим правилам



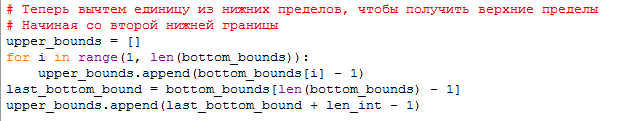
1. В секции **«вывод данных»** добавить код вывода полученных значений



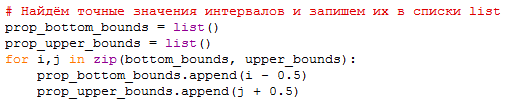
1. В секции **«анализ данных»** вычислить нижние границы интервалов и добавить их в список **bottom\_bounds**



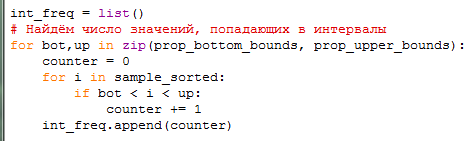
1. Вычислить верхние границы интервалов



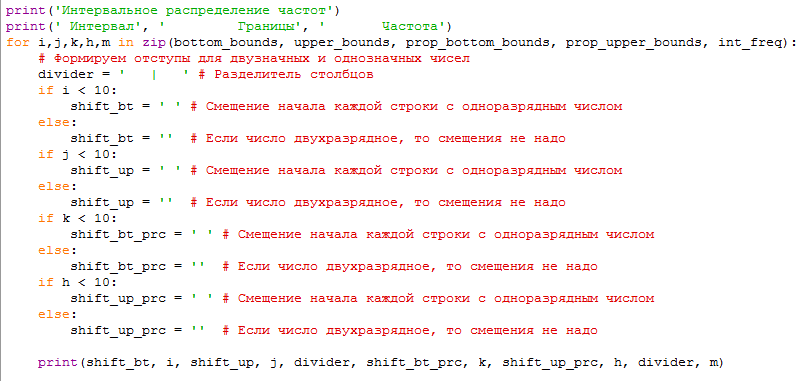
1. Найти точные границы интервалов



1. Вычислить количество значений выборки, попадающие в каждый интервал



1. В секции **«вывод данных»** добавить таблицу, содержащую интервалы, точные границы интервалов и частоты значений выборки, попадающих в эти границы



1. Оформить отчёт, раздел «Выполнение работы» которого должен содержать:
   1. Упорядоченную выборку из 40 значений, сгенерированную модулем.
   2. Минимум, максимум, диапазон, число интервалов, длину интервалов этой выборки, рассчитанные при помощи модуля, с точностью до второго знака дробной части.
   3. Таблицу интервального распределения частот этой выборки (аналогичную табл. 3)

## Лабораторная работа №10. Измерение центральной тенденции

Цель работы: вычислить параметры центральной тенденции выборки данных

### Теоретическая часть

Измерение центральной тенденции состоит в выборе одного числа, которое наилучшим образом описывает все значения признака из набора данных. Такое число называют центром, типическим значением для набора данных, мерой центральной тенденции.

**Мода** – наиболее часто встречающееся значение в выборке, наборе данных. В случае, если данные сгруппированы и построено распределение частот, модой является значение, имеющее наибольшую частоту. Обычно наличие нескольких мод указывает на то, что набор данных не подчиняется нормальному распределению.

**Среднее** определяется как среднее арифметическое выборки, то есть как сумма всех значений выборки, деленная на ее объем. Следуя определению, будем находить среднее значение по формуле:



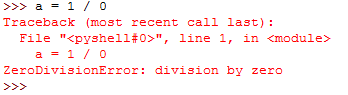
где  - сумма всех значений выборки,

 - объем выборки.

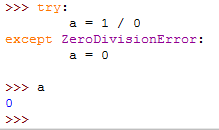
**Медиана** определяется как серединное значение выборки, или значение, выше и ниже которого располагается одинаковое число наблюдений. Для нахождения медианы обязательно **упорядочить** данные. Медиана является точной серединой выборки. Обозначается  и определяется по-разному для выборок с четным и нечетным числом элементов. Для нечетного количества наблюдений медиана есть наблюдение с номером . Для четного количества наблюдений медиана вычисляется как среднее значение наблюдений с номерами  и . В случае нечетного количества наблюдений медиана есть просто середина выборки, выше и ниже которой располагается одинаковое количество наблюдений.

