StatisticsAndModeling

1.0.0

Создано системой Doxygen 1.9.1

1 NB distribution 1

2 Иерархический список классов 3
2.1 Иерархия классов . . . 3

3 Алфавитный указатель классов 5
3.1 Классы . . . 5

4 Список файлов 7
4.1 Файлы . . . 7

5 Классы 9
5.1 Класс ChiSqHist . . . 9
5.1.1 Подробное описание . . . 10
5.1.2 Конструктор(ы) . . . 10
5.1.2.1 ChiSqHist() [1/3] . . . 10
5.1.2.2 ChiSqHist() [2/3] . . . 10
5.1.2.3 ChiSqHist() [3/3] . . . 11
5.1.3 Методы . . . 11
5.1.3.1 get_exp_freq() . . . 11
5.1.3.2 get_num_freq() . . . 11
5.1.3.3 get_p_value() . . . 12
5.1.3.4 get_th_freq() . . . 12
5.1.3.5 operator=() . . . 12
5.1.3.6 set_data() . . . 13
5.2 Класс Doc_NB . . . 13
5.2.1 Подробное описание . . . 14
5.2.2 Конструктор(ы) . . . 14
5.2.2.1 Doc_NB() [1/2] . . . 14
5.2.2.2 Doc_NB() [2/2] . . . 15
5.2.3 Методы . . . 15
5.2.3.1 change_param() . . . 15
5.2.3.2 get_chi_sq() . . . 15
5.2.3.3 get_d0() . . . 16
5.2.3.4 get_d1() . . . 16
5.2.3.5 get_num_p_value() . . . 16
5.2.3.6 get_p_value() . . . 16
5.2.3.7 get_Sample() . . . 17
5.2.3.8 get_sign_lv() . . . 17
5.2.3.9 operator=() . . . 17
5.3 Класс NB_distr . . . 18
5.3.1 Подробное описание . . . 18
5.3.2 Конструктор(ы) . . . 18
5.3.2.1 NB_distr() . . . 18
5.3.3 Методы . . . 19

5.3.3.1 get_k() . . . 19
5.3.3.2 get_p() . . . 19
5.3.3.3 get_prob_now() . . . 19
5.3.3.4 name_of_distr() . . . 20
5.3.3.5 next_prob() . . . 20
5.4 Класс Sample . . . 20
5.4.1 Подробное описание . . . 22
5.4.2 Конструктор(ы) . . . 22
5.4.2.1 Sample() [1/3] . . . 22
5.4.2.2 Sample() [2/3] . . . 22
5.4.2.3 Sample() [3/3] . . . 22
5.4.3 Методы . . . 24
5.4.3.1 change_param() . . . 24
5.4.3.2 get_n() . . . 24
5.4.3.3 get_name() . . . 25
5.4.3.4 operator[]() . . . 25
5.4.3.5 simulate_one() . . . 25
5.4.3.6 swap() . . . 25
5.5 Класс Sample_Bernulli . . . 26
5.5.1 Подробное описание . . . 27
5.5.2 Конструктор(ы) . . . 27
5.5.2.1 Sample_Bernulli() [1/3] . . . 27
5.5.2.2 Sample_Bernulli() [2/3] . . . 28
5.5.2.3 Sample_Bernulli() [3/3] . . . 28
5.5.3 Методы . . . 28
5.5.3.1 get_name() . . . 28
5.5.3.2 operator=() . . . 29
5.5.3.3 simulate_one() . . . 30
5.6 Класс Sample_Table . . . 30
5.6.1 Подробное описание . . . 31
5.6.2 Конструктор(ы) . . . 32
5.6.2.1 Sample_Table() [1/3] . . . 32
5.6.2.2 Sample_Table() [2/3] . . . 32
5.6.2.3 Sample_Table() [3/3] . . . 32
5.6.3 Методы . . . 33
5.6.3.1 change_param() . . . 33
5.6.3.2 get_name() . . . 33
5.6.3.3 operator=() . . . 33
5.6.3.4 simulate_one() . . . 34

6 Файлы 35
6.1 Файл src/Doc_NB.h . . . 35
6.1.1 Подробное описание . . . 36

7 Примеры 37
7.1 main.cpp 37

Предметный указатель 39

Глава 1

NB distribution

Автор

Олейник Михаил

Дата

03.05.2023

Основная страница набора классов по моделированию отрицательно-биномиального распределения. Их возможности позволяют

1. Моделировать выборки случайных величин, распределённых по отрицательно биномиальному закону, методами Бернулли и табличным;
2. Менять их параметры (такие как вероятность успеха, количество успехов, размер выборки);
3. Получать теоретические вероятности и эмперические частоты;
4. Считать критерий χ^2 и p-value;
5. Моделировать выборку p-value и менять для этого метод моделирования, размер выборки, гипотезу, на основе которой моделируется выборка.

NB_distr - класс распределения, хранящий параметры отрицательно-биномиального распределения.

