Долгосрочное домашнее задание

Свиридов Иван СКБ 182

14 декабря 2020 г.

1. Домашнее задание. Вероятностные распределения

1.1. Равномерное дискретное распределение

1.1.1. Основные понятия

Дискретное равномерное распределение: P(x)=N^{-1}, x $\in$ [1,..,N],(P(x)=P(ξ= x_i) =p_i
Найдем функцию распределения A для нашего дискретного равномерного распределения по определению:
Исправлено:

F_ξ(x) $= P(\xi \le x) \Rightarrow$ Применим для нашего распределения $\Rightarrow$

$$F_\xi(x) = \begin{cases} 0, & \text{при } x < 1 \\ \sum_{i:x_i \le x} \le P(x_i) & 1 \text{ при } x \le N \\ 1, & \text{при } x > N \end{cases} \Rightarrow \begin{cases} 0, & \text{при } x < 1 \\ \frac{|x|}{N} & 1 \text{ при } x \le N \\ 1, & \text{при } x > N \end{cases}$$

-т.к F_ξ(x)=0, при x<1

$$\lim_{x\to-\infty} F_\xi(x) = 0$$

-т.к F_ξ(x)=1, при x>N

$$\lim_{x\to\infty} F_\xi(x) = 1$$

-функция не является непрерывной, так как наблюдается разрыв типа "скачок"

Def: Математическое ожидание(среднее значение, первый момент) случайной величины ξ с дискретным распределением называется число M_ξ= $\sum$ x_i p_i=$\sum\limits_i$ x_i P(ξ=x_i), если данный ряд сходится абсолютно, то есть, если $\sum$ x_i $\cdot$ p_i<∞, иначе не существует.
Для нашего равномерного дискретного распределения: M_ξ=$\sum\limits_{i=1}^{N}$ x_i $\cdot$ p_i=$\sum\limits_{i=1}^{N}$ x_i $\cdot$ $\frac{1}{N}$=$\frac{1}{N}$ $\sum\limits_{i=1}^{N}$ x_i=$\frac{1}{N}$ $\cdot$ $\frac{1}{2}$ $\cdot$N $\cdot$(N+1)=$\frac{N+1}{2}$

Def: Дисперсия случайной дискретной величины ξ - число: D_ξ=M(ξ-M_ξ)^2= $\sum\limits_{i=1}^{N}$ p_i(x_i-M_ξ)^2, по смыслу - среднее значение квадрата отклонение ξ от своего среднего.
Для нашего равномерного дискретного распределения: D_ξ=$\sum\limits_{i=1}^{N}$ p_i(x_i-M_ξ)^2=M_ξ^2-(M_ξ)^2=(в силу линейности M_ξ)=$\sum\limits_{i=1}^{N}$ x_i^2 $\cdot$ p_i -$\frac{(N+1)^2}{4}$=
=$\frac{1}{6}$ $\cdot$ (N+1)(2N+1)-$\frac{(N+1^2)}{4}$=(N+1)$\cdot$($\frac{1}{6}$ $\cdot$ 2N+$\frac{1}{6}$-$\frac{N}{4}$-$\frac{1}{4}$)=(N+1)$\cdot$($\frac{4N-3N+2-3}{12}$)=(N+1)($\frac{N-1}{12}$)=$\frac{1}{12}$ $\cdot$ (N^2-1)=$\frac{N^2-1}{12}$

Def: Мода распределения случайной дискретной величини ξ - величина, которая равна значению (случайной величины) с наибольшей вероятностью.
Исправлено:

$P_\xi(x) = const \; \forall \; x \in [1, ..., N] \Rightarrow$ мода - любое значение из $[1, ..., N]$

Def: Медиана распределения дискретной случайной величины ξ - величина, равная значению (случайной величины), для которого вероятность меньших значений равна вероятности больших значений.

$$M_e = \begin{cases} \nexists, & \forall i \in N => \sum\limits_{s=1}^{i-1} P_s \neq \sum\limits_{s=1+i}^{N} P_s \\ x_i, & \exists i \in N => \sum\limits_{s=1}^{i-1} P_s = \sum\limits_{s=1+i}^{N} P_s \end{cases}$$

Для нашего дискретного равномерного распределения:
если N - нечетное, то $M_e = \frac{N+1}{2}$
если N - четное, то $M_e \; \forall \; [\frac{N}{2}, \frac{N}{2}+1]$
Из этого следует, что $M_e = \frac{N+1}{2}$