Для использования статистических функций нужно установить модуль **statistics**. Он содержит набор стандартных функций для вычисления параметров центральной тенденции и вариации.

В некоторых вариантах данной работы будет использоваться обработка исключений Python. Эта конструкция позволяет не прерывать работу модуля при возникновении исключений (exceptions). Это тип данных в языке Python. Они необходимы для того, чтобы сообщать программисту об ошибках. Пример исключения – деление на ноль



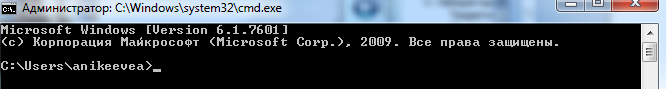
Имя возникшего исключения – ZeroDivisionError. Для обработки исключений используется конструкция try - except



Таким образом, модуль, обработав это исключение, продолжит работу.

### Порядок выполнения работы

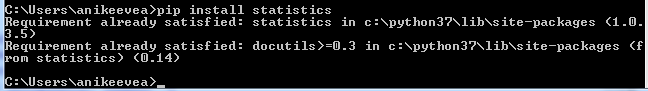
1. Для упорядоченной выборки (10 значений) из табл. 1 найти моду, медиану, среднее.
2. Запустить командную строку Windows, набрав **cmd** в строке поиска кнопки «Пуск».



1. Установить для текущего пользователя библиотеку анализа статистических данных **statistics** или убедиться в её наличии. Для установки и удаления модулей **Python** используется специальная программа – установщик пакетов **pip**. В командной строке нужно набрать команду для установки нужного пакета.



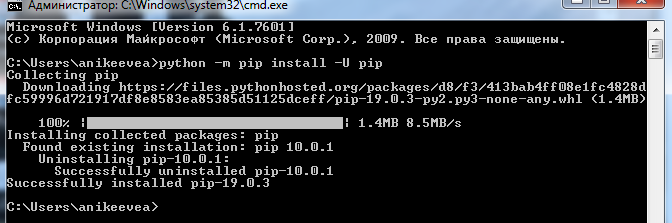
Если библиотека установлена, то в командной строке появится сообщение:



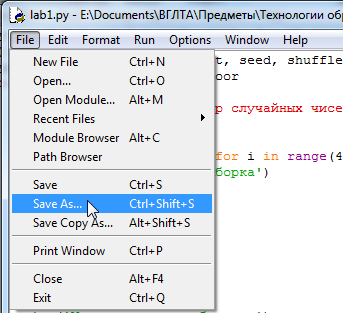
Если библиотека отсутствует, то установщик **pip** её добавит:



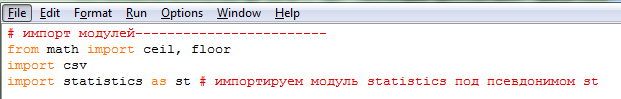
1. Если появится сообщение о необходимости обновления установщика, нужно это сделать.



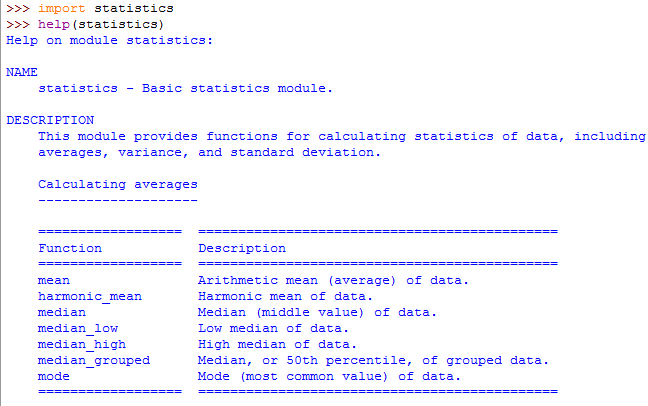
1. Открыть модуль **lab9\_Ivanov** (лаб. работа №8) и сохранить под именем **lab10\_Ivanov**. Модуль **lab9\_Ivanov** закрыть.