Sample - базовый класс для моделирования выборок.

Sample_Table - класс моделирования выборок табличным методом.

Sample_Bernulli - класс моделирования выборок методом Бернулли.

ChiSqHist - класс критерия согласия χ^2.

Doc_NB - класс моделирования выборок p-value.

Пример использования классов:

```
#include <iostream>
#include "../Doc_NB.h"
int main(int argc, char** argv){
    // Инициализация распределений по умолчанию и с параметрами.
    NB_distr d0, d1(0.84, 11);
```

```
    // Название распределения.
    std::cout « d0.name_of_distr() « "\n";
    // Инициализация метода моделирования Бернулли.
    Sample_Bernulli sam_ber(100, &d0);
    // Моделирование выборки.
    sam_ber.simulate();
    // Название метода.
    std::cout « sam_ber.get_name() « ": ";
    // Выборка.
    for (size_t i = 0; i < sam_ber.get_n(); ++i)
        std::cout « sam_ber[i] « " ";

    std::cout « "\n";
    // Изменение размера выборки.
    sam_ber.change_param(10);
    sam_ber.simulate();
    std::cout « sam_ber.get_name() « ": ";
    for (size_t i = 0; i < sam_ber.get_n(); ++i)
        std::cout « sam_ber[i] « " ";

    std::cout « "\n";
    // Инициализация табличного метода моделирования.
    Sample_Table sam_table(70, &d0);
    sam_table.simulate();
    std::cout « sam_table.get_name() « ": ";
    for (size_t i = 0; i < sam_table.get_n(); ++i)
        std::cout « sam_table[i] « " ";

    std::cout « "\n";
    // Инициализация класса моделирования выборок p-value.
    Doc_NB doc;
    // Изменение параметров класса моделирования выборок p-value.
    doc.change_param(d0, d1, 100, 120, 0.1);
    // Установка в качестве распределения альтернативной гипотезы.
    doc.set_hyp_d1();
    // Установка в качестве метода моделирования метода Бернулли.
    doc.set_bernulli_method();
    // Моделирование выборки p-value.
    doc.make_p_value();
    // Получение доступа к классу хи квадрат.
    ChiSqHist* chi_sq = doc.get_chi_sq();
    // Вывод вычисленного p-value.
    std::cout « "P-value: " « chi_sq->get_p_value() « "\n";
    // Вычисления таблицы теоретичесикх вероятностей.
    chi_sq->calc_th_freq();
    std::cout « "Теоретические вероятности: ";
    // Вывод теоретических вероятностей.
    for (size_t i = 0; i < chi_sq->get_num_freq(); ++i)
        std::cout « chi_sq->get_th_freq()[i] « " ";

    std::cout « "\n";
    // Вычисление таблицы эмперичесикх частот.
    chi_sq->calc_exp_freq();
    std::cout « "Эмперические частоты: ";
    // Вывод эмперических частот.
    for (size_t i = 0; i < chi_sq->get_num_freq(); ++i)
        std::cout « chi_sq->get_exp_freq()[i] « " ";

    std::cout « "\n";
    std::cout « "Выборка p-value: ";
    // Вывод выборки p-value.
    for (size_t i = 0; i < doc.get_num_p_value(); ++i)
        std::cout « doc.get_p_value(i) « " ";
    std::cout « "\n";
    // Установка в качестве распределения нулевой гипотезы.
    doc.set_hyp_d0();
    // Установка в качестве метода моделирования табличного метода.
    doc.set_table_method();
    doc.make_p_value();
    std::cout « "Выборка p-value: ";
    for (size_t i = 0; i < doc.get_num_p_value(); ++i)
        std::cout « doc.get_p_value(i) « " ";
    std::cout « "\n";
    return 0;
}
```

Глава 2

Иерархический список классов

2.1 Иерархия классов

Иерархия классов.

ChiSqHist 9
Doc_NB 13
NB_distr 18
Sample 20
  Sample_Bernulli 26
  Sample_Table 30

Глава 3

Алфавитный указатель классов

3.1 Классы

Классы с их кратким описанием.

ChiSqHist
: Класс критерия согласия 9

Doc_NB
: Класс моделирования и гипотез 13

NB_distr
: Класс отрицательно-биномиального распределения 18

Sample
: Класс моделирования распределений 20

Sample_Bernulli
: Класс моделирования распределения методом Бернулли 26

Sample_Table
: Класс моделирования распределения табличным методом 30

Глава 4

Список файлов

4.1 Файлы

Полный список документированных файлов.

src/Doc_NB.h
Моделирование отрицательно-биномиального распределения 35

Глава 5

Классы

5.1 Класс ChiSqHist

Класс критерия согласия.

