Содержание

Введение

Математическое ожидание

Накапливая данные, мы получили совокупность, состоящую из n исходов, пусть n_i – число исходов -го типа. Отсюда $\sum_i n_i = n$. Допустим далее, что X_i – значение исхода -го типа. Тогда среднее значение принимает вид:

$$X_{\text{ср}} = \frac{1}{n}\sum n_i x_i = \bar{X} \quad (1)$$

В данном случае среднее является некоторой величиной, полученной наблюдением. Теперь нам необходимо <u>предсказать</u> среднее значение. <u>Прогнозированное среднее</u> значение случайной величины X называется <u>математическим ожиданием</u>. Если случайная величина X может принимать значения $X_1, X_2, \dots, X_i$ с вероятностями $P_1, P_2, \dots, P_i$, то средним значением величины X, то есть математическим ожиданием (обозначают МОЖ, МО или М), будет:

$$M(\bar{X}) = \sum_{i=1}^{n} P_i X_i \quad (2)$$

Математическое ожидание является абстрактным понятием. Однако, когда n велико, среднее значение $\bar{X}$ и $M(\bar{X})$ по существу оказываются численно равными.

Рассмотрим такой пример: в цехе на 4 станках изготавливают одинаковые детали. За месяц на первом станке изготовлено 100 деталей, из них вероятность первого сорта 0,9, на втором – 150 деталей с вероятностью первого сорта 0,8, на третьем – 100 деталей с вероятностью первого сорта 0,95, и на четвертом – 120 деталей с вероятностью первого сорта 0,75. Найти $M(\bar{X})$ общего числа деталей первого сорта, изготовленных на четырех станках.

Следует сразу заметить, что изготовление деталей первого сорта есть величина случайная, следовательно искомое

$M(\bar{X}) = M(\bar{X}_1) + M(\bar{X}_2) + M(\bar{X}_3) + M(\bar{X}_4) = 100 * 0{,}9 + 150 * 0{,}8 + 100 * 0{,}95 + 120 * 0{,}75 = 395$ деталей.

Рассмотрим пример другого рода. Пусть независимые случайные величины X и Y заданы:

Таблица 1

X	10	3	6	2
P_x	0,2	0,3	0,1	0,4

Y	15	10	20
P_y	0,2	0,7	0,1

Найти $M(\bar{X})$ случайной величины XY.

Для нахождения воспользуемся (1):

$$M(\bar{X}) = 10 * 0{,}2 \ + \ 3 * 0{,}3 \ + \ 6 * 0{,}1 \ + 2 * 0{,}4 = 4{,}3$$

$$M(\bar{Y}) = 15 * 0{,}2 + \ 10 * 0{,}7 + 20 * 0{,}1 = 12$$

Так как случайные величины X и Y независимы, то искомое $M(X, Y)$

$$M(X, Y) = \ M(\bar{X}) * M(\bar{Y}) = 12 * 4{,}3 = 51{,}6$$

Дисперсия и среднее квадратическое отклонение

Среднее значение не всегда может характеризовать действительную картину учитываемых величин.

Рассмотрим две группы величин:

$$1\colon \{0{,}48;\ 0{,}49; 0{,}50; 0{,}50; 0{,}53\} \quad \bar{1} = 0{,}5$$
$$2\colon \{0{,}20;\ 0{,}30;\ 0{,}50; 0{,}60; 0{,}90\} \quad \bar{2} = 0{,}5$$

Как характеризовать данный разброс величин?

<u>Величина, называемая дисперсией</u>, используется как характеристика <u>разброса</u> или <u>ожидаемого разброса</u>. Дисперсия совокупности имеет вид:

$$D^2(X) = \frac{1}{n}\sum_i n_i(X - M)^2 \quad (3)$$

Прогнозируемая дисперсия:

$$D^2(X) = \sum_i P_i(X - M)^2 \quad (4)$$

Рассмотрим приведенные выше группы: $P_i = \frac{1}{5} = 0{,}2$

$$M(X_1) = 0{,}48 * 0{,}2 + 0{,}49 * 0{,}2 \ + \ 0{,}5 * 0{,}2 + 0{,}5 * 0{,}2 + 0{,}53 * 0{,}2 = 0{,}5$$

$$M(X_2) = 0{,}2 * 0{,}2 + 0{,}3 * 0{,}2 \ + \ 0{,}5 * 0{,}2 + 0{,}6 * 0{,}2 + 0{,}9 * 0{,}2 = 0{,}5$$

Исходов для групп:

Таблица 2

I группа		II группа	
(0,48-0,5)^2	0,0004	(0,2-0,5)^2	0,09
(0,49-0,5)^2	0,0001	(0,3-0,5)^2	0,04
(0,5-0,5)^2	0	(0,5-0,5)^2	0
(0,5-0,5)^2	0	(0,6-0,5)^2	0,01
(0,53-0,5)^2	0,0009	(0,9-0,5)^2	0,16
Итого	0,0014	Итого	0,3
D^2	0,00028	D^2	0,06

Чтобы понять смысл дисперсии, заметим, что для I группы, значения данных которой мало отличаются друг от друга, дисперсия равна 0,00028, в то же время для II группы, характеризуемой значительным разбросом, дисперсия равна 0,06. Среднее квадратическое отклонение, которое является просто квадратным корнем из дисперсии, употребляется чаще, чем дисперсия и имеет ту же размерность, что и среднее значение. В нашем примере, среднее квадратическое отклонение для I группы данных – 0,017. Это показывает, что большая часть данных лежит в интервале 0,5 ± 0,017 , тогда как для второй группы среднее квадратическое отклонение равно 0,5 ± 0,245. Часто необходимо использовать оценку числа исходов n или оценку вероятностей P для вычисления средних значений и средних квадратческих отклонений, но не менее важно уметь решать обратную задачу. Если известны средние значения и средние квадратические отклонения, то какова вероятность появления каждого i-го события?

Рассмотрим такой пример. Завод выпускает станки 4 типов стоимостью 7, 9, 11 и 15 тысяч рублей. Количество станков неизвестно, но известно, что средняя стоимость станка составляет 9 тысяч рублей. Какова оценка распределения станков по типам является лучшей (Потребность в каждом станке будем считать одинаковой)? Обратная задача такого типа имеет бесконечное множество решений, удовлетворяющих среднему значению. Какое же решение является оптимальным? Чтобы ответить на этот вопрос и решить пример, нужно напомнить одно из положений теории информации.

Неопределенность результата события возрастает с увеличением числа равновероятных исходов. Следовательно, должно возрастать количество информации в сообщении о результате. Величина, определяющая количественную меру неопределенности исхода события, называется энтропией события (H).

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i, \qquad (5)$$

где P – вероятность, i – исход события, n – число всех возможных

Если новый станок с ЧПУ дает не более 5% брака, то следовательно, вероятность изготовления годных деталей на нем $P_г = 0{,}95$, а бракованных $P_б = 0{,}05$. Энтропия этого станка

$$H = 0{,}95 * \log_2 \frac{1}{0{,}95} + 0{,}05 * \log_2 \frac{1}{0{,}05} = 0{,}29$$

Для изношенного станка появление годных и бракованных деталей становиться одинаково вероятным, неопределенность достигает наибольшего значения $P_г = P_б = 0{,}5$ и энтропия достигает максимального значения для события с двумя исходами:

$$H = 0{,}5 * \log_2 \frac{1}{0{,}5} + 0{,}5 * \log_2 \frac{1}{0{,}5} = 1$$

С помощью этой формулы можно подсчитать количество информации. Единица информации содержит в себе какое-либо законченной сообщение. Эту единицу информации называют бит – бинарная или двойная единица. Используя формулу (H) для нашего конкретного примера, наименее смещенную оценку, максимизируя функцию $H = -\sum_{i=1}^{n} P_i \log_2 P_i$, при заданных ограничениях. В данном случае ограничениями являются $\sum_i P_i = 1$ и $\sum_i P_i n = \bar{n} = 9000$. Функция H – средняя неопределенность информации о системе. Чтобы максимизировать функцию H, обычно используют множители Лагранжа. Для нахождения ее оптимального значения продифференцируем функцию H и результат приравняем нулю. Затем продифференцируем уравнения для ограничивающих условий и умножим каждое из них а множитель Лагранжа (множители Лагранжа λ и β пока еще неизвестны, при чем они различны для различных уравнений).