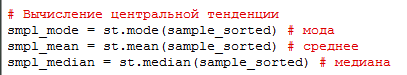
1. В секции **«импорт модулей»** импортировать модуль **statistics**, добавив код:



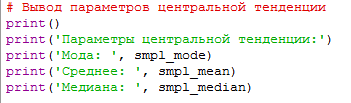
1. В командной строке интерпретатора Python запросить справку по модулю.



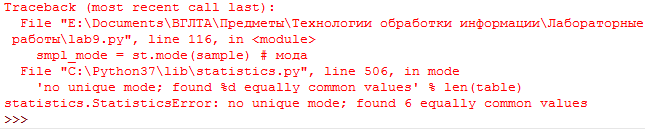
1. В секции **«анализ данных»** конце модуля добавить код вычисления моды, медианы и среднего выборки sample



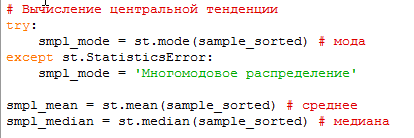
1. В секции **«вывод данных»** добавить код вывода результатов



1. В случае, если распределение имеет несколько значений, повторяющихся одинаковое количество раз (т.е. оно многомодовое), при запуске модуля появится сообщение об ошибке



В этом случае в отчёте нужно отметить, что распределение многомодовое, и указать значения моды в отчёте, проанализировав таблицу частот значений выборки. Для того, чтобы модуль продолжал работу, а не прекращал её при возникновении исключения, нужно в секции **«вывод данных»** изменить часть **«Вычисление центральной тенденции»**, применив конструкцию обработки исключений



1. Оформить отчёт, который должен содержать:
   1. Упорядоченную выборку из 10 значений согласно своему варианту (табл.1).
   2. Параметры центральной тенденции для этой выборки.
   3. Упорядоченную выборку из 40 значений, сгенерированную модулем (или должен быть в наличии отчёт по предыдущей лабораторной работе, содержащий эту выборку).
   4. Таблицу интервального распределения этой выборки.
   5. Параметры центральной тенденции этой выборки, рассчитанные при помощи модуля, с точностью до второго знака дробной части.

### Варианты заданий

Вариант задания в этой работе соответствует варианту в предыдущей работе.

## Лабораторная работа №11. Измерение вариации

Цель работы: вычислить параметры вариации для выборки данных

### Теоретическая часть

Измерение вариации состоит в нахождении чисел, которые характеризуют степень разброса данных относительно центра распределения. К параметрам вариации относят размах, квартильный размах, дисперсию, стандартное отклонение и коэффициент вариации.

**Размах** – это разница между наибольшим и наименьшим значениями. Для нахождения размаха прежде рекомендуется упорядочить данные в порядке возрастания. Можно записать размах с помощью формулы:

.

**Дисперсия** для набора данных или выборки – это среднее арифметическое квадратов отклонений значений от их среднего.

Дисперсия обозначается . Основная формула (по определению) для нахождения дисперсии:



Формула означает, что нам следует вычитать среднее из каждого значения выборки, суммировать квадраты разности, а затем разделить полученную сумму на количество наблюдений минус 1.

**Стандартное отклонение** – квадратный корень из дисперсии выборки. Обозначается s и вычисляется по формуле:

.

**Коэффициент вариации** вычисляется как отношение стандартного отклонения к среднему значению выборки.

Формула для коэффициента вариации:

.

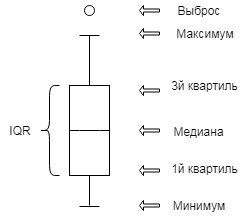
**Квартили** – это значения, которые делят вариационный ряд на четыре равные по объему части. Для начала данные следует упорядочить. Затем отыскивается медиана, которая является вторым квартилем по определению. После этого находится первый квартиль как медиана части выборки от первого значения до второго квартиля. Затем – третий квартиль как медиана части выборки от второго квартиля до последнего значения.

**Квартильный размах** (Inter Quartile Range, IQR) – это разница между третьим и первым квартилями. **Квартильный размах** находится по формуле:



Таких значений должно быть три: первый, второй и третий квартиль соответственно. Все они находятся как медианы. Второй квартиль – это медиана выборки. Первый квартиль – это медиана части выборки от первого значения до медианного. Третий квартиль – это медиана части выборки от медианного значения до последнего.