#include <Doc_NB.h>

Открытые члены

- ChiSqHist (NB_distr *_d, Sample *_s)
  Конструктор критерия согласия по указателям на распределение и метод моделирования.
- ChiSqHist (ChiSqHist &c)
  Конструктор копирования.
- ChiSqHist (ChiSqHist &&c)
  Конструктор перемещения.
- ChiSqHist & operator= (ChiSqHist c)
  Оператор присваивания для класса ChiSqHist.
- double get_p_value () const
  Доступ к p-value.
- size_t get_num_freq () const
  Доступ к размеру массива с вероятностями.
- const size_t * get_exp_freq () const
  Доступ к эмперическим частотам.
- const double * get_th_freq () const
  Доступ к теоретическим вероятностям.
- void set_data (NB_distr *_d, Sample *_s)
  Изменение распределения и метода моделирования, на основе, которых вычисляется критерий.
- void calc_th_freq ()
  Составление таблицы теоретических вероятностей.
- void calc_exp_freq ()
  Составление таблицы эмперических частот.
- void calc_chi_sq ()
  Вычисление критерия χ^2 и p-value.
- ~ChiSqHist ()
  Деструктор ChiSqHist.

5.1.1 Подробное описание

Класс критерия согласия.

Класс, который хранит вычисленные теоретические и эмперические вероятности распределения и выборки, вычисляет критерий χ^2 и значение p-value. Позволяет сменить распределение и метод моделирования.

Примеры

main.cpp.

5.1.2 Конструктор(ы)

5.1.2.1 ChiSqHist() [1/3]

```
ChiSqHist::ChiSqHist (
            NB_distr * _d,
            Sample * _s )
```

Конструктор критерия согласия по указателям на распределение и метод моделирования.

Аргументы

in	↩ _↩ d	Указатель на распределение.
in	↩ _↩ s	Указатель на метод моделирования.

5.1.2.2 ChiSqHist() [2/3]

```
ChiSqHist::ChiSqHist (
            ChiSqHist & c )
```

Конструктор копирования.

Аргументы

in	c	Объект класса ChiSqHist.

5.1.2.3 ChiSqHist() [3/3]

ChiSqHist::ChiSqHist (
ChiSqHist && c)

Конструктор перемещения.

Аргументы

in	c	Объект класса ChiSqHist.

5.1.3 Методы

5.1.3.1 get_exp_freq()

const size_t* ChiSqHist::get_exp_freq ( ) const [inline]

Доступ к эмперическим частотам.

Возвращает

Указатель на массив эмпиричесикх частот.

Примеры

main.cpp.

5.1.3.2 get_num_freq()

size_t ChiSqHist::get_num_freq ( ) const [inline]

Доступ к размеру массива с вероятностями.

Возвращает

Размер массива с вероятностями.

Примеры

main.cpp.

5.1.3.3 get_p_value()

double ChiSqHist::get_p_value ( ) const [inline]

Доступ к p-value.

Возвращает

Значение p-value.

Примеры

main.cpp.

5.1.3.4 get_th_freq()

const double* ChiSqHist::get_th_freq ( ) const [inline]

Доступ к теоретическим вероятностям.

Возвращает

Указатель на массив теоретических вероятностей.

Примеры

main.cpp.

5.1.3.5 operator=()

ChiSqHist& ChiSqHist::operator= (
ChiSqHist c)

Оператор присваивания для класса ChiSqHist.

Аргументы

in	c	Объект класса ChiSqHist.

Возвращает

Результат присваивания, объект класса ChiSqHist.

5.1.3.6 set_data()

void ChiSqHist::set_data (
NB_distr * _d,
Sample * _s)

Изменение распределения и метода моделирования, на основе, которых вычисляется критерий.

Аргументы

in	↩ _↩ d	Указатель на распределение.
in	↩ _↩ s	Указатель на метод моделирования.

Объявления и описания членов класса находятся в файле:

- src/Doc_NB.h

5.2 Класс Doc_NB

Класс моделирования и гипотез.

#include <Doc_NB.h>

Открытые члены

- Doc_NB ()
  Конструктор класса моделирования и гипотез.
- Doc_NB (Doc_NB &d)
  Конструктор копирования.
- Doc_NB (Doc_NB &&d)
  Конструктор перемещения.
- Doc_NB & operator= (Doc_NB d)
  Оператор присваивания для класса Doc_NB.
- Sample * get_Sample () const
  Доступ к методу моделирования.
- size_t get_num_p_value () const
  Доступ к размеру выборки p-value.
- double get_sign_lv () const
  Доступ к уровню значимости.
- ChiSqHist * get_chi_sq ()
  Доступ к критерию согласию χ^2.
- const NB_distr * get_d0 () const
  Доступ к нулевой гипотезе.
- const NB_distr * get_d1 () const
  Доступ к альтернативной гипотезе.