Исправлено:

Def: Производящая функция дискретной неотрицательной случайной величины - величина, равная $M_\xi(\mathrm{t}) = M[e^{\xi t}]$. Для нашего равномерного дискретного распределения:
$M_\xi(\mathrm{t}) = \sum\limits_{i=1} e^{tx_i} \cdot p_i = \sum\limits_{i=1} e^{tx_i} \cdot \frac{1}{N} = \frac{1}{N} \cdot \sum\limits_{i=1} e^{tx_i} = \frac{e^t(e^{Nt}-1)}{N(e^t-1)}$

Def: Характеристическая функция дискретной случайной величины ξ - величина: $\phi_\xi(\mathrm{t}) = \sum\limits_{k} e^{itx_k} \cdot p_k$
Для нашего равномерного дискретного распределения распределения:
$\phi_\xi(\mathrm{t}) = \sum\limits_{k=1} e^{itx_k} \cdot p_k = \sum\limits_{k=1} e^{itx_k} \cdot \frac{1}{N} = \frac{1}{N} \cdot \sum\limits_{k=1} e^{itx_k} = \frac{e^{it}(e^{iNt}-1)}{N(e^{it}-1)}$

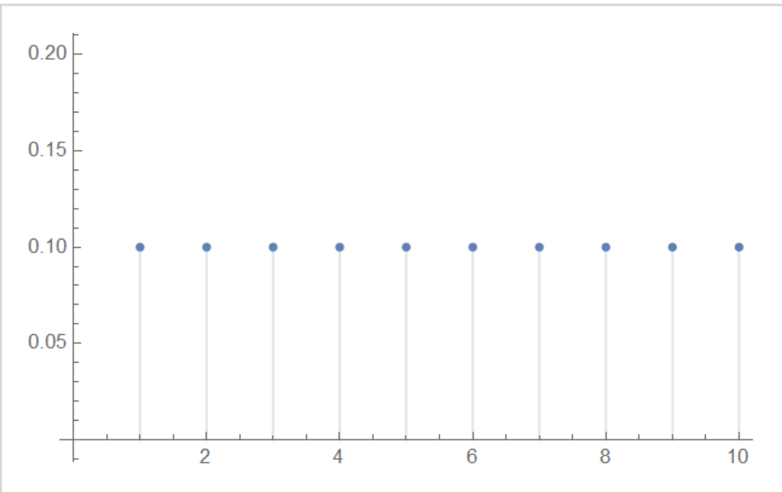

Рис. 1: Гистограмма распределения

Рис. 2: При n=10

```
Manipulate[DiscretePlot[1/N, {x, 1, N}, PlotRange → Automatic], {N, 1, 100, 1}]
|варьировать |график последовательности |ч··· |отображаем··· |автоматический |численное прибли
```

Рис. 3: Код Wolfram Mathematica 12.1

1.1.2. Типичная и нетипичная интерпретация

Типичная интерпритация

Рассмотрим пример дискретного равномерного распределения:
Когда мы бросаем игральные кости, на кубе выпадает случайное ребро со своим значением. Это пример дискретного равномерного распределения, естественно, если кость "идеальна как и условия бросания. Таким образом мы расссматриваем случайную величину ξ, с законом распределения $P_\xi=\frac{1}{N}$, x $\in$ [1,..,N], где N=6 (у куба 6 сторон). Нас может интересовать такой вопрос, а сколько очков выпадает в среднем на игральной кости? Сформулировать этот вопрос на языке теории вероятности, мы придем к задаче о поиске математического ожидания величины ξ. Как нам известно, математическое ожидание(или первый момент) ищется по формуле $M_\xi=\sum\limits_i p_i\ x_i$, для нашего случая:

$M_\xi=\sum\limits_{i=1}^{6} \frac{1}{6}\ x_i=\frac{1}{6} \sum\limits_{i=1}^{6}\ x_i=\frac{1}{6}\ (1+2+3+4+5+6)=\frac{24}{6}=3{,}5$-количество очковбвыпадающее в среднем на игральной кости.

