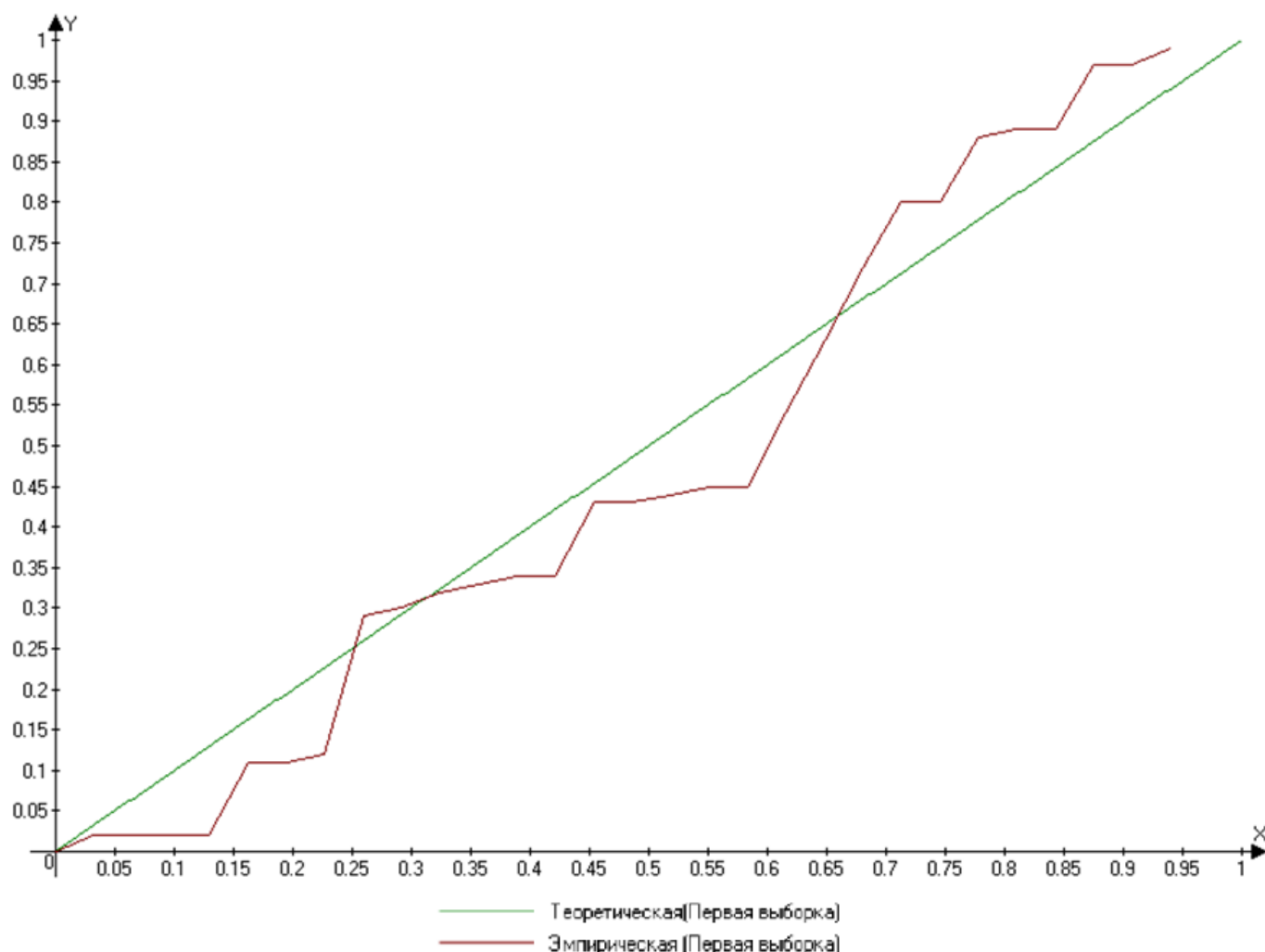


1. Цель задания

Реализовать указанные в заданиях программы, по полученным результатам построить графики теоретических и практических результатов. Оценить выборочное среднее, выборочную дисперсию, исправленную выборочную дисперсию и медиану полученных выборок.

2. Реализация выполнения задания №1

	Выборка №1	Выборка №2	Ожидаемое значение
Выборочное среднее	0,511420	0,534500	0,5
Выборочная дисперсия	0,060189	0,070338	0,083333
Исправленная дисперсия	0,060189	0,070338	
Медиана	0,593000	0,539000	0,5



Выполнил: студент группы 23656/4

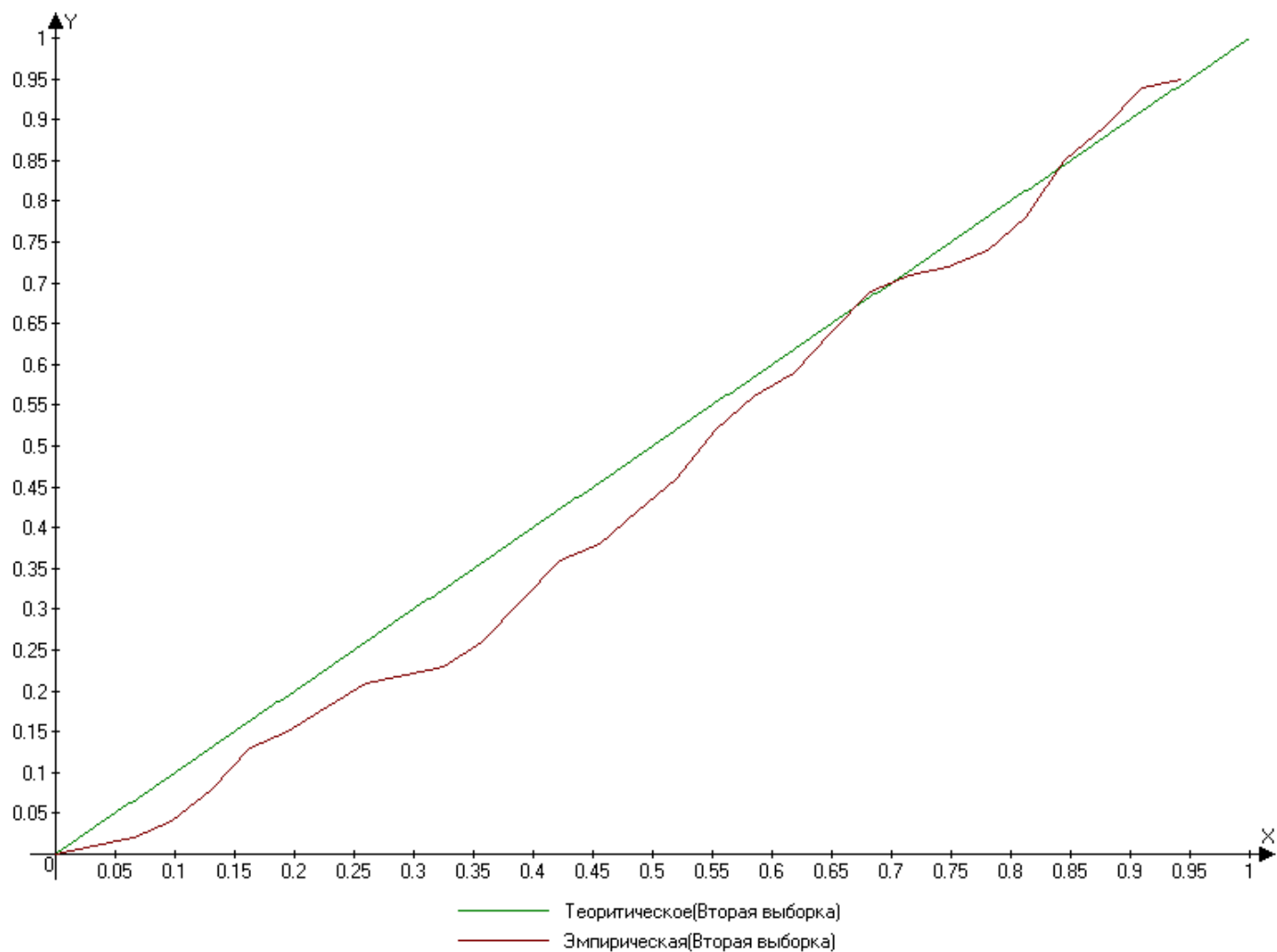
Э.А. Гасанов

(подпись, дата)

Проверил: доцент кафедры, к.т.н.