- void make_p_value ()

  Моделирование выборки p-value.
- double get_p_value (size_t i) const

  Доступ к элементу выборки p-value.
- void change_param (NB_distr _d0, NB_distr _d1, size_t _num_p_value, size_t _n, double _sign_lv)

  Позволяет изменить параметры гипотез, размер выборки p-value, размер выборки и уровень значимости.
- void set_table_method ()

  Установка в качестве метода моделирования табличного метода.
- void set_bernulli_method ()

  Установка в качестве метода моделирования метода Бернулли.
- void set_hyp_d0 ()

  Установка для моделирования нулевую гипотезу.
- void set_hyp_d1 ()

  Установка для моделирования альтернативную гипотезу.
- ~Doc_NB ()

  Деструктор Doc_NB.

5.2.1 Подробное описание

Класс моделирования и гипотез.

Класс, который хранит нулевую и альтернативную гипотезы, метод моделирования, объект критерия χ^2, выборку p_value, уровень значимости. Позволяет менять параметры распеределений, методы моделирования, критерий и размер выборки p-value. Моделирует выборку p-value.

Примеры

main.cpp.

5.2.2 Конструктор(ы)

5.2.2.1 Doc_NB() [1/2]

Doc_NB::Doc_NB (
Doc_NB & d)

Конструктор копирования.

Аргументы

in	d	Объект класса Doc_NB.

5.2.2.2 Doc_NB() [2/2]

```
Doc_NB::Doc_NB (
            Doc_NB && d )
```

Конструктор перемещения.

Аргументы

in	d	Объект класса Doc_NB.

5.2.3 Методы

5.2.3.1 change_param()

```
void Doc_NB::change_param (
            NB_distr _d0,
            NB_distr _d1,
            size_t _num_p_value,
            size_t _n,
            double _sign_lv )
```

Позволяет изменить параметры гипотез, размер выборки p-value, размер выборки и уровень значимости.

Аргументы

in	_d0	Нулевая гипотеза.
in	_d1	Альтернативная гипотеза.
in	_num_p_value	Размер выборки p-value.
in	_n	Размер выборки.
in	_sign_lv	Уровень значимости.

Примеры

main.cpp.

5.2.3.2 get_chi_sq()

```
ChiSqHist* Doc_NB::get_chi_sq ( )  [inline]
```

Доступ к критерию согласию χ^2.

Возвращает

Указатель на критерий согласия χ^2.

Примеры

main.cpp.

5.2.3.3 get_d0()

const NB_distr* Doc_NB::get_d0 ( ) const [inline]

Доступ к нулевой гипотезе.

Возвращает

Указатель на нулевую гипотезу.

5.2.3.4 get_d1()

const NB_distr* Doc_NB::get_d1 ( ) const [inline]

Доступ к альтернативной гипотезе.

Возвращает

Указатель на альтернативную гипотезу.

5.2.3.5 get_num_p_value()

size_t Doc_NB::get_num_p_value ( ) const [inline]

Доступ к размеру выборки p-value.

Возвращает

Размер выборки p-value.

Примеры

main.cpp.

5.2.3.6 get_p_value()

double Doc_NB::get_p_value (
size_t i) const

Доступ к элементу выборки p-value.

Аргументы

in	i	Индекс элемента выборки p-value.

Возвращает

Значение элемента выборки p-value.

Примеры

main.cpp.

5.2.3.7 get_Sample()

Sample* Doc_NB::get_Sample ( ) const [inline]

Доступ к методу моделирования.

Возвращает

Указатель на метод моделирования.

5.2.3.8 get_sign_lv()

double Doc_NB::get_sign_lv ( ) const [inline]

Доступ к уровню значимости.

Возвращает

Уровень значимости.

5.2.3.9 operator=()

Doc_NB& Doc_NB::operator= (
Doc_NB d)

Оператор присваивания для класса Doc_NB.

Аргументы

in	d	Объект класса Doc_NB.

Возвращает

Результат присваивания, объект класса Doc_NB.

Объявления и описания членов класса находятся в файле:

- src/Doc_NB.h

5.3 Класс NB_distr

Класс отрицательно-биномиального распределения.

#include <Doc_NB.h>

Открытые члены

- NB_distr (double _p=0.8, size_t _k=10)
  Конструктор по параметрам распределения: вероятности успеха и количества успехов.
- double get_p () const
  Доступ к p.
- size_t get_k () const
  Доступ к k.
- double get_prob_now () const
  Доступ к prob_now.
- double next_prob ()
  Вычисляет следующую вероятность распределения.
- const char * name_of_distr () const
  Функция содержащая название распределения.
- void reset ()
  Обнуляет вычисление вероятностей.

5.3.1 Подробное описание

Класс отрицательно-биномиального распределения.

Класс, содержащий параметры отрицательно-биномиального распределения и вычисляющий его вероятности.

Примеры

main.cpp.

5.3.2 Конструктор(ы)

5.3.2.1 NB_distr()

```
NB_distr::NB_distr (
            double _p = 0.8,
            size_t _k = 10 )
```

Конструктор по параметрам распределения: вероятности успеха и количества успехов.