Исправлено:

Интерпритация

Представим нетипичную интерпритацию, когда мы играем с младшим братом в игру вытяни карточку с действием из шляпы. Всего существует N карточек внутри шляпы. Мы написали на каждой карточке какие-то действия. И брат вытягивает каждый раз карточку с действием. Вероятность что достанет нужное(удобное для него) действие составляет $\frac{1}{N}$. Так как он достает карточку в слепую, и не знает что на ней написано, то исход не зависит от человека. Из этого примера следует, что дискретным равномерным распределением можно описать игру с выбором карточки с действием из шляпы.

1.1.3. Моделирование

Реализуем выборочное значение α стандартного случайного числа и полагаем m=[α N]+1 = [α N +1]. Рассмотрим теперь подход, который состоит в приведении вероятностей к общему знаменателю.
Пусть ξ — дискретное распределение с конечным, относительно небольшим количеством значений N. Пусть также все вероятности pi — это обыкновенные дроби, которые можно привести к общему знаменателю M, то есть $p_i=\frac{m_i}{M}$, причем $m_1+...+m_n$=M и M $\leq M_0$, где M_0 - размер максимального массива для данного компьютера и языка программирования.
Рассмотрим случайную величину ξ', такую, что $\xi'= x_j'= x_i$ при j=$\sum\limits_{s=1}^{i-1} m_s+1,...,\sum\limits_{s=1}^{i} m_s$, причем P($\xi' = x_j'$)=$\frac{1}{M}$. Это значит, что первые m_1 значений x_j' равна x_1, следующие m_2 значит x_j' равны x_2 и так далее, причем все значения x_j' равновероятны.
Выборочные значения случайной величины ξ' реализуются согласно правилу: $\xi' = x_m'$, при m'=Mα+1.

```python
import numpy
import random
from matplotlib import pyplot, axes
from statsmodels.distributions.empirical_distribution import ECDF

def generate(N=10):
    lx = []
    dic = {}
    for i in range(N):
        dic[i+1]=(i/N,(i+1)/N)
    for i in range(600):
        rand_number = random.random()
        #we need to create helpfull dictionary:
        for i in dic:
            if dic[i][0] <= rand_number < dic[i][1]:
                x = i
        lx.append(x)
    lx.sort()
    ly = []
    #counting probabilities of samples and appending to ly
    for i in range(N):
         ly.append(lx.count(i+1)/600)
           #dont need samples lx more
    lx = [i+1 for i in range(N)]

    print(lx, ly)
    pyplot.plot([i+1 for i in range(N)], [1/N for _ in range(N)], 'ro')
    axes= pyplot.axes()
    axes.set_ylim([0,1])
    pyplot.plot(lx, ly, 'go')
    pyplot.show()

generate()
```

Рис. 4: Код

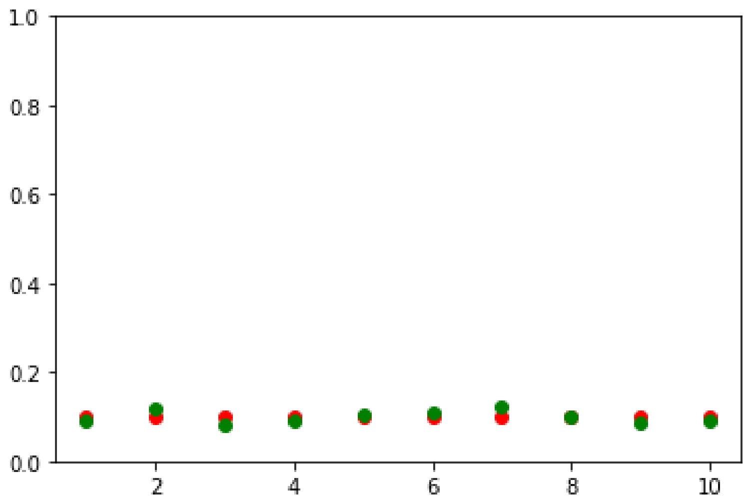

Рис. 5: При n=10

1.2. Экспоненциальное распределение

Непрерывное экспоненциальное (показательное) распределение $f(x) = \lambda \cdot e^{-\lambda \cdot x}$
Мода экспоненциального распределения равна нулю, так как f(0)=λ>f(x), ∀ x>0
Найдем функция распределения для показательного распределения:

$F(x) = \int_{-\infty}^{x} f(x)dx = \int_{-\infty}^{x} \lambda e^{-\lambda x}dx = 1 - e^{-\lambda x}, (x > 0)$