Д.С. Лаврова

(подпись, дата)



Вывод по заданию №1: Вторая выборка лучше моделирует равномерное распределение

3. Реализация выполнения задания №2

m	μ_m , выборка 1	$\frac{\mu_m}{m}$	μ_m , выборка 2	$\frac{\mu_m}{m}$
10	9	0,9	7	0.7
20	16	0,8	14	0.7
30	22	0,733333	23	0.766667
40	28	0,700000	29	0.725000
50	34	0,680000	39	0.780000
60	40	0,666667	46	0.766667
70	46	0,657143	52	0.742857
80	52	0,650000	56	0.700000
90	58	0,644444	61	0.677778
100	64	0,640000	64	0.640000

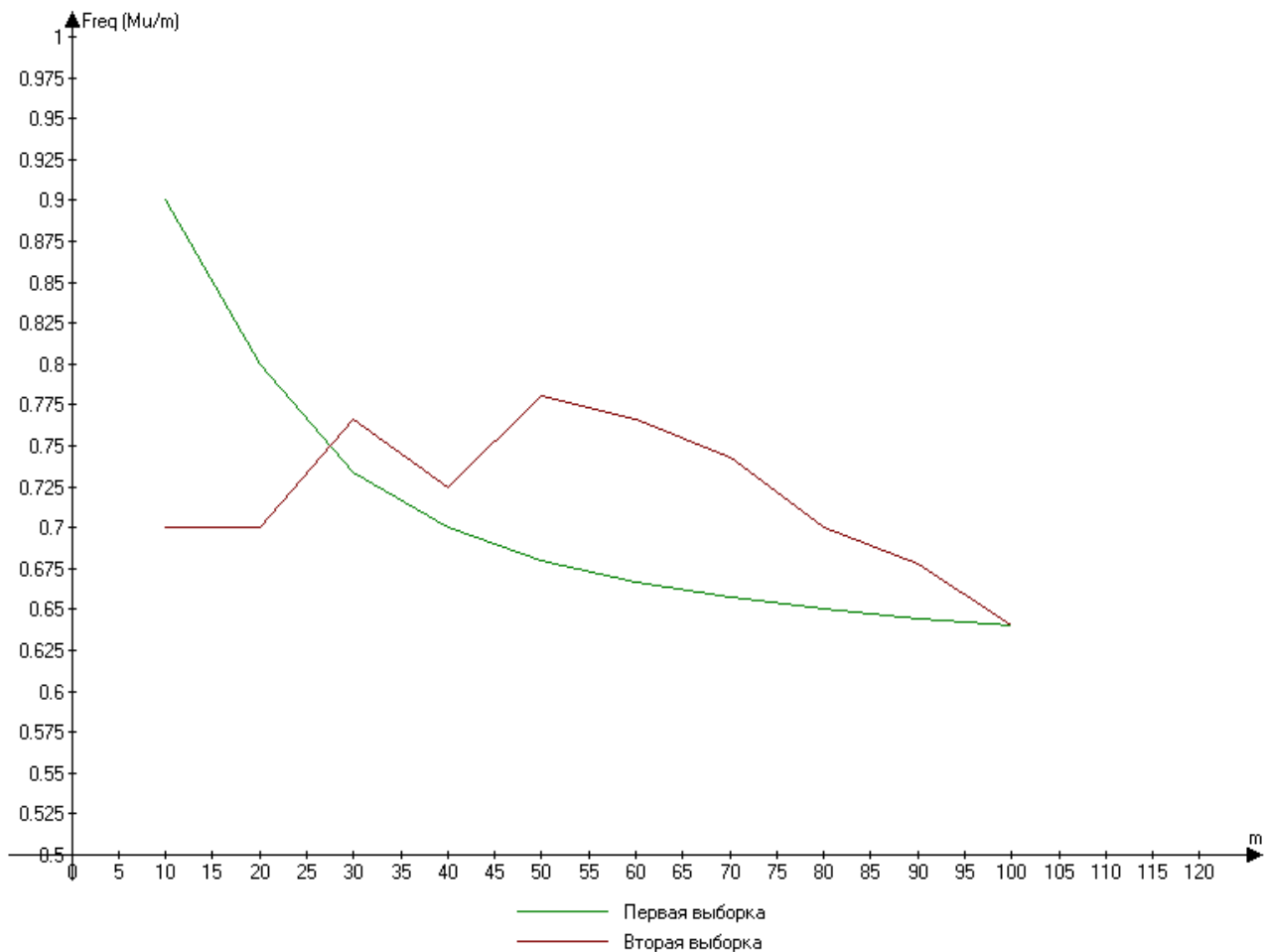


График зависимости частот от количества испытаний

Вывод по заданию №2: Частоты стремятся к заданному значению с ростом выборки.

Реализация выполнения задания №3

Случайную величину с показательным (экспоненциальным) законом распределения можно сформировать путем преобразования в $y = -\frac{1}{\lambda} \ln x$. [http://sernam.ru/book_dm.php?id=6]. Где лямбда-это заданное вариантом μ или k . А x – равномерно распределенная случайная величина.

Выборочное среднее:

$$x_{\text{выб.сред.}} = \frac{1}{n} \sum_{i=1}^n x_i$$

Выборочная дисперсия:

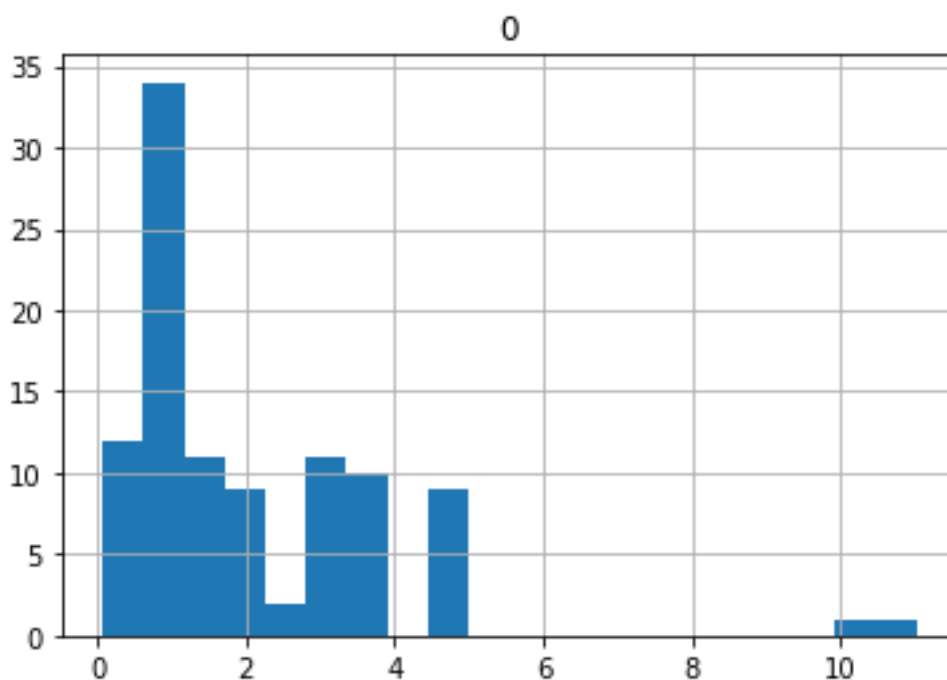
$$S^2 = \frac{1}{n} \sum_{i=1}^n (x_i - x_{\text{выб.сред.}})^2$$

Исправленная дисперсия:

$$\frac{n}{n-1} S^2$$

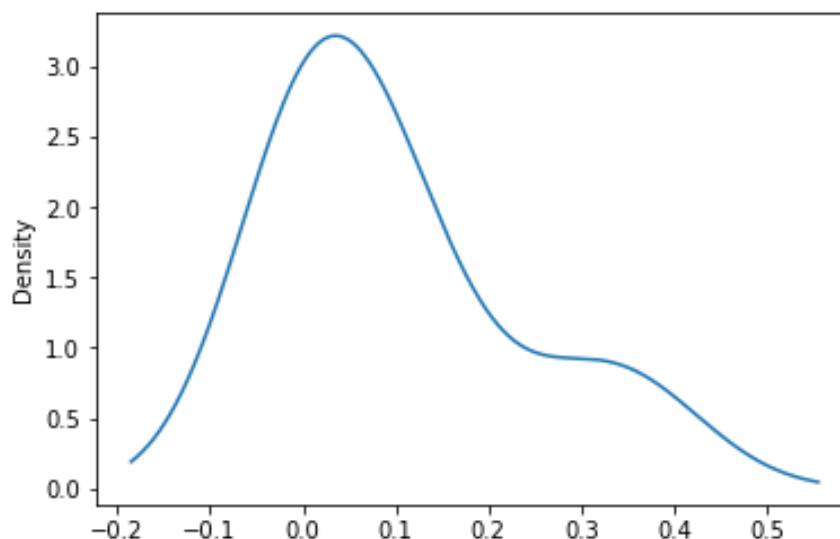
	Расчеты для $a_1 = 0.418$			Расчеты для $a_2 = 3.18$		
	Выборка №1	Выборка №2	Ожидаемое значение	Выборка №1	Выборка №2	Ожидаемое значение
Выборочное среднее	2.067921	1.966335	2,392344	0.271821	0.258468	0,314465
Выборочная дисперсия	3.386729	3.048455	5,723312	0.058517	0.052672	0,098888
Исправленная дисперсия	3.386729	3.048455		0.058517	0.052672	
Медиана	1.250146	1.478630	1,658247	0.164327	0.194361	0,217971