$$\alpha_н = \sum_i (\ln P_i + 1)\alpha P_i = 0$$

$$\lambda \sum_i \alpha P_i = 0$$

$$\beta \sum_i n_i \alpha P_i = 0$$

Суммируя эти выражения, и замечая, что $\alpha P_i = 0$, получим

$$\textstyle\sum_i (\ln P_i + \lambda + \beta n_i)\, \alpha P_i = 0,$$

где $P_i = \exp(-\lambda - \beta n_i)$

Множители Лагранжа λ и β найдем путём использования двух исходых уравнений для ограничений.

$$\lambda = \ln\left[\sum_i \exp(-\beta n_i)\right]$$

$$n = \frac{\sum_i n_i \exp(-\beta n_i)}{\sum_i \exp(-\beta n_i)}$$

Следовательно, $P_i = \exp(-\lambda - \ \beta) = \frac{\exp(-\beta n_i)}{\sum_i \exp(-\beta n_i)}$

Для нашего примера имеем:

$$\bar{n} = \frac{\sum_i n_i \exp(-\beta n_i)}{\sum_i \exp(-\beta n_i)} = \frac{7x^2 + 9x^9 + 11x^{11} + 15x^{15}}{x^7 + \ x^9 + x^{11} + \ x^{15}} = 9$$

Где $x = \exp(-\beta)$, а значения n взяты в тысячах. После преобразований

$$6x^{15} \ + 2x^{11} - 2x^7 = 0$$

Корень $x = 0$ не имеет смысла, то $6x^8 + 2x^4 - 2 = 0$, решая квадратное уравнение получим, что $x = 0{,}81177$.

Определим вероятности: $P_i = \frac{x^{n_i}}{\sum_i x^{n_i}}$

$$P_1 = \frac{0{,}809^7}{0{,}809^7 + 0{,}809^9 + 0{,}809^{11} + 0{,}809^{15}} = 0{,}438$$

$$P_2 = 0{,}289, \qquad P_3 = 0{,}19, \qquad P_4 = 0{,}083$$

Таким образом, наилучшей оценкой, которую можно сделать на основании имеющейся информации, является следующая: 44% станков стоят по 7 тысяч рублей, 29% - по 9, 19% - по 11 и 8% - по 15.

Корреляция

Огромная ценность теории вероятности и математической статистики заключается в том, что они позволяют выразить количественно неопределенность или субъективные ощущения. На диаграмме 1 изображены два графика. На глаз кажется, что величины *u* и *v* связаны между собой в большей степени, чем величины *x* и *y*. Часто полезно иметь количественный показатель так называемой «корреляции». Для этой цели в математической статичтике введен параметр, называемый коэффициентом корреляции *r*. Его выбирают таким образом, чтобы в случае, когда зависимость между рассматриваемыми переменными выражается прямой линией, значение *r* было равно 1. Когда эти величины совершенно случайны, коэффициент корреляции равен 0.