Воспользуемся уравнением F(Me)=$\frac{1}{2}$ для нахождения медианы распределения:
$F(Me) = 1 - e^{-\lambda Me} = \frac{1}{2} -> Me = \frac{\ln x}{\lambda}$

Вычислим матетматчиеское ожидание:

$$M_\xi = \int_{-\infty}^{\infty} xf(x)dx = \lambda \int_{0}^{\infty} xe^{-\lambda x}dx = \lim_{b\to\infty}((x + \tfrac{1}{\lambda})e^{-\lambda x}) =$$

$$= -\lim_{b\to\infty}((b + \tfrac{1}{\lambda})e^{-\lambda b}) -> 0 - (0 + \tfrac{1}{\lambda})e^{0}) = -(0 - \tfrac{1}{\lambda}) = \tfrac{1}{\lambda}$$

Def: Вычислим дисперсию:

$$D_\xi = \int_{-\infty}^{+\infty} x^2 f(x)dx - (M_\xi)^2$$

$$(*) \int_{-\infty}^{+\infty} x^2 f(x)dx = \lambda \int_{0}^{+\infty} x^2 e^{-\lambda x}dx = (*simplify*) = -(x^2 + \tfrac{2x}{\lambda} + \tfrac{2}{\lambda^2})e^{-\lambda x} = -\lim_{b\to+\infty}(x^2 + \tfrac{2x}{\lambda} + \tfrac{2}{\lambda^2}$$

$$D_\xi = \tfrac{2}{\lambda^2} - (\tfrac{1}{\lambda}) = \tfrac{2}{\lambda^2} - \tfrac{1}{\lambda^2} = \tfrac{1}{\lambda^2}$$

Подправлено:

Def: Производящая функция $M_\xi(t)$:

По определению: $M_\xi(t) = M[e^{t\xi}]$. Для нахождения, воспользуемся свойством:
Где ξ - случайная величина, $M[g(\xi)] = \int_{\mathbb{R}} g(x_i) \cdot f_\xi(x)dx$

Найдем $M_\xi(t)$ для нашего распределения:

$$M_\xi(t) = M[e^{t\xi}] = \int_{0}^{\infty} e^{tx} \cdot f_\xi(x)dx = \int_{0}^{\infty} e^{tx} \cdot \lambda e^{-\lambda x}dx = \lambda \int_{0}^{\infty} e^{x(t-\lambda)}dx =$$

$$= \frac{\lambda}{\lambda - t} = (1 - \frac{t}{\lambda})^{-1}$$

Def: Характеристическая функция $\varphi_\xi(t)$:

По определению: $\varphi_\xi(t) = M[e^{it\xi}]$. Для нахождения, воспользуемся свойством: $\int_R e^{itx}P(x)dx$
Найдем $\varphi_\xi(t)$ для нашего распределения:

$$\varphi_\xi(t) = M[e^{it\xi}] = \int_{0}^{\infty} e^{itx} \cdot f_\xi(x)dx = \int_{0}^{\infty} e^{itx} \cdot \lambda e^{-\lambda x}dx = \lambda \int_{0}^{\infty} e^{x(it-\lambda)}dx =$$