Аргументы

in	↩ _↩ p	Вероятность успеха.
in	↩ _↩ k	Количество успехов.

5.3.3 Методы

5.3.3.1 get_k()

size_t NB_distr::get_k ( ) const [inline]

Доступ к k.

Возвращает

Количество успехов.

5.3.3.2 get_p()

double NB_distr::get_p ( ) const [inline]

Доступ к p.

Возвращает

Вероятность успеха.

5.3.3.3 get_prob_now()

double NB_distr::get_prob_now ( ) const [inline]

Доступ к prob_now.

Возвращает

Текущую вычисленную вероятность.

5.3.3.4 name_of_distr()

const char* NB_distr::name_of_distr ( ) const

Функция содержащая название распределения.

Возвращает

Строку "Negative Binomial Distribution".

Примеры

main.cpp.

5.3.3.5 next_prob()

double NB_distr::next_prob ( )

Вычисляет следующую вероятность распределения.

Возвращает

Следующую вероятность распределения.

Объявления и описания членов класса находятся в файле:

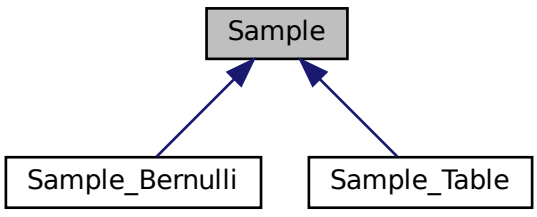
- src/Doc_NB.h

5.4 Класс Sample

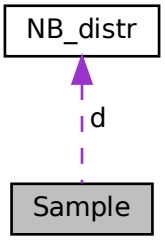
Класс моделирования распределений.

#include <Doc_NB.h>

Граф наследования:Sample:



Граф связей класса Sample:



Открытые члены

- Sample (size_t _n, NB_distr *_d)
  Конструктор модирования распределений по размеру выборки и распределению.
- Sample (const Sample &s)
  Конструктор копирования.
- Sample (Sample &&s)
  Конструктор перемещения.
- size_t get_n () const
  Доступ к размеру выборки.
- virtual const char * get_name () const =0
  Доступ к названию методу моделирования.
- void simulate ()
  Симулирует выборку, записывая её во внутренний массив.
- virtual size_t simulate_one ()=0
  Симулирует один элемент выборки.
- void change_param (size_t _n)
  Изменяет размер выборки.
- size_t operator[] (int i) const
  Доступ к элементам выборки по индексу.
- virtual ~Sample ()=0
  Деструктор Sample.

Защищенные члены

- void swap (Sample &s)
  Осуществляет обмен полями между объектом класса и переданным s.

Защищенные данные

- size_t n
  Размер выборки.
- NB_distr * d
  Указатель на класс распределения.
- size_t * sam
  Массив с выборкой.

5.4.1 Подробное описание

Класс моделирования распределений.

Базовый класс для моделирования распределений, содержащий размер выборки, указатель на распределение и массив выборки. Позволяет генерировать выборку, изменять её размер и получать её параметры и название метода.

5.4.2 Конструктор(ы)

5.4.2.1 Sample() [1/3]

```
Sample::Sample (
            size_t _n,
            NB_distr * _d )
```

Конструктор модирования распределений по размеру выборки и распределению.

Аргументы

in	↩ _↩ n	Размер выборки.
in	↩ _↩ d	Указатель на распределение.

5.4.2.2 Sample() [2/3]

```
Sample::Sample (
            const Sample & s )
```

Конструктор копирования.

Аргументы

in	s	Объект класса Sample.

5.4.2.3 Sample() [3/3]

```
Sample::Sample (
            Sample && s )
```

Конструктор перемещения.

Аргументы

in	s	Объект класса Sample.

5.4.3 Методы

5.4.3.1 change_param()

void Sample::change_param (
size_t _n)

Изменяет размер выборки.

Аргументы

in	↩ _↩ n	Размер выборки.

Примеры

main.cpp.

5.4.3.2 get_n()

size_t Sample::get_n ( ) const [inline]

Доступ к размеру выборки.

Возвращает

Размер выборки.

Примеры

main.cpp.

5.4.3.3 get_name()

virtual const char* Sample::get_name ( ) const [pure virtual]

Доступ к названию методу моделирования.

Возвращает

Константную строку с названием метода моделирования.

Замещается в Sample_Bernulli и Sample_Table.

5.4.3.4 operator[]()

size_t Sample::operator[] (
int i) const

Доступ к элементам выборки по индексу.

Аргументы

in	i	Индекс элемента выборки.

Возвращает

Значение элемента выборки.

5.4.3.5 simulate_one()

virtual size_t Sample::simulate_one ( ) [pure virtual]

Симулирует один элемент выборки.

Возвращает

Значение элемента выборки.

Замещается в Sample_Bernulli и Sample_Table.

5.4.3.6 swap()

void Sample::swap (
Sample & s) [protected]

Осуществляет обмен полями между объектом класса и переданным s.