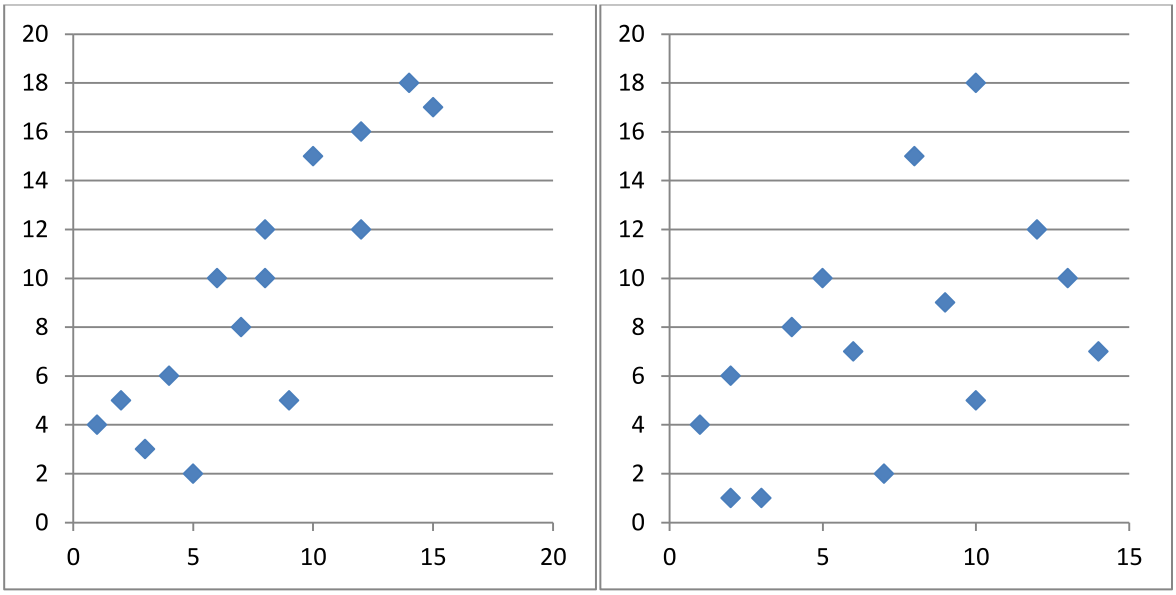

Диаграмма 1

Чтобы получить коэффициент *r* между двумя случайными величинами, необходимо для каждой из них вычислить среднее значение и среднее квадратическое отклонение. Для величин *u* и *v* в таблице 3 среднее значение и среднее квадратическое отклонение рассчитываются по формулам:

$$\bar{u} = \frac{1}{n}\sum_i^n u_i \quad \sigma_u = \sqrt{\frac{\sum(u_i-\bar{u})^2}{n-1}} \quad (6)\ (7)$$

Таблица 3

1	4	1	4
5	2	2	1
7	8	2	6
2	5	4	8
9	5	3	1
8	10	5	10
8	12	7	2
10	15	8	15
12	12	10	18
12	16	10	5
15	17	12	12
4	6	14	7
3	3	9	9
6	10	13	10
14	18	6	7
u = 7,73	v = 9,53	x = 7,07	y = 7,67

В данном случае коэффициент корреляции равен:

$$r = \frac{\sum_i^n (u_i - \bar{u})(v_i - \bar{v})}{(n-1)\sigma_u \sigma_v} \quad (8)$$

Для данных значений r =0.84, что указывает на наличие сильной корреляции, тогда как для второй серии данных коэффициент корреляции составил 0.47. Однако к данному показателю следует относиться очень осторожно. Если между двумя величинами существует корреляция, то это еще не означает , что они связаны друг с другом как причина и следствие.

Практическая часть

Дисперсия и среднее квадратическое отклонение

Для закрепления знаний по понятиям математической статистике и заданием было выбрана разработка программы для обработки результатов измерений и подсчетов математических ожиданий, дисперсии и среднего квадратического отклонения. Программа была реализована на языке программирования С++ и имеет следующий функционал:

- Ввод результатов измерений через консоль;
- Ввод результатов измерений из оформленного файла input.txt;
- Подсчет математического ожидания для введённых данных для количества исходов и их значений;
- Подсчет математического ожидания для введенных данных для вероятности исхода и его значения;
- Подсчет дисперсии системы для разных вариантов данных;
- Вывод переданных данных, вместе с их обработкой, математическим ожиданием, дисперсией и средним квадратическим отклонением.

Рассмотрим заголовочный файл программы lib.h:

```
#ifndef LIB_H
#define LIB_H

class base {
public:
  void input();
  void finput();
  void me_count();
  void ma_count();
  void de_count();
  void da_count();
  void output();

private:
  double *data[3];
  int size;
  double average;
  double dis;
};

class application : public base {
public:
  void init();
  void exec();
```

```
private:
  bool choice;
};

#endif
```

В рассматриваемом файле описывается базовый класс base, который хранит массив из трех ссылок на тип double, которые будут использоваться для хранения и обработки информации, переменную size, хранящий количество исходов, average хранит математическое ожидание для хранящихся данных, dis хранит дисперсию хранящихся данных. В классе base описаны следующие методы:

- input() производит чтение данных из консоли;
- finput() производит чтение данных из файла input.txt;
- me_count() производит подсчет математического ожидания величины, при заданных параметрах вероятности исхода и его величины;
- ma_count() производит подсчет математического ожидания величины, при известном количестве исходов разных типов и их величин;
- de_count() рассчитывает дисперсию при данных вероятностях;
- da_count() рассчитывает дисперсию при данных о количестве исходов разных типов;
- output() производит вывод в консоль и файл output.txt о рассчитанных данных о квадратичном отклонении каждого исхода, о математическом ожидании для данных сведений, дисперсии и среднем квадратичном отклонении.