$$= \frac{\lambda}{\lambda - it} = (1 - \frac{it}{\lambda})^{-1}$$

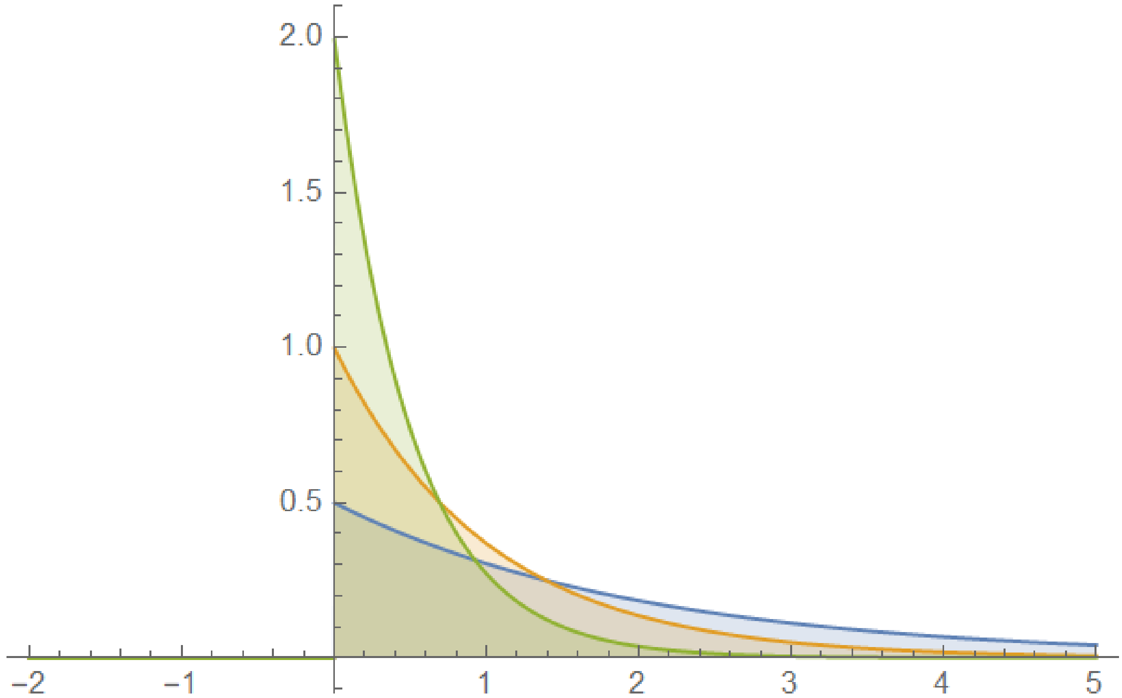

Рис. 6: Плотность при λ=$\frac{1}{2}$,1,2

```
Plot[Table[PDF[ExponentialDistribution[λ], x],
    {λ, {1/2, 1, 2}}] // Evaluate, {x, -2, 5}, Filling → Axis,
 PlotRange → All]
```

Рис. 7: Код Wolfram Mathematica 12.1

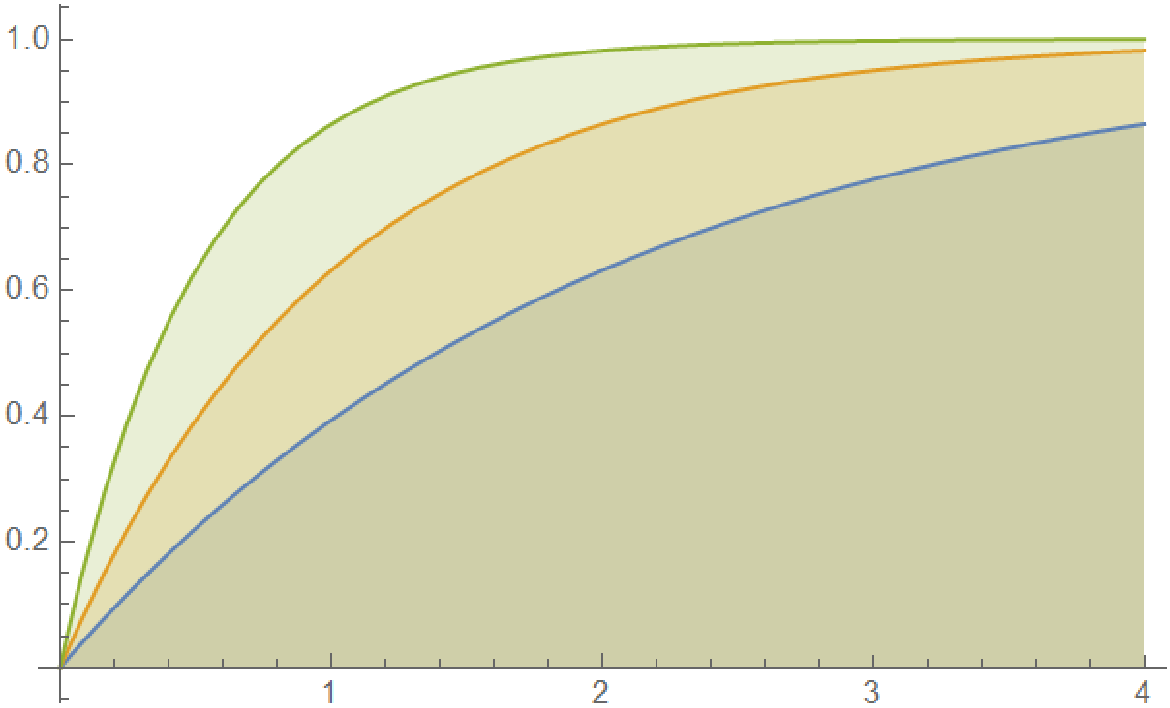

Рис. 8: Функция распределения при $\lambda=\frac{1}{2}$,1,2

```
Plot[Table[CDF[ExponentialDistribution[λ], x],
    {λ, {1/2, 1, 2}}] // Evaluate, {x, 0, 4}, Filling → Axis]
```