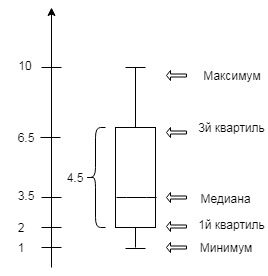
**Коробковая диаграмма** позволяет наглядно сравнивать распределение и параметры центральной тенденции нескольких выборок. На ней отображается медиана, первый и третий квартили, максимальное и минимальное значения, а также выбросы. Это те значения выборки, которые отстоят далеко от основной массы значений (на полтора квартильного размаха).



Коробковая диаграмма также называется «ящик с усами». Она показывает средние 50 процентов значений данных (квартильный размах, IQR). Он отражает изменчивость в наборе значений. Большой IQR указывает на большой разброс значений.

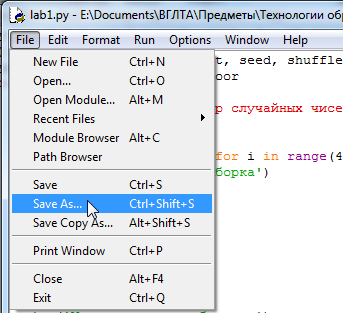
Построим «ящик с усами» для выборки

[4, 2, 3, 7, 9, 2, 5, 1, 9, 2, 1, 10, 3, 4, 6, 1, 2, 2, 5, 9]

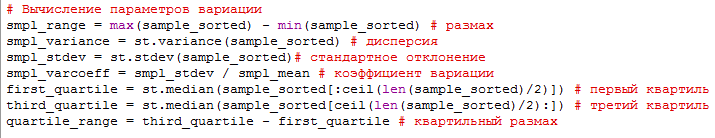


### Порядок выполнения работы

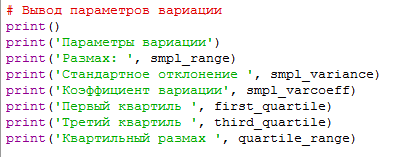
1. Для своего варианта упорядоченной выборки из табл. 1 найти параметры вариации: стандартное отклонение, коэффициент вариации, первый и третий квартили, а также квартильный размах.
   1. Размах.
   2. Дисперсию.
   3. Стандартное отклонение.
   4. Коэффициент вариации.
   5. Первый, второй и третий квартили.
   6. Квартильный размах.
2. Открыть модуль **lab9\_Ivanov** (лаб. работа №8) и сохранить под именем **lab10\_Ivanov**.



1. В конце секции **«анализ данных»** добавить код вычисления параметров вариации выборки



1. В конце модуля добавить код вывода результатов вычисления параметров вариации.



1. Оформить отчёт, который должен содержать:
2. Упорядоченную выборку из 10 значений согласно своему варианту (табл.1).
3. Расчёты и результаты выполнения пункта 1.
4. Коробковую диаграмму выборки из 10 значений.
5. Упорядоченную выборку из 40 значений, сгенерированную модулем
6. Параметры вариации для этой выборки, рассчитанные при помощи модуля. Результаты округлять до второго знака.

### Варианты заданий

Вариант задания в этой работе соответствует варианту в предыдущей работе.

## Лабораторная работа №12. Визуализация данных

Цель работы: построить графики значений выборки, гистограмму и коробковую диаграмму.

### Теоретическая часть

Для работы с двумерной графикой в языке Python существуют различные библиотеки. Наиболее популярной является matplotlib. Она представляет собой модуль для Python.

Объектом самого высокого уровня при работе с matplotlib является рисунок (Figure). Любой рисунок в matplotlib имеет вложенную структуру.

**Figure(Рисунок) -> Axes(Область рисования) -> Axis(Координатная ось)**

**Рисунок (Figure)** является объектом верхнего уровня. На нём располагаются одна или несколько областей рисования (Axes), элементы рисунка Artisits (заголовки, легенда и т.д.), а также основа-холст (Canvas). Рисунок может содержать одну или более областей рисования Axes. Каждая область рисования Axes может принадлежать только одному рисунку Figure.

**Область рисования (Axes)** является объектом среднего уровня. Это часть изображения с пространством данных. Каждая область рисования Axes содержит две (или три в случае трёхмерных данных) координатных оси (Axis объектов), которые упорядочивают отображение данных.