  Класс application является наследником класса base и имеет одно свойство choice, хранящее в себе значение выбора обработки данных и два метода:
- init(), который заполняет объект данными;
- exec(), который выполняет подсчеты и выводит данные.

Внизу представлен файл реализации lib.cpp:

```
#include "lib.h"
#include <cmath>
#include <fstream>
#include <iostream>
using namespace std;
void base::input() {
  cout << "Введите количество пар значений" << endl;
  cin >> size;
  data[0] = new double[size];
  data[1] = new double[size];
  data[2] = new double[size];
```

```
  cout << "Введите " << size << " пар элементов через пробел" << endl;
  for (int i = 1; i <= size; i++) {
    cout << "Введите " << i << " пару элементов ";
    cin >> data[i - 1][0] >> data[i - 1][1];
  }
}
void base::finput() {
  fstream file("input.txt", ios::in);
  file >> size;
  data[0] = new double[size];
  data[1] = new double[size];
  data[2] = new double[size];
  for (int i = 0; i < size; i++) {
    file >> data[0][i] >> data[1][i];
  }
  file.close();
}
void base::ma_count() {
  average = 0.0;
  int n = 0;
  for (int i = 0; i < size; i++) {
    average += data[0][i] * data[1][i];
    n += (int)data[0][i];
  }
  average /= n;
}
void base::me_count() {
  average = 0.0;
  for (int i = 0; i < size; i++) {
    average += data[0][i] * data[1][i];
  }
}
void base::da_count() {
  dis = 0.0;
  int n = 0;
  ma_count();
  for (int i = 0; i < size; i++) {
    data[2][i] = data[0][i] * (data[1][i] - average) * (data[1][i] - average);
    dis += data[2][i];
    n += data[0][i];
  }
  dis /= n;
}
void base::de_count() {
  dis = 0.0;
```

```
    me_count();
    for (int i = 0; i < size; i++) {
      data[2][i] = data[0][i] * (data[1][i] - average) * (data[1][i] - average);
      dis += data[2][i];
    }
  }
  void base::output() {
    fstream file("output.txt", ios::out);
    for (int i = 0; i < size; i++) {
      for (int j = 0; j < 3; j++) {
        cout.width(15);
        cout << data[j][i];
        file.width(15);
        file << data[j][i];
        if (j == 2) {
          file << endl;
          cout << endl;
        }
      }
    }
    cout << "Математическое ожидание для данного множества: M(X) = " <<
average
         << endl;
    cout << "Дисперсия для данного множества: D^2 = " << dis << endl;
    cout << "Среднеквадратическое отклонение: sqrt(D^2) = " << sqrt(dis);
    file << "Математическое ожидание для данного множества: M(X) = " <<
average
         << endl;
    file << "Дисперсия для данного множества: D^2 = " << dis << endl;
    file << "Среднеквадратическое отклонение: sqrt(D^2) = " << sqrt(dis);
    file.close();
  }
  void application::init() {
    cout << "Хотите вводить данные вручную или из файла (0 и 1,
соответственно)"
         << endl;
    int ch;
    cin >> ch;
    if (ch) {
      finput();
    } else {
      input();
    }
  }
  void application::exec() {
```

```
    cout << "Хотите посчитать для количества исходов или их вероятностей (0 и 1 соотвественно)"
         << endl;
    cin >> choice;
    if (choice) {
      de_count();
    } else {
      da_count();
    }
    output();
  }
```

Ниже представлен файл основной программы main.cpp:

```
#include "lib.h"
#include <iostream>

int main() {
  application app;
  app.init();
  app.exec();
  return 0;
}
```

Для тестирования данной программы были использованы следующие данные input.txt:

```
5
1 0.48
1 0.49
1 0.50
1 0.50
1 0.53
```

В результате работы программы в файле output.txt было следующее:

1	0.48	0.0004
1	0.49	0.0001
1	0.5	0
1	0.5	0
1	0.53	0.0009

Математическое ожидание для данного множества: M(X) = 0.5

Дисперсия для данного множества: D^2 = 0.00028

Среднеквадратическое отклонение: sqrt(D^2) = 0.0167332

Возвращаясь к примеру из теоретического введения, убеждаемся, что программа действительно работает правильно.