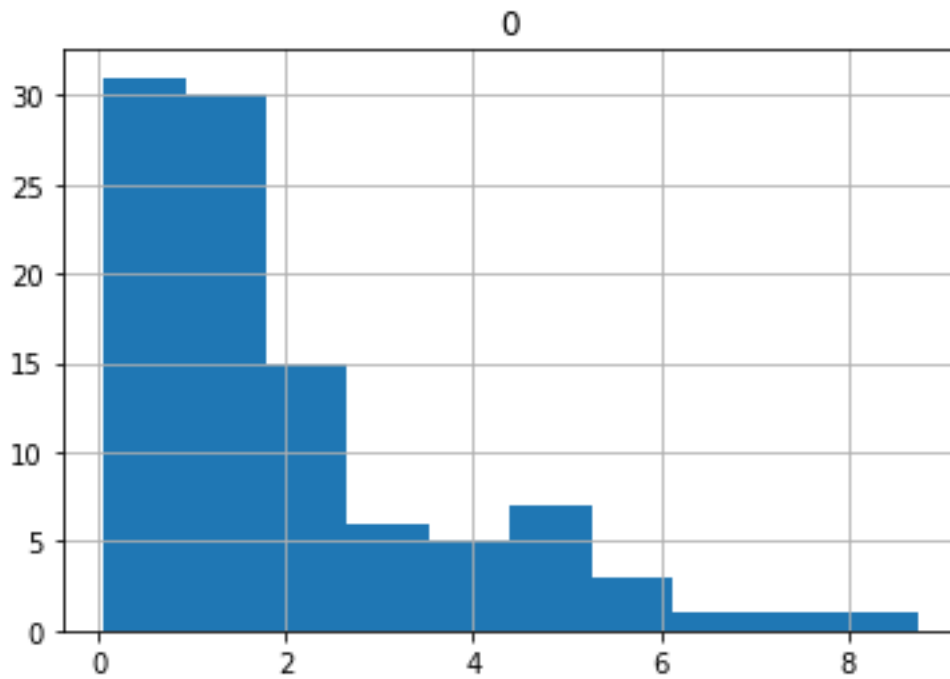
Теоретическая показательная функция $1 - e^{-\lambda x}$



Гистограмма. Первая выборка. $a_1=0.418$

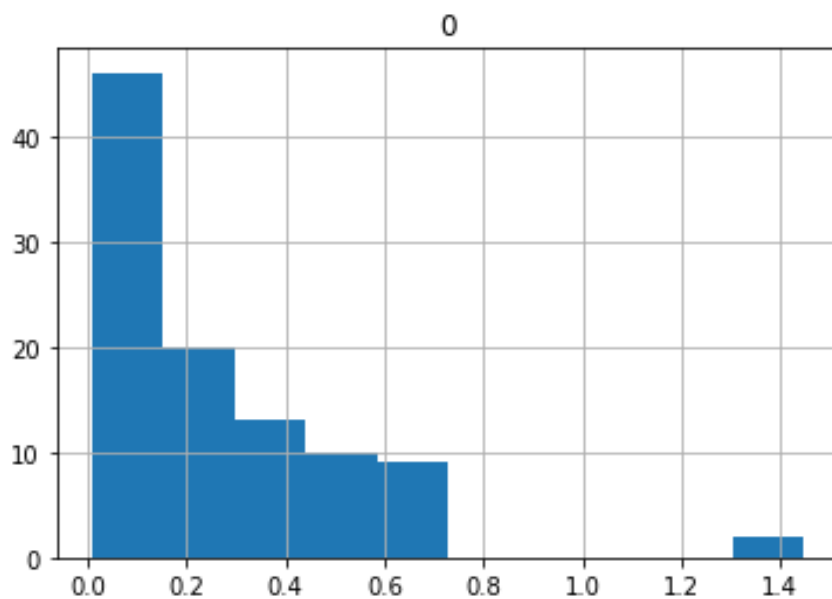
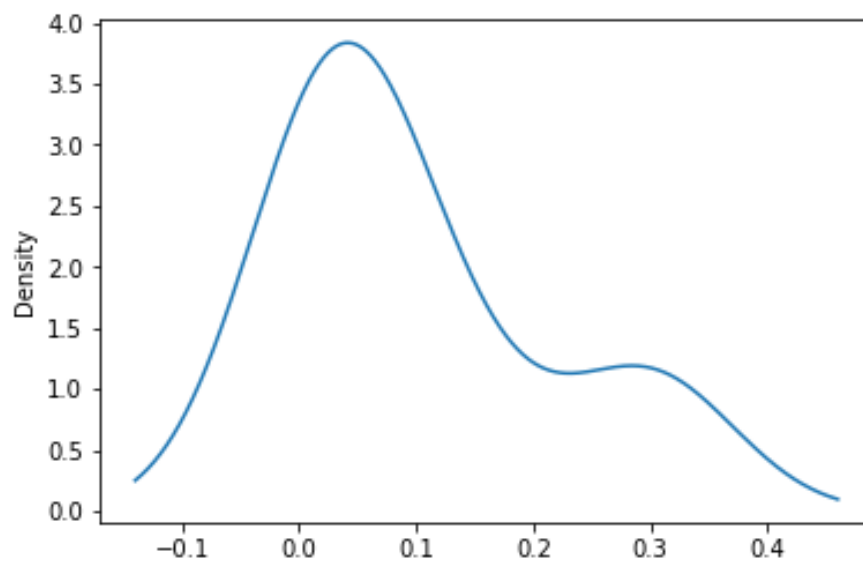
Сравним с плотностью вероятностей с этими же данными





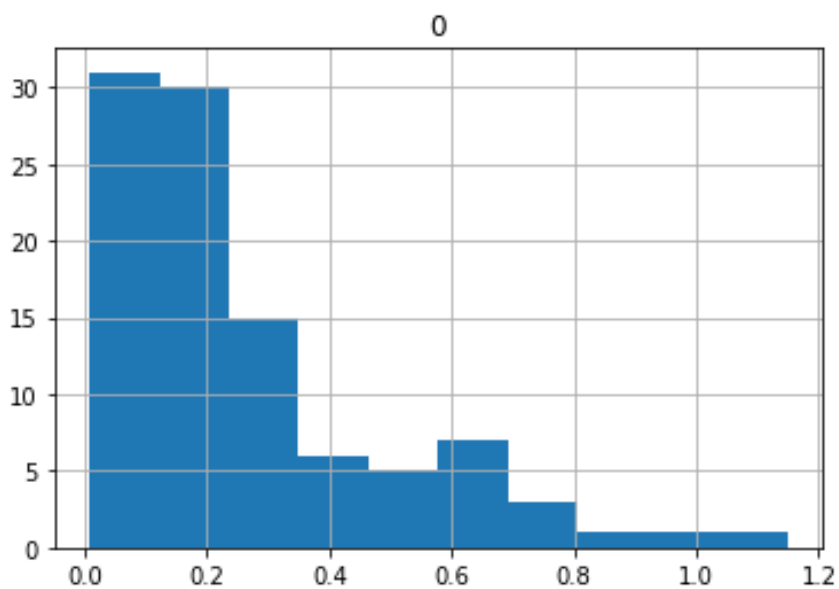
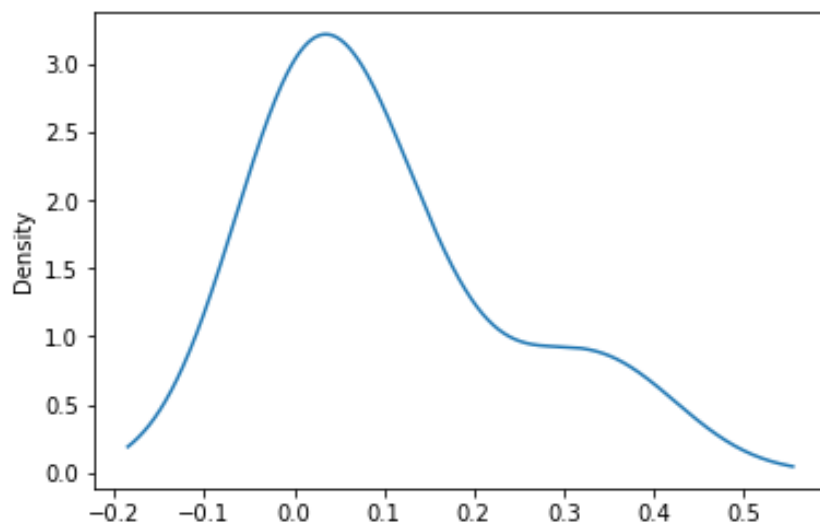
Гистограмма. Вторая выборка. $A1=0.418$