**Координатные оси (Axis)** также являются объектом среднего уровня. Они определяют область изменения данных. На них наносятся деления **ticks** и подписи к делениям **ticklabels**.

**Элементы рисунка (Artists)** включают в себя такие простые объекты как текст (**Text**), плоская линия (**Line2D**), фигура (**Patch**) и другие.

Когда происходит отображение рисунка (figure rendering), все элементы рисунка **Artists** наносятся на основу-холст (**Canvas**). Они связываются с областью рисования **Axes**. Каждый элемент рисунка принадлежит только одной области рисования и не может быть перемещён на другую.

Существует два типа объектов-классов **Artists**: контейнеры (**containers**) и примитивы (**primitives**).

Примитивы представляют собой стандартные графические объекты: плоскую линию, прямоугольник (**Rectangle**), текст (**Text**), изображение (**AxesImage**) и т.д.

Контейнеры представляют собой объекты-хранилища. На них рисуются графичиеские примитивы. К контейнерам относятся рисунок (**Figure**), область рисования (**Axes**), координатная ось (**Axis**), деления (**Ticks**). Рассмотрим контейнеры подробнее, так как именно с помощью обращений к различным контейнерам класса Artists, объединённых логически в единую структуру, будет осуществляться пользовательская настройка рисунков в matplotlib.

Всего существует 4 вида Artists контейнеров:

**Контейнеры рисунка (Figure containers)**

Figure - это контейнер самого высокого уровня. На нём располагаются все другие контейнеры и графические примитивы.

**Контейнеры областей рисования (Axes containers)** - это области, располагающиеся в контейнере **Figure.** Для них можно задавать координатную систему (декартовая или полярная). На нём располагаются все другие контейнеры, кроме **Figure,** а также графические примитивы. Это области на рисунке, на которых располагаются графики и диаграммы, в которые вставляются изображения и т.д. Многооконные рисунки состоят из набора областей **Axes**.

**Контейнеры осей (Axis containers)** отвечают за создание координатных осей (а именно линий осей), на которые будут наноситься деления, подписи делений и линий вспомогательной сетки. Его специализация (и отличие от контейнера **Tick**) - это расположение делений и линий, их позиционирование и форматирование подписей делений, а также их отображение.

**Контейнеры делений (Tick containers)** являются **к**онтейнерами низшего уровня. С их помощью можно задавать характеристики (цвет, толщина линий) линий сетки, делений и их подписей (размеры и типы шрифтов).

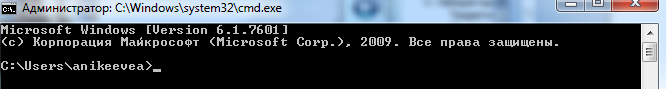
Общий алгоритм создания рисунка в matplotlib обычно включает в себя следующие шаги:

1. Создать экземпляр класса **Figure** (Figure instance)
2. На нём выделить одну или нескольких областей **Axes** или экземпляров **Subplot**
3. Использовать вспомогательные методы экземпляра класса **Axes** (Axes instance) для создания графических примитивов (primitives).

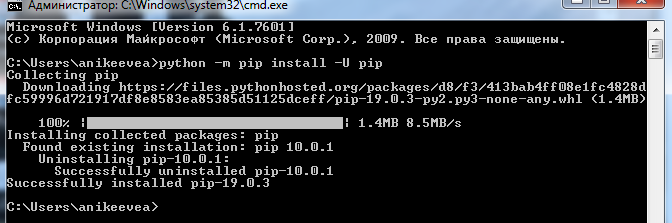
Если автоматически подобранные характеристики координатной сетки, делений и их подписей не устраивают пользователя, то они настраиваются с помощью экземпляров контейнеров **Axis** и **Tick**. Эти контейнеры всегда присутствуют на созданной области рисования **Axes**.

### Порядок выполнения работы

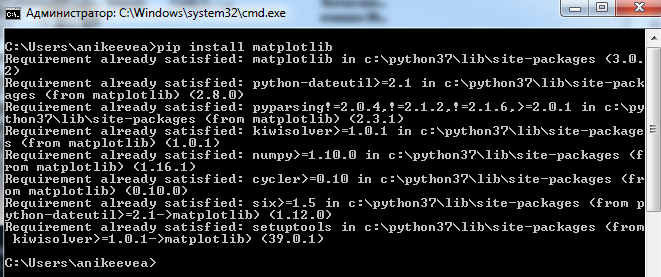
1. Запустить командную строку Windows, набрав **cmd** в строке поиска кнопки «Пуск».