Для второго теста использовались значения вероятностей исходов, содержание input.txt:

5
0.2 0.2
0.2 0.3
0.2 0.5
0.2 0.6
0.2 0.9

В результате работы программы в файле output.txt было следующее:

0.2	0.2	0.018
0.2	0.3	0.008
0.2	0.5	0
0.2	0.6	0.002
0.2	0.9	0.032

Математическое ожидание для данного множества: M(X) = 0.5

Дисперсия для данного множества: D^2 = 0.06

Среднеквадратическое отклонение: sqrt(D^2) = 0.244949

Сопоставляя с примером из теоретического введения, убеждаемся в корректной работе программы, для пользователя приложена инструкция в приложении 1.

Корреляция

Следующая программа, реализованная на языке программирования C++, обладает следующей функциональностью:

- Ввод информации из текстового файла input.txt;
- Вывод в текстовый файл output.txt;
- Подсчет среднего квадратического отклонения и среднего значение, а также коэффициента корреляции.

Рассмотрим заголовочный файл программы lib.h:

```
#ifndef LIB_H
#define LIB_H

class data_base {
private:
  double **p_double;
  double adis, bdis, corel;
  double aver_a = 0.0, aver_b = 0.0;
  int n;

public:
  bool ready = true;
  void input();
  void count();
  void output();
};

class application : public data_base {
public:
  void init();
  void exec();
```

```
};
#endif
```

Как видно из заголовочного файла программа реализована в объектно-ориентированном стиле, при этом имеется базовый класс data_base, со свойствами double **p_double, с помощью которого будет выделяться память и осуществляться доступ к вводимым данным, adis и bdis, хранящие средние квадратические отклонения, aver_a и aver_b, хранящие средние значения, целочисленной n, хранящее размер вводимых данных и булевое значение ready, со следующими методами:

- input(), осуществляющий ввод из файла input.txt;
- count(), осуществляющий все необходимые расчёты;
- output(), реализующий вывод данных из программы в файл output.txt;

Реализован также дочерний класс application с методами:

- init(), обрабатывающий ввод данных;
- exec(), который инициализирует подсчеты и вывод данных в файл.

Внизу представлен файл реализации lib.cpp:

```
#include "lib.h"
#include <cmath>
#include <fstream>
#include <iostream>
using namespace std;

void application::init() {
  cout << "Готов ли файл input.txt к обработке? Если нет введите 0, если да, "
          "то 1"
       << endl;
  int choice;
```

```
  cin >> choice;
  if (choice) {
    input();
  } else {
    ready = 0;
    return;
  }
}
void application::exec() {
  if (ready) {
    count();
    output();
  }
}

void data_base::input() {
  fstream file;
  file.open("input.txt", ios::in);
  file >> n;
  p_double = new double *[2];
  p_double[0] = new double[n];
  p_double[1] = new double[n];
  for (int i = 0; i < n; i++) {
    file >> p_double[0][i] >> p_double[1][i];
  }
  file.close();
}
void data_base::count() {
  double summ = 0.0;
  for (int i = 0; i < n; i++) {
```

```
    aver_a += p_double[0][i];
    aver_b += p_double[1][i];
  }
  aver_a /= n;
  aver_b /= n;
  double a, b;
  for (int i = 0; i < n; i++) {
    a = p_double[0][i] - aver_a;
    b = p_double[1][i] - aver_b;
    summ += a * b;
    adis += a * a;
    bdis += b * b;
  }
  adis = sqrt(adis / (n - 1));
  bdis = sqrt(bdis / (n - 1));
  summ = summ / (n - 1);
  corel = summ / (adis * bdis);
}
void data_base::output() {
  fstream file;
  file.open("output.txt", ios::out);
  for (int i = 0; i < n; i++) {
    file.width(15);
    file << p_double[0][i];
    file.width(15);
    file << p_double[1][i] << endl;
  }
  file << "Среднеквадратическое отклонение для первой величины: " <<
adis
      << endl;
```

```
  file << "Среднее для первой величины: " << aver_a << endl;
  file << "Среднеквадратическое отклонение для второй величины: " <<
bdis
       << endl;
  file << "Среднее для второй величины: " << aver_b << endl;
  file << "Коэффициент корреляции: " << corel << endl;
  file.close();
  delete[] p_double[0];
  delete[] p_double[1];
  delete[] p_double;
}
```

Далее предоставлен файл prog.cpp:

```
#include "lib.h"
#include <iostream>
using namespace std;

int main() {
  application app;
  app.init();
  app.exec();
  return 0;
}
```

Программа была испытана на компьютере с операционной системой Ubuntu 18.04 с набором компиляции GCC. В файле input.txt находились следующие данные:

```
15
1	4
5	2
7	8
2	5
9	5
```

8	10
8	12
10	15
12	12
12	16
15	17
4	6
3	3
6	10
14	18

В результате работы программы в файле output.txt оказалось следующее:

1	4
5	2
7	8
2	5
9	5
8	10
8	12
10	15
12	12
12	16
15	17
4	6
3	3
6	10
14	18

Среднеквадратическое отклонение для первой величины: 4.31719

Среднее для первой величины: 7.73333

Среднеквадратическое отклонение для второй величины: 5.33006

Среднее для второй величины: 9.53333

Коэффициент корреляции: 0.872673

Обращаясь к разобранному во введении примеру можем убедиться в корректности работы программы, инструкция пользователя находиться в приложении 2.

Вывод

Изучил такие понятия математической статистики как математическое ожидание, дисперсия, среднее квадратическое отклонение, энтропия системы и процесса, ознакомился с поиском оптимального решения задачи на распределение путем множителей Лагранжа, изучил понятие корреляции и его приложение к количественному описанию неопределенности. Закрепил полученные знания путем написания программы для обработки данных и соответствующего расчета математического ожидания, дисперсии и среднего квадратического значения, а также реализацией программы для подсчета корреляции двух параметров.

Список литературы

1. **Михайлович, Тарасевич Олег.** Методы оптимизации и анализ процессов в иехнологии машиностроения. Москва : Редакционно-издательский отдел, 1976 г.

2. [В Интернете] https://ru.wikipedia.org/wiki/Среднеквадратическое_отклонение.

Приложение 1

Инструкция пользователя программы

Программа обрабатывает пары пары значений, и требует изначально выбрать пользователя предпочитаемый способ ввода значений.

В случае выбора ручного ввода, программа попросит изначально ввести вариант обработки данных, либо для подсчет математического ожидания для введённых данных для количества исходов и их значений, либо для вероятности исхода и его значения. После выбора программа попросит ввести одним числом количество пар значений и после необходимо вводить по два значения через пробел, окончанием ввода одной пары является переход на новую строку.

После окончания ввода программа выведет на консоль и в файл output.txt результаты своих вычислений, в первых двух столбцах находятся введенные данные, в третьем их квадратическое отклонение соответствующей пары величин, ниже приведены такие данные как математическое ожидание, дисперсия и квадратическое отклонение для введенных данных.

В случае выбора ввода из файла, предварительно должен быть оформлен файл с именем input.txt, в котором первой строкой должно идти количество пар элементов, в последующих - пары элементов, разделенных пробелом или табуляцией. Далее пользователя попросят ввести формат обработки данных, либо для подсчет математического ожидания для введённых данных для количества исходов и их значений, либо для вероятности исхода и его значения.

После окончания ввода программа выведет на консоль и в файл output.txt результаты своих вычислений, в первых двух столбцах находятся введенные данные, в третьем их квадратическое отклонение соответствующей пары величин, ниже приведены такие данные как математическое ожидание, дисперсия и квадратическое отклонение для введенных данных.

Приложение 2

Инструкция пользователя программы

Программа обрабатывает пары значений из файла input.txt, в котором в первой строке должно стоять количество пар значений, в последующих – пары значений, разделенных пробелом.

Программа спросит пользователя о готовности файла input.txt, и в случае положительного ответа, произведет обработку данных и их вывод в файл output.txt, в котором первые два столбца отображают введенные данные, а последующие строки перечисляют подсчитанные для каждого из столбцов данных среднее квадратическое отклонение и среднее значение, последней строкой выводится линейный коэффициент корреляции.