Сравним с плотностью вероятностей с этими же данными

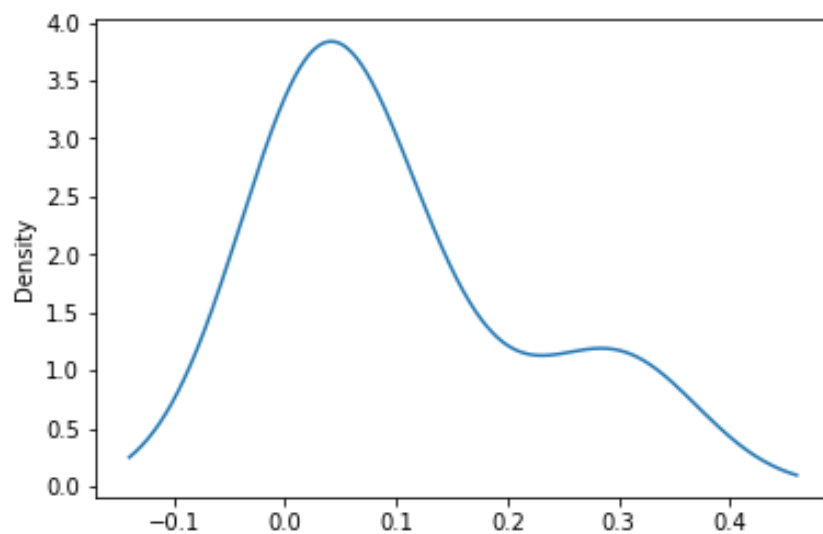


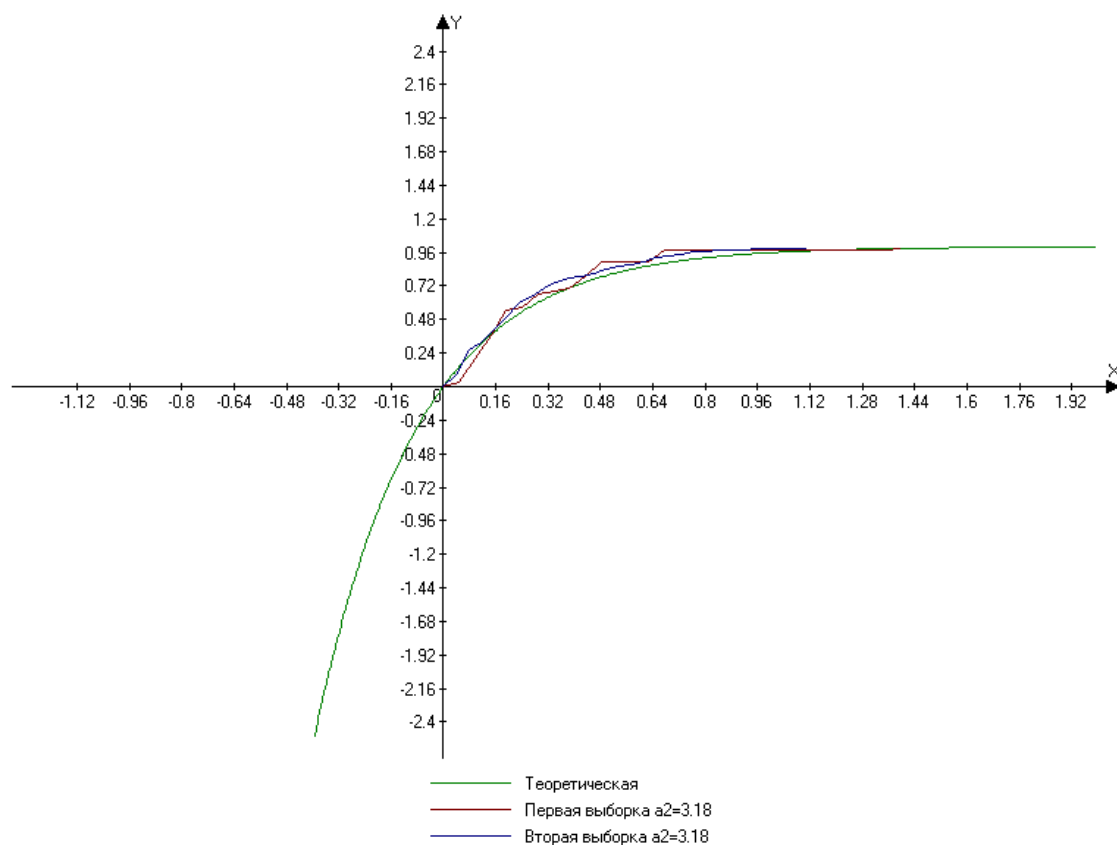
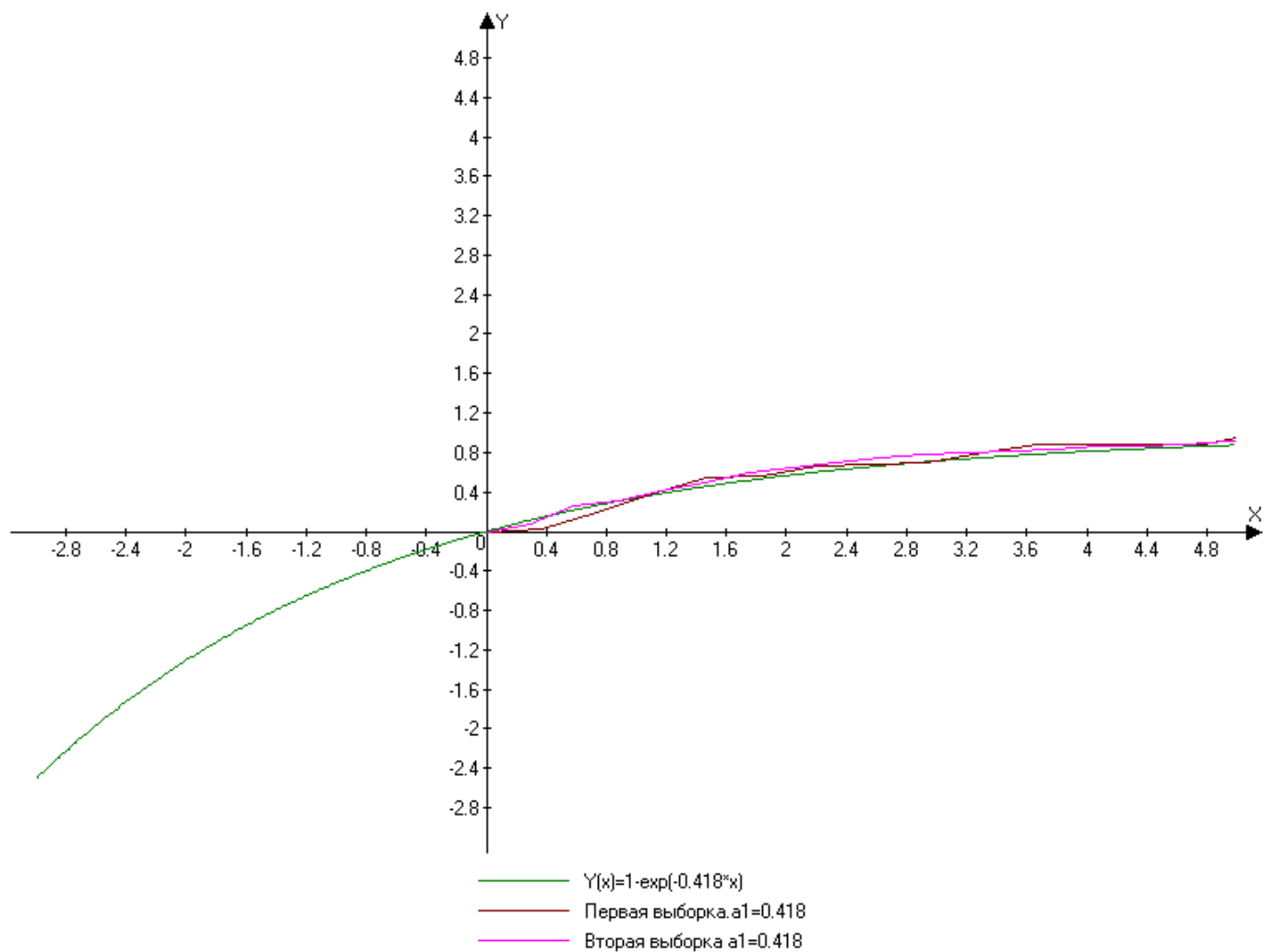
Гистограмма. Первая выборка. $a2=3.18$