1. Установить для текущего пользователя библиотеку двумерной графики **matplotlib** или убедиться в её наличии. В командной строке нужно набрать команду для установки нужного пакета при помощи установщика **pip**.
2. Если появится сообщение о необходимости обновления установщика, нужно это сделать.



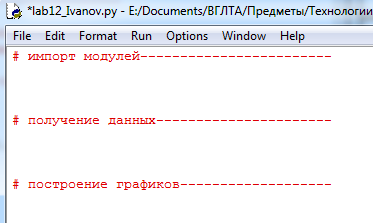
1. Если библиотека установлена, то в командной строке появится сообщение:



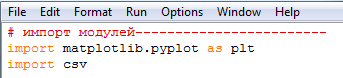
1. Если библиотека отсутствует, то установщик **pip** её добавит:



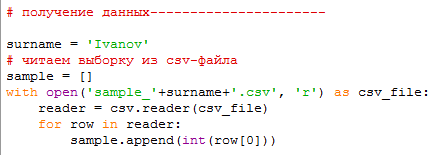
1. Для того чтобы не перегружать модуль расчётов, графический модуль напишем отдельно. При этом воспользуемся данными выборки, сохранёнными в файле **sample\_Ivanov.csv**.
2. Окрыть **IDLE**, создать и сохранить файл **lab12\_Ivanov**. Он должен находиться в той же папке ,что и файл с данными **sample\_Ivanov.csv**.
3. Добавить в файл **lab12\_Ivanov** разметку секций кода.



1. В секцию **«импорт модулей»** добавить код импорта библиотек графики и работы с csv-файлами.



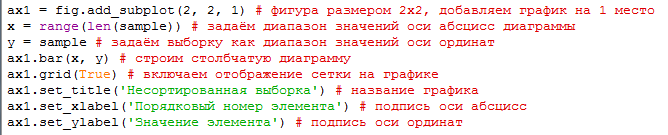
1. В секцию «получение данных» добавить код чтения выборки из файла.



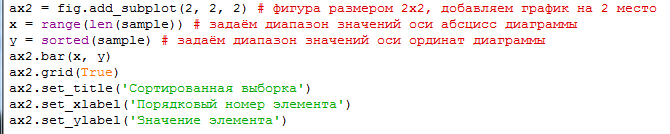
1. В секцию **«построение графиков»** добавить код создания фигуры, которая будет содержать графики, и задать её размер.



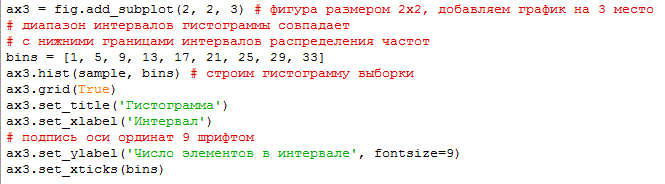
1. Добавить код построения графика несортированной выборки.



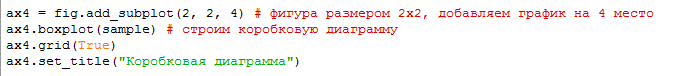
1. Добавить код построения графика сортированной выборки.



1. Добавить код построения гистограммы.



1. Добавить код построения коробковой диаграммы.



1. Добавить код вывода изображений на экран.



1. Оформить отчёт, который должен содержать:
2. Графики неупорядоченной и упорядоченной выборки.
3. Гистограмму
4. Коробковую диаграмму
5. На коробковой диаграмме обозначить медиану, первый и третий квартили, минимальное и максимальное значения.

### Варианты заданий

Вариант задания в этой работе соответствует варианту в предыдущей работе.

## Список литературы

1. http://www.ibm.com/developerworks/ru/library/l-scilab1/
2. http://ru.wikibooks.org/wiki/Scilab/Программирование
3. <http://help.scilab.org/docs/5.5.0/ru_RU/index.html>
4. Python.org