Аргументы

s	Объект класса Sample.

Объявления и описания членов класса находятся в файле:

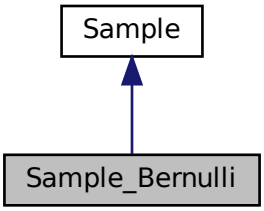
- src/Doc_NB.h

5.5 Класс Sample_Bernulli

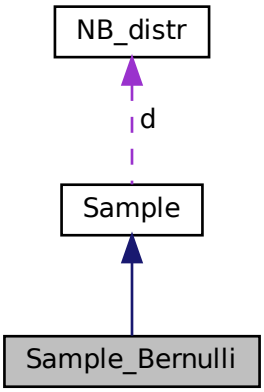
Класс моделирования распределения методом Бернулли.

#include <Doc_NB.h>

Граф наследования:Sample_Bernulli:



Граф связей класса Sample_Bernulli:



Открытые члены

- Sample_Bernulli (size_t _n, NB_distr *_d)

  Конструктор модирования распределений методом Бернулли по размеру выборки и распределению.
- Sample_Bernulli (const Sample_Bernulli &s)

  Конструктор копирования.
- Sample_Bernulli (Sample_Bernulli &&s)

  Конструктор перемещения.
- Sample_Bernulli & operator= (Sample_Bernulli s)

  Оператор присваивания для класса Sample_Bernulli.
- virtual const char * get_name () const override

  Доступ к названию методу моделирования.
- virtual size_t simulate_one () override

  Симулирует один элемент выборки.

Дополнительные унаследованные члены

5.5.1 Подробное описание

Класс моделирования распределения методом Бернулли.

Дочерний к Sample класс для моделирования распределений методом Бернулли, содержащий размер выборки, указатель на распределение и массив выборки. Позволяет генерировать выборку, изменять её размер и получать её параметры и название метода.

Примеры

main.cpp.

5.5.2 Конструктор(ы)

5.5.2.1 Sample_Bernulli() [1/3]

```
Sample_Bernulli::Sample_Bernulli (
            size_t _n,
            NB_distr * _d )
```

Конструктор модирования распределений методом Бернулли по размеру выборки и распределению.

Аргументы

in	↩ _↩ n	Размер выборки.
in	↩ _↩ d	Указатель на распределение.

5.5.2.2 Sample_Bernulli() [2/3]

Sample_Bernulli::Sample_Bernulli (
const Sample_Bernulli & s)

Конструктор копирования.

Аргументы

in	s	Объект класса Sample_Bernulli.

5.5.2.3 Sample_Bernulli() [3/3]

Sample_Bernulli::Sample_Bernulli (
Sample_Bernulli && s)

Конструктор перемещения.

Аргументы

in	s	Объект класса Sample_Bernulli.

5.5.3 Методы

5.5.3.1 get_name()

virtual const char* Sample_Bernulli::get_name ( ) const [override], [virtual]

Доступ к названию методу моделирования.

Возвращает

Константную строку "Bernulli Method".

Замещает Sample.

Примеры

main.cpp.

5.5.3.2 operator=()

Sample_Bernulli& Sample_Bernulli::operator= (
Sample_Bernulli s)

Оператор присваивания для класса Sample_Bernulli.

Аргументы

in	s	Объект класса Sample_Bernulli.

Возвращает

Результат присваивания, объект класса Sample_Bernulli.

5.5.3.3 simulate_one()

virtual size_t Sample_Bernulli::simulate_one ( ) [override], [virtual]

Симулирует один элемент выборки.

Возвращает

Значение элемента выборки.

Замещает Sample.

Объявления и описания членов класса находятся в файле:

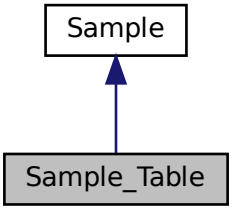
- src/Doc_NB.h

5.6 Класс Sample_Table

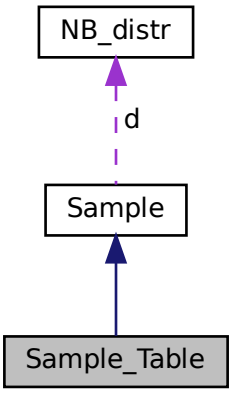
Класс моделирования распределения табличным методом.

#include <Doc_NB.h>

Граф наследования:Sample_Table:



Граф связей класса Sample_Table:



Открытые члены

- Sample_Table (size_t _n, NB_distr *_d)
  Конструктор модирования распределений табличным методом по размеру выборки и распределению.
- Sample_Table (const Sample_Table &s)
  Конструктор копирования.
- Sample_Table (Sample_Table &&s)
  Конструктор перемещения.
- Sample_Table & operator= (Sample_Table s)
  Оператор присваивания для класса Sample_Table.
- virtual const char * get_name () const override
  Доступ к названию методу моделирования.
- void change_param (size_t _n)
  Изменяет размер выборки.
- virtual size_t simulate_one () override
  Симулирует один элемент выборки.
- ~Sample_Table ()
  Деструктор Sample_Table.