Сравним с плотностью вероятностей с этими же данными



Гистограмма. Вторая выборка. $a_2=3.18$
Сравним с плотностью вероятностей с этими же данными

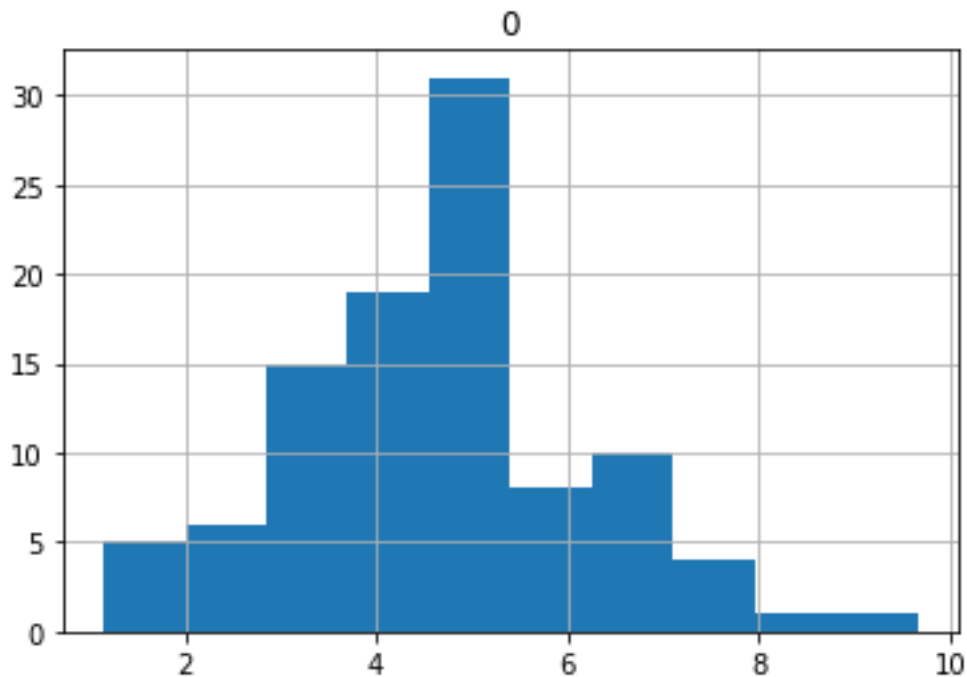




Вывод по заданию №3: параметр $a_1 = 3.18$ и вторая выборка позволяют получить результаты более близкие к своим теоретическим значениям.

4. Реализация выполнения задания №4

	Выборка №1	Выборка №2	Ожидаемое значение
Выборочное среднее	4.665995	4.814095	5.000000
Выборочная дисперсия	2.456932	4.268265	4.180000
Исправленная дисперсия	2.456932	4.268265	
Медиана	4.700015	4.364297	5.000000



Гистограмма. Первая выборка.

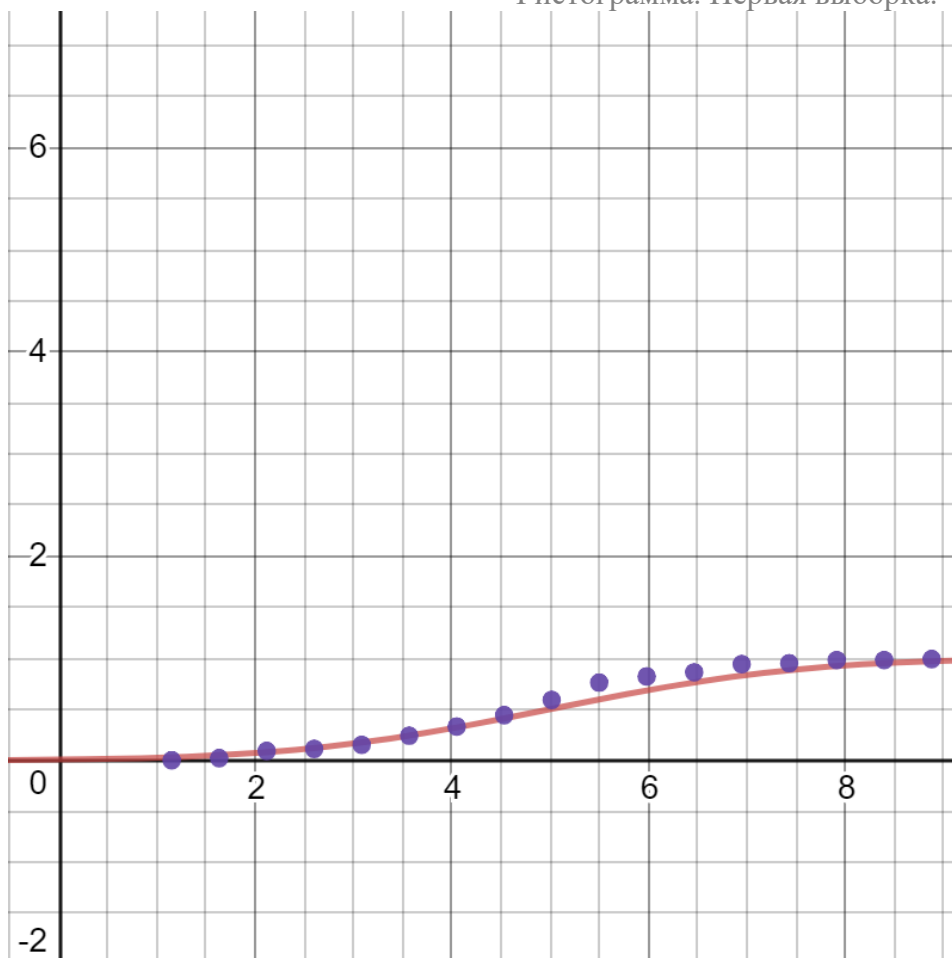
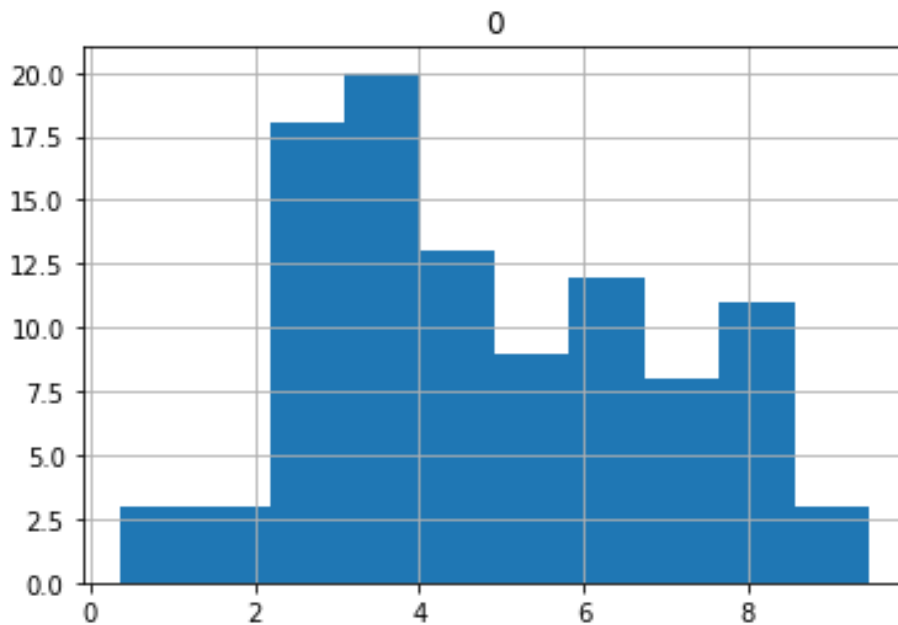


График теоретической функции нормального распределения $\frac{1}{2} \cdot \operatorname{erf}\left(\frac{(x-5)}{\sqrt{(2) \cdot 4.18}}\right) + \frac{1}{2}$ построен красным цветом. Таблично задана эмпирическая функция распределения.