Рис. 9: Код Wolfram Mathematica 12.1

1.3 Типичная и нетипичная интерпретация

Типичная интерпретация. Рассмотрим случай, когда мы пришли в ремонтную мастерскую и нам необходимо починить персональный компьютер в течении 5 дней, а мастер говорит, что успеют починить ,но возможно потребуется больше 5 дней. По заверению мастера, в среднем один компьютер чинится 3 дня. Значит наш параметр показательного распределения λ=$\frac{1}{3}$. Как бы нам не хотелось, чтобы чинили больше 5 дней, нам необходимо узнать вероятность, что на ремонт ноутбука потребуется не меньше 5 дней. Так как X - случайная величина, распределенная по показатальному закону, ее функция распределения равна 1-$e^{-\lambda x}$(x>0), а вероятность попадания в интервал вычисляется по формуле показательного распределения:

$P(a < x < b) = e^{-\lambda a} - e^{-\lambda b} = e^{-\frac{a}{3}} - e^{-\frac{b}{3}}$

Тогда вероятность того, что на ремонт ноутбука потребуется не менее 5 дней. То есть X$\geq$5, равна:

$P(X \geq 5) = 1 - P(0 < x < 5) = 1 - (e^{-\frac{0}{3}} - e^{-\frac{5}{3}}) = 1 - 1 + e^{-\frac{5}{3}} \approx 0,189$

Таким образом, с помощью показательного распределения мы можем определить вероятность, придется ли нам больше ждать, если нам срочно нужен ноутбук.

Подправлено:

Нетипичная интерпретация.Рассмотрим нетипичную интерпритацию для показательного распределения. Возьмем некоторое видео на платформе "Youtube"в интернете. И рассмотрим, как появляются комметарии под этим видео. Существует правило, что чем дольше видео находится на площадке интернет-роликов, тем реже появляются новые комментарии под этим видео. Пускай каждый новый пользователь оставляет новый комментарий независимо от других пользователей.Также оставляется не более одного комментария за малый промежуток времени.

Обозначим через X_s - число комментариев написанных под видео за определенное время S. Из выше описанных условий, будет следовать, что параметр λ - интенсивность появления комментариев. В данной модели ставится вопрос, а как распределено время между появлениям соседних комменатриев. Из всего этого следует, что время между появлениями комментариев распределено экспоненциально. Таким образом в модели, касающейся казалось бы дискретных объектов, проявит себя экспоненциальное распределение. Анализ проводился видео от 04.11.2020 (`https://www.youtube.com/watch?v=-oCqhfiGbfU&ab_channel=Wylsacom`)на канале Wylsacom(`https://www.youtube.com/user/Wylsacom`). С помощью сервиса `https://popsters.ru/app/dashboard`

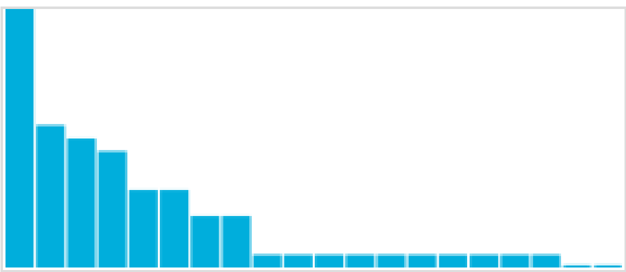

Рис. 10: Комметарии

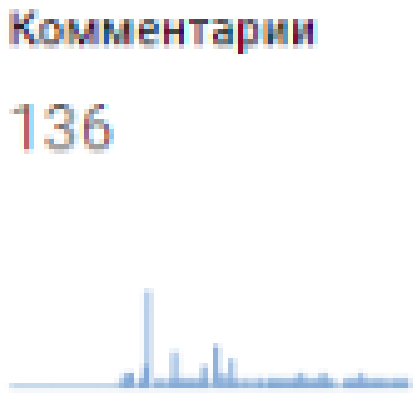

Рис. 11: Комметарии