Дополнительные унаследованные члены

5.6.1 Подробное описание

Класс моделирования распределения табличным методом.

Дочерний к Sample класс для моделирования распределений табличным методом, содержащий размер выборки, указатель на распределение, массив выборки и массив суммированных вероятностей. Позволяет генерировать выборку, изменять её размер и получать её параметры и название метода.

Примеры

main.cpp.

5.6.2 Конструктор(ы)

5.6.2.1 Sample_Table() [1/3]

```
Sample_Table::Sample_Table (
            size_t _n,
            NB_distr * _d )
```

Конструктор модирования распределений табличным методом по размеру выборки и распределению.

Аргументы

in	↩ _↩ n	Размер выборки.
in	↩ _↩ d	Указатель на распределение.

5.6.2.2 Sample_Table() [2/3]

```
Sample_Table::Sample_Table (
            const Sample_Table & s )
```

Конструктор копирования.

Аргументы

in	s	Объект класса Sample_Table.

5.6.2.3 Sample_Table() [3/3]

```
Sample_Table::Sample_Table (
            Sample_Table && s )
```

Конструктор перемещения.

Аргументы

in	s	Объект класса Sample_Table.

5.6.3 Методы

5.6.3.1 change_param()

void Sample_Table::change_param (
size_t _n)

Изменяет размер выборки.

Аргументы

in	↩ _↩ n	Размер выборки.

5.6.3.2 get_name()

virtual const char* Sample_Table::get_name ( ) const [override], [virtual]

Доступ к названию методу моделирования.

Возвращает

Константную строку "Table Method".

Замещает Sample.

Примеры

main.cpp.

5.6.3.3 operator=()

Sample_Table& Sample_Table::operator= (
Sample_Table s)

Оператор присваивания для класса Sample_Table.

Аргументы

in	s	Объект класса Sample_Table.

Возвращает

Результат присваивания, объект класса Sample_Table.

5.6.3.4 simulate_one()

virtual size_t Sample_Table::simulate_one ( ) [override], [virtual]

Симулирует один элемент выборки.

Возвращает

Значение элемента выборки.

Замещает Sample.

Объявления и описания членов класса находятся в файле:

- src/Doc_NB.h

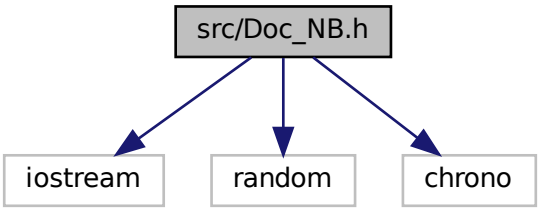
Глава 6

Файлы

6.1 Файл src/Doc_NB.h

Моделирование отрицательно-биномиального распределения.

#include <iostream>
#include <random>
#include <chrono>
Граф включаемых заголовочных файлов для Doc_NB.h:



Классы

- class NB_distr
  Класс отрицательно-биномиального распределения.
- class Sample
  Класс моделирования распределений.
- class Sample_Table
  Класс моделирования распределения табличным методом.
- class Sample_Bernulli
  Класс моделирования распределения методом Бернулли.
- class ChiSqHist
  Класс критерия согласия.
- class Doc_NB
  Класс моделирования и гипотез.

Переменные

- unsigned int seed

  Инициация генератора случайных чисел.
- std::default_random_engine generator

  Генератор псевдослучайных чисел.

6.1.1 Подробное описание

Моделирование отрицательно-биномиального распределения.

файл содержит классы, используемые для моделирования выбоки отрицательно-биномиального распределения. Они позволяют получать выборки, изменять параметры распределений, выбирать метод моделирования (табличный или Бернулли); вычислять эмперические частоты и теоретические вероятности, критерий согласия χ^2 и p-value на его основе; строить выборки из p-value.