Гистограмма. Вторая выборка

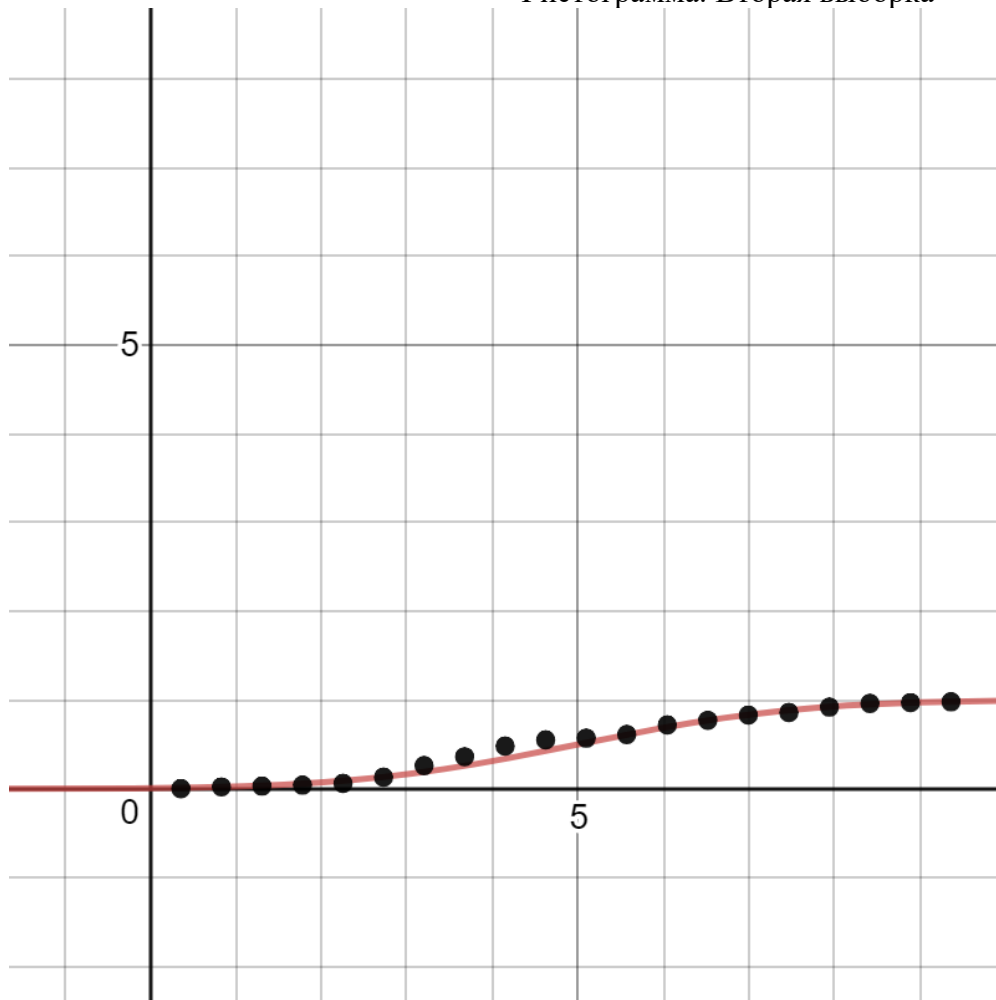


График теоретической функции нормального распределения $\frac{1}{2} \cdot \operatorname{erf}\left(\frac{(x-5)}{\sqrt{(2) \cdot 4.18}}\right) + \frac{1}{2}$ построен красным цветом. Таблично задана функция эмпирическая функция нормального распределения. Вывод по заданию №4: первая выборка наиболее близка к теоретическим статистикам нормального распределения.

5. Ответы на контрольные вопросы

1. Чем отличаются задачи теории вероятностей и математической статистики?

В теории вероятностей обычно распределение задано тем или иным образом, и требуется найти вероятности, числовые характеристики (например, математическое ожидание, дисперсию и т.п.), построить графики функции и плотности распределения, то в задачах математической статистики, напротив, известны данные (выборка), собранные по результатам какого-то эксперимента или наблюдения, по которым следует определить закон распределения, наиболее подходящий в данном случае.

2. Что такое «генеральная совокупность»?

Генеральная совокупность – это совокупность всех мысленно возможных объектов данного вида, над которыми проводятся наблюдения с целью получения конкретных значений определенной случайной величины.

3. Какие основные характеристики есть у распределения, опишите их.

- **Функция распределения.** Определена для любого вещественного распределения. Для случайной величины X ее функцией распределения называется:

$$F_X(z) = P(X \leq z), \quad -\infty < z < \infty$$

- **Плотность распределения.** Определена для непрерывных распределений. Представляет собой производную от функции распределения:

$$f_X(z) = F'_X(z), \quad -\infty < z < \infty.$$

- **Функция вероятности.** Альтернативный способ описания дискретных распределений. Если распределение случайной величины X сосредоточено в конечном или счетном числе точек $x_1, x_2, \dots, x_n, \dots$ то его можно описать вероятностями принятия случайной величиной X соответствующих значений:

Значение	x_1	x_2	\dots	x_n	\dots
Вероятность	p_1	p_2	\dots	p_n	\dots

Здесь $p_k = f(x_k) = P(X = x_k)$, $k=1, 2, \dots, n, \dots$

- **Мода** – элемент выборки, который встречается чаще всего.
- **Медиана** - значение, приходящееся на середину упорядоченного по возрастанию вариационного ряда. Если количество элементов в ряду нечётное, то медианой является $(N+1)$ -е значение, Если количество элементов в ряду чётное, то медиана равна полусумме N – го и $(N+1)$ – ого значений.
- **Математическое ожидание**-сумма произведений всех возможных её значений на соответствующие вероятности.
- **Среднеквадратичное отклонение** – корень из дисперсии.

4. Каков смысл выборочной дисперсии и выборочного среднего?

Выборочное среднее является оценкой математического ожидания случайной величины и представляет собой несмещенную оценку. Выборочная дисперсия оценивает дисперсию генеральной совокупности и является смещенной оценкой.

5. Используются ли в сфере информационных технологий рассматриваемые в задании распределения? Если да, то где? Если нет, то почему?

Генератор псевдослучайных чисел реализующий равномерное распределение, используемый в играх, например, в тетрисе, чтобы каждый раз игра начиналась не с одной и той же фигуры и чтобы цвета менялись хаотично.

6. Выводы по расчетной работе

В результате выполнения работы были получены навыки моделирования выборки для нормального, показательного, равномерного распределений и распределений Бернулли. А также строить их эмпирические функции распределений и строить на основе выборки их гистограммы.

7. Приложение

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define N 100
# define M_PI          3.14159265358979323846 /* pi */
#define partition 10
const double x0 = 0.65;

void qsortx(float *a, int low, int high) {

    int i = 0, j = 0;
    float tmp = 0, medianofthree = 0;

    i = low;
    j = high;
    float k = 0, l = 0, m = 0;
    k = *(a + 0);
    l = *(a + high);
    m = *(a + (high + low) / 2);
    /*Median-of-three*/
    if (k > l && k < m || k < l && k > m)
        medianofthree = k;
    else
        if (l > k && l < m || l < k && l > m)
            medianofthree = l;
        else
            medianofthree = m;
    /*Median-of-three*/
    //printf("\n%d\n", medianofthree);
    do {
        while (*(a + i) < medianofthree)
        {
            i++;
        }
        while (*(a + j) > medianofthree)
```

```

        {
            j--;
        }
        if (i <= j)
        {
            if (*(a + i) > *(a + j))
            {
                tmp = *(a + i);
                *(a + i) = *(a + j);
                *(a + j) = tmp;
            }
            i++;
            if (j > 0)
            {
                j--;
            }
        }
    } while (i <= j);

    if (i < high)
    {
        qsortx(a, i, high);
    }
    if (j > low)
    {
        qsortx(a, low, j);
    }
}

void show(float *x)
{
    int i = 0;
    for (i = 0; i < N; i++)
        printf("%f ", x[i]);
}

void Input(float* x)
{
    int i = 0;
    x[0] = 0.65;
    //i++;
    int t = 6;
    int a = 2;
    int K = 10 * t + a;
    for (i=1; i < N; i++)

```

```

{
    //x[i] = K * x[i - 1]-int (K * x[i - 1]);
    x[i] = (int((K * x[i - 1] - int(K * x[i - 1])) * 1000)) / 1000.0; // три знака после запятой
}
}

/*void Input1(float* Uniq, float *x,float* y)
{
    int counter = 0;
    float temp = x[0];
    int local = 0;
    for (int i = 0; i < N; i++)
    {
        //temp = x[i];
        counter=0;
        while (x[i]==temp)
        {
            counter++;
        }
    }
}*/

int LookingForUniq(float* x, float* Uniq)
{
    int i = 0;
    Uniq[0] = x[0];
    int counter = 0;
    for (i = 1; i < N; i++)
    {
        if (x[i] != x[i - 1])
        {
            counter++;
            Uniq[counter] = x[i];
        }
    }
    return counter;
}

void FillinFrequency(float* x,float* Uniq,float* y)
{
    int k = 0;

```

```

int freq = 0;
for (int i = 0; i < N; i++) //идем по x
{
    if (Uniq[k] == x[i])
    {
        freq++;
        //y[k] = freq / 100.0;
        y[k] = ((double)freq) / N;
    }
    else
    {
        //y[k] = freq / 100;
        k++;
        freq = 1;
        //y[k] = freq / 100.0;
        y[k] = ((double)freq) / N;
    }
}

}

float VyborSred(float* x)
{
    float sum = 0;
    for (int temp = 0; temp < N; temp++)
    {
        sum = sum + x[temp];
    }
    return ((1.0/N)*sum);
}

float VyborDispersia(float* x, float ChosenX)
{
    float sum = 0;
    for (int temp = 0; temp < N; temp++)
    {
        sum = sum + (x[temp]-ChosenX)*(x[temp] - ChosenX);
    }
    return ((1.0/N)*sum);
}

float FixedDispersia(float* x, float ChosenD)
{
    //ChosenD = ChosenD * ChosenD;
    float final = 0;
    final = (N / (N - 1))*ChosenD;
    //return sqrt(final);
}

```

```

        return final;
    }
void whiteinfile(float* Uniq,float* F,int n)
{
    FILE* file;
    file = fopen("C:\\Users\\Elvin\\source\\repos\\MathStat1\\result.txt", "w+t");
    for (int i = 0; i < n+1; i++)
    {

        fprintf(file, "%f ", Uniq[i]);
        fprintf(file, "\n");
    }
    fprintf(file, "\n");
    for (int i = 0; i < n+1; i++)
    {

        fprintf(file, "%f ", F[i]);
        fprintf(file, "\n");
    }
}
void saveinfile(float* x)
{
    FILE* file;
    file = fopen("C:\\Users\\Elvin\\source\\repos\\MathStat1\\1.txt", "w+t");
    for (int i = 0; i < N; i++)
    {

        fprintf(file, "%.3f ", x[i]);
        fprintf(file, "\n");
    }
    fclose(file);

}
void Input2(float* x)
{
    int i = 0;
    x[0] = 0.65;
    //i++;

    for (i = 1; i < N; i++)
    {
        x[i] = (int(((11 * x[i - 1]+M_PI) - int(11 * x[i - 1]+ M_PI)) * 1000)) / 1000.0; // три
занка после запятой
    }
}

```

```

}
double MakeSumTill(int m,float* Bernulli,int n)
{
    float locsum = 0.0;
    if (m <=n)
    {
        for (int i = 0; i <m; i++)
        {
            locsum = locsum + Bernulli[i];
        }
    }
    return locsum;
}
float freqfunc(float x1, float* Uniq, float* y1,int n1)
{
    float temploc = 0;
    int i1 = 0;
    if (x1 <= Uniq[0])
    {
        return 0;
    }
    if (x1 > Uniq[n1])
    {
        for (int i = 0; i < n1; i++)
        {
            temploc = temploc + y1[i];
        }
        return 1;
    }
    if (x1>Uniq[0] && x1<=Uniq[n1])
    {
        while (Uniq[i1]<=x1 && i1<=n1)
        {

            temploc = temploc + y1[i1];
            i1++;

        }
        return temploc;
    }
}

```



```

void saveinfilelist(float* partx4_2,float* party4_2)
{
    FILE* file;
        //printf("[%f ; %f] , %f \n", partx4_2[i], partx4_2[i + 1], party4_2[i]);// гистограмма
    file = fopen("C:\\Users\\Elvin\\.spyder\\Gist.txt", "w+t");
    for (int i = 0; i < partition; i++)
    {

        //fprintf(file, "%.3f ", [i]);
        //fprintf(file, "\n");
        fprintf(file, "[%f ; %f] , %f \n", partx4_2[i], partx4_2[i + 1], party4_2[i]);// гистограмма
    }
    fclose(file);
}

int main()
{
    float x[N];
    //int y[N];
    //Input(x);
    Input2(x);
    //show(x);

    /*for (int i = 0; i < N - 1; i++)
        y[i] = int(1000*x[i]);
    //qsortx(y, 0, N - 1);*/
    printf("In file viborka\n");
    show(x);
    saveinfile(x);

    qsortx(x, 0, N - 1); //вариационный ряд
    printf("\nvariacionnii rrd\n");
    show(x);

    float y[N];//массив частот
    float Uniq[N];//уникальные
    int n = LookingForUniq(x,Uniq);
    puts("\n");
    //show(Uniq);
    for (int i = 0; i<n + 1; i++)
        printf("%f ", Uniq[i]);

    //Input1(Uniq, x,y);

```

```

FillinFrequency(x,Uniq,y);
puts("\n");

for(int i=0;i<n+1;i++)
printf("%f ", y[i]);//массив частот

/*float temp1 = 0;
for (int i = 0; i < n+1; i++)
    temp1 = temp1 + y[i];
printf("\n %f \n", temp1);

for (int i = 0; i<n+1; i++)
    printf("%f ", y[i]);//массив частот

float F[N];
F[0] = y[0];//первый скачок
for (int i = 1; i < n+1; i++)
{
    F[i] = y[i]+F[i-1];

}
puts("\n");
for (int i = 0; i < n+1; i++)
{
    printf("%f  %f \n", Uniq[i],F[i] );//массив частот
}

whiteinfile(Uniq, F,n);*/
float countofdots = 30.0;
printf("cumulative function or func of raspredelenie\n");
for (int i = 0; i < countofdots; i++)
{

    printf("%f , %f \n", i*(Uniq[n]) / countofdots, freqfunc(i*(Uniq[n]) / countofdots, Uniq,
y, n));// первый аргумент - x

}

float ChosenX=VyborSred(x);
printf("\n%f is Vyborochnoe Srednie, theoretic 0.5",ChosenX);

float ChosenD = VyborDispersia(x,ChosenX);

```

```
printf("\n%f is Vyborochnoe Dispersia, theortic %f ", ChosenD, 1/12.0);
```

```
float FixedD = FixedDispersia(x, ChosenD);
```

```
printf("\n%f is Fixed Dispersia", FixedD);
```

```
float mediana = 0;
```

```
mediana = (x[49] + x[50]) / 2;
```

```
printf("\n%f is mediana, theoretic 0.5\n", mediana);
```

```
//медиана для
```

```
//ожидаемое
```

```
//-----
```

```
/*
```

```
2 задание
```

```
*/
```

```
printf("2 task");
```

```
float Bernulli[N];
```

```
//Input(x);
```

```
Input2(x); //вторая выборка !!!!!!!!!!!!!и для 3
```

```
for (int i = 0; i < N; i++)
```

```
{
```

```
    if (x[i] > x0)
```

```
        Bernulli[i] = 0;
```

```
    else
```

```
        Bernulli[i] = 1;
```

```
}
```

```
show(Bernulli);
```

```
double Mu[10];
```

```
//printf("\n%f\n", MakeSumTill(20, Bernulli, N));
```

```
for (int j = 0; j < 10; j++)
```

```
{
```

```
    Mu[j] = (MakeSumTill((j+1) * 10, Bernulli, N) ) / ((j+1) * 10.0);
```

```
}
```

```
puts("\n");
```

```
for (int i = 0; i < 10; i++)
```

```
{
```

```
    printf("%d  %f \n", (i+1)*10, Mu[i]); //от него построить график
```

```
}
```

```

//int amount=MakeSumTill;
//-----
//3 задание
printf("\n3 task\n");
float Expon1[N];
float u = 0.418;
float y1[N]; //массив с частотами

for (int i = 0; i < N; i++)
{
    Expon1[i] = (-1/u)*log(x[i]);
    //Expon1[i] = int((Expon1[i] - int(Expon1[i]) )* 1000) / 1000.0; // три знака
    после запятой
}

for (int i = 0; i<N; i++)
    printf("%f ", Expon1[i]);
saveinfile(Expon1);
qsortx(Expon1, 0, N - 1); //вариационный ряд

int n1 = LookingForUniq(Expon1,Uniq);

FillinFrequency(Expon1, Uniq, y1);
//float x1=0;
//float height =freqfunc(x1,Uniq, y1,n1);
puts("\n");

//for (int i = 0; i<n1+1; i++)
//printf("%f ", Expon1[i]);

puts("\n");

//for(int i=0;i<n1+1;i++)
//printf("%f ", Uniq[i]);
puts("\n");

// [0-Uniq[n1]]/20
float changablepar = 30;
printf("cumulative function or func of raspredelenie\n");
for (int i = 0; i < changablepar; i++)
{

```

```

        printf("%f , %f \n", i*(Uniq[n1]) / changablepar, freqfunc(i*(Uniq[n1]) / changablepar,
Uniq, y1, n1) );//

    }
    //теоритическая функция 1-е^(-ux)

    //гисторграмма

    int part = 10;
    float partx[11]; //полученный от разбиения отрезка на 10
    float party[10];
    for (int i = 0; i < 11; i++)
    {
        partx[i] = 0 + ((Uniq[n1] - 0) * i) / 10.0;
    }

    for (int i = 0; i < 10; i++)
    {
        party[i] = freqfunc(partx[i+1], Uniq, y1, n1) - freqfunc(partx[i], Uniq, y1, n1);
    }
    puts("\n");
    printf("Gistogramma\n");
    for (int i = 0; i < 10; i++)
    printf("[%f ; %f] , %f \n", partx[i], partx[i+1] , party[i]); // гистограмма
    saveinfilegist(partx, party);