Глава 7

Примеры

7.1 main.cpp

```
#include <iostream>
#include "../Doc_NB.h"
int main(int argc, char** argv){
    // Инициализация распределений по умолчанию и с параметрами.
    NB_distr d0, d1(0.84, 11);
    // Название распределения.
    std::cout « d0.name_of_distr() « "\n";
    // Инициализация метода моделирования Бернулли.
    Sample_Bernulli sam_ber(100, &d0);
    // Моделирование выборки.
    sam_ber.simulate();
    // Название метода.
    std::cout « sam_ber.get_name() « ": ";
    // Выборка.
    for (size_t i = 0; i < sam_ber.get_n(); ++i)
        std::cout « sam_ber[i] « " ";

    std::cout « "\n";
    // Изменение размера выборки.
    sam_ber.change_param(10);
    sam_ber.simulate();
    std::cout « sam_ber.get_name() « ": ";
    for (size_t i = 0; i < sam_ber.get_n(); ++i)
        std::cout « sam_ber[i] « " ";

    std::cout « "\n";
    // Инициализация табличного метода моделирования.
    Sample_Table sam_table(70, &d0);
    sam_table.simulate();
    std::cout « sam_table.get_name() « ": ";
    for (size_t i = 0; i < sam_table.get_n(); ++i)
        std::cout « sam_table[i] « " ";

    std::cout « "\n";
    // Инициализация класса моделирования выборок p-value.
    Doc_NB doc;
    // Изменение параметров класса моделирования выборок p-value.
    doc.change_param(d0, d1, 100, 120, 0.1);
    // Установка в качестве распределения альтернативной гипотезы.
    doc.set_hyp_d1();
    // Установка в качестве метода моделирования метода Бернулли.
    doc.set_bernulli_method();
    // Моделирование выборки p-value.
    doc.make_p_value();
    // Получение доступа к классу хи квадрат.
    ChiSqHist* chi_sq = doc.get_chi_sq();
    // Вывод вычисленного p-value.
    std::cout « "P-value: " « chi_sq->get_p_value() « "\n";
    // Вычисления таблицы теоретичесикх вероятностей.
    chi_sq->calc_th_freq();
    std::cout « "Теоретические вероятности: ";
    // Вывод теоретических вероятностей.
    for (size_t i = 0; i < chi_sq->get_num_freq(); ++i)
        std::cout « chi_sq->get_th_freq()[i] « " ";

    std::cout « "\n";
    // Вычисление таблицы эмперичесикх частот.
    chi_sq->calc_exp_freq();
```

```
    std::cout « "Эмперические частоты: ";
    // Вывод эмперических частот.
    for (size_t i = 0; i < chi_sq->get_num_freq(); ++i)
        std::cout « chi_sq->get_exp_freq()[i] « " ";

    std::cout « "\n";
    std::cout « "Выборка p-value: ";
    // Вывод выборки p-value.
    for (size_t i = 0; i < doc.get_num_p_value(); ++i)
        std::cout « doc.get_p_value(i) « " ";
    std::cout « "\n";
    // Установка в качестве распределения нулевой гипотезы.
    doc.set_hyp_d0();
    // Установка в качестве метода моделирования табличного метода.
    doc.set_table_method();
    doc.make_p_value();
    std::cout « "Выборка p-value: ";
    for (size_t i = 0; i < doc.get_num_p_value(); ++i)
        std::cout « doc.get_p_value(i) « " ";
    std::cout « "\n";
    return 0;
}
```

Предметный указатель

change_param
    Doc_NB, 15
    Sample, 24
    Sample_Table, 33
ChiSqHist, 9
    ChiSqHist, 10
    get_exp_freq, 11
    get_num_freq, 11
    get_p_value, 11
    get_th_freq, 12
    operator=, 12
    set_data, 12

Doc_NB, 13
    change_param, 15
    Doc_NB, 14
    get_chi_sq, 15
    get_d0, 16
    get_d1, 16
    get_num_p_value, 16
    get_p_value, 16
    get_Sample, 17
    get_sign_lv, 17
    operator=, 17

get_chi_sq
    Doc_NB, 15
get_d0
    Doc_NB, 16
get_d1
    Doc_NB, 16
get_exp_freq
    ChiSqHist, 11
get_k
    NB_distr, 19
get_n
    Sample, 24
get_name
    Sample, 24
    Sample_Bernulli, 28
    Sample_Table, 33
get_num_freq
    ChiSqHist, 11
get_num_p_value
    Doc_NB, 16
get_p
    NB_distr, 19
get_p_value
    ChiSqHist, 11
    Doc_NB, 16
get_prob_now
    NB_distr, 19
get_Sample
    Doc_NB, 17
get_sign_lv
    Doc_NB, 17
get_th_freq
    ChiSqHist, 12

name_of_distr
    NB_distr, 19
NB_distr, 18
    get_k, 19
    get_p, 19
    get_prob_now, 19
    name_of_distr, 19
    NB_distr, 18
    next_prob, 20
next_prob
    NB_distr, 20

operator=
    ChiSqHist, 12
    Doc_NB, 17
    Sample_Bernulli, 28
    Sample_Table, 33
operator[]
    Sample, 25

Sample, 20
    change_param, 24
    get_n, 24
    get_name, 24
    operator[], 25
    Sample, 22
    simulate_one, 25
    swap, 25
Sample_Bernulli, 26
    get_name, 28
    operator=, 28
    Sample_Bernulli, 27, 28
    simulate_one, 30
Sample_Table, 30
    change_param, 33
    get_name, 33
    operator=, 33
    Sample_Table, 32
    simulate_one, 34
set_data

ChiSqHist, 12
simulate_one
Sample, 25
Sample_Bernulli, 30
Sample_Table, 34
src/Doc_NB.h, 35
swap
Sample, 25