    ChosenX = VyborSred(Expon1);
    printf("\n%f is Vyborochnoe Srednie, theoretic %f", ChosenX, 1.0/u);

    ChosenD = VyborDispersia(Expon1, ChosenX);
    printf("\n%f is Vyborochnoe Dispersia, theortic %f ", ChosenD, 1.0 / (u*u) );

    FixedD = FixedDispersia(Expon1, ChosenD);
    printf("\n%f is Fixed Dispersia", FixedD);

    mediana = 0;
    mediana = (Expon1[49] + Expon1[50]) / 2;
    printf("\n%f is mediana, theoretic %f\n", mediana, log(2)/u);

    //для k
    float Expon2[N];

```

```

float k = 3.18;
float y2[N]; //массив с частотами

for (int i = 0; i < N; i++)
{
    Expon2[i] = (-1 / k)*log(x[i]);
    //Expon2[i] = int((Expon2[i] - int(Expon2[i])) * 1000) / 1000.0; // три знака после
запятой

}
//выборка
for (int i = 0; i < N; i++)
    printf("%f ", Expon2[i]);
saveinfile(Expon2);
qsortx(Expon2, 0, N - 1); //вариационный ряд

n1 = LookingForUniq(Expon2, Uniq);

FillinFrequency(Expon2, Uniq, y2);
puts("\n");

//for (int i = 0; i < n1 + 1; i++)
//    printf("%f ", Expon2[i]);

puts("\n");

//for (int i = 0; i < n1 + 1; i++)
//    printf("%f ", Uniq[i]);
puts("\n");

// [0-Uniq[n1]]/100
float changable = 30;
for (int i = 0; i < changable; i++)
{
    printf("%f , %f \n", i*Uniq[n1] / changable, freqfunc(i*Uniq[n1] / changable, Uniq, y2,
n1));//

}
//теоритическая функция 1-e^(-ux)

//гисторграмма

```

```

part = 10;

for (int i = 0; i < 11; i++)
{
    partx[i] = 0 + ((Uniq[n1] - 0)*i) / 10.0;
}

for (int i = 0; i < 10; i++)
{
    party[i] = freqfunc(partx[i + 1], Uniq, y2, n1) - freqfunc(partx[i], Uniq, y2, n1);
}
puts("\n");
for (int i = 0; i < 10; i++)
    printf("[%f ; %f] , %f \n", partx[i], partx[i + 1], party[i]); // гистограмма
saveinfilegist(partx, party);

//-----

ChosenX = VyborSred(Expon2);
printf("\n%f is Vyborochnoe Srednie, theoretic %f", ChosenX, 1.0 / k);

ChosenD = VyborDispersia(Expon2, ChosenX);
printf("\n%f is Vyborochnoe Dispersia, theortic %f ", ChosenD, 1.0 / (k*k));

FixedD = FixedDispersia(Expon2, ChosenD);
printf("\n%f is Fixed Dispersia", FixedD);

mediana = 0;
mediana = (Expon2[49] + Expon2[50]) / 2;
printf("\n%f is mediana, theoretic %f\n", mediana, log(2) / k);
//-----4
printf("\n4 task\n");
const float E = 5.0;
const float d = 4.18;

float x4_1[N]; //кси
float x4_2[N];
Input(x4_1);
Input2(x4_2);
//эта
float Eta1[N];
float Eta2[N];
for (int i=0;i<N;i++)
{

```

```

    Eta1[i] = E+cos(2 * M_PI*x4_1[i])*sqrt(-2 * log(x4_2[i]))*sqrt(d);
    Eta2[i]= E+sin(2 * M_PI*x4_1[i])*sqrt(-2 * log(x4_2[i]))*sqrt(d);
    //Eta1[i]=int((Eta1[i] - int(Eta1[i])) * 1000) / 1000.0;
    //Eta2[i] = int((Eta2[i] - int(Eta2[i])) * 1000) / 1000.0;
}
printf("Viborki");
for (int i = 0; i < N; i++)
{
    printf("%f ", Eta1[i]);
}
saveinfile(Eta1);
float y4[N];
qsortx(Eta1, 0, N - 1); //вариационный ряд

n1 = LookingForUniq(Eta1, Uniq);

FillinFrequency(Eta1, Uniq, y4);
//float x1=0;
//float height =freqfunc(x1,Uniq, y1,n1);
puts("\n");

//for (int i = 0; i<n1 + 1; i++)
//    printf("%f ", Eta1[i]);

puts("\n");

//for (int i = 0; i<n1 + 1; i++)
//    printf("%f ", Uniq[i]);
puts("\n");

// [0-Uniq[n1]]/20
printf("cumulative function or func of raspredelenie\n");
changable = 20;
for (int i = 0; i < changable; i++)
{

    printf("%f , %f \n", Uniq[0]+i*(Uniq[n1]) / changable, freqfunc(Uniq[0] + i*(Uniq[n1]) /
changable, Uniq, y4, n1));

}

//теоритическая функция 1-e^(-ux)

```



```
//гистограмма
```

```
float partx4[partition+1]; //полученный от разбиения отрезка на 10
float party4[partition];
for (int i = 0; i < partition+1; i++)
{
    partx4[i] = Uniq[0] + ((Uniq[n1] - Uniq[0])*i) / partition;
}

for (int i = 0; i < partition; i++)
{
    party4[i] = freqfunc(partx4[i + 1], Uniq, y4, n1) - freqfunc(partx4[i], Uniq, y4, n1);
}
puts("\n");
printf("Gistogramma\n");
for (int i = 0; i < partition; i++)
    printf("[%f ; %f] , %f \n", partx4[i], partx4[i + 1], party4[i]); // гистограмма
saveinfilegist(partx4, party4);

ChosenX = VyorSred(Eta1);
printf("\n%f is Vyborochnoe Srednie, theoretic %f", ChosenX, E);

ChosenD = VyorDispersia(Eta1, ChosenX);
printf("\n%f is Vyborochnoe Dispersia, theortic %f ", ChosenD, d);

FixedD = FixedDispersia(Eta1, ChosenD);
printf("\n%f is Fixed Dispersia", FixedD);

mediana = 0;
mediana = (Eta1[49] + Eta1[50]) / 2;
printf("\n%f is mediana, theoretic %f\n", mediana, E);

//-----для эта 2
float y4_2[N];
for (int i = 0; i < n1 + 1; i++)
    printf("%f ", Eta2[i]);
saveinfile(Eta2);
qsortx(Eta2, 0, N - 1); //вариационный ряд

n1 = LookingForUniq(Eta2, Uniq);

FillinFrequency(Eta2, Uniq, y4_2);
//float x1=0;
//float height =freqfunc(x1,Uniq, y1,n1);
puts("\n");
```

```

//for (int i = 0; i<n1 + 1; i++)
//    printf("%f ", Eta2[i]);

puts("\n");

//for (int i = 0; i<n1 + 1; i++)
//    printf("%f ", Uniq[i]);
puts("\n");

// [0-Uniq[n1]]/20
printf("cumulative function or func of raspredelenie\n");
changable = 20;
for (int i = 0; i < changable; i++)
{
    printf("%f , %f \n", Uniq[0] + i * (Uniq[n1]) / changable, freqfunc(Uniq[0] + i *
(Uniq[n1]) / changable, Uniq, y4_2, n1));//

}
//теоритическая функция 1-e^(-ux)
//гисторграмма
float partx4_2[partition + 1];//полученный от разбиения отрезка на 10
float party4_2[partition];
for (int i = 0; i < partition + 1; i++)
{
    partx4_2[i] = Uniq[0] + ((Uniq[n1] - Uniq[0])*i) / partition;
}

for (int i = 0; i <partition; i++)
{
    party4_2[i] = freqfunc(partx4_2[i + 1], Uniq, y4_2, n1) - freqfunc(partx4_2[i], Uniq,
y4_2, n1);
}
puts("\n");
printf("Gistogramma\n");
for (int i = 0; i < partition; i++)
    printf("[%f ; %f] , %f \n", partx4_2[i], partx4_2[i + 1], party4_2[i]);// гистограмма
saveinfilegist(partx4_2, party4_2);

ChosenX = VyorSred(Eta2);
printf("\n%f is Vyborchnoe Srednie, theoretic %f", ChosenX, E);

```

```

ChosenD = VbyorDispersia(Eta2, ChosenX);
printf("\n%f is Vbyorochnoe Dispersia, theortic %f ", ChosenD, d);

FixedD = FixedDispersia(Eta2, ChosenD);
printf("\n%f is Fixed Dispersia", FixedD);

mediana = 0;
mediana = (Eta2[49] + Eta2[50]) / 2;
printf("\n%f is mediana, theoretic %f\n", mediana, E);

return 0;
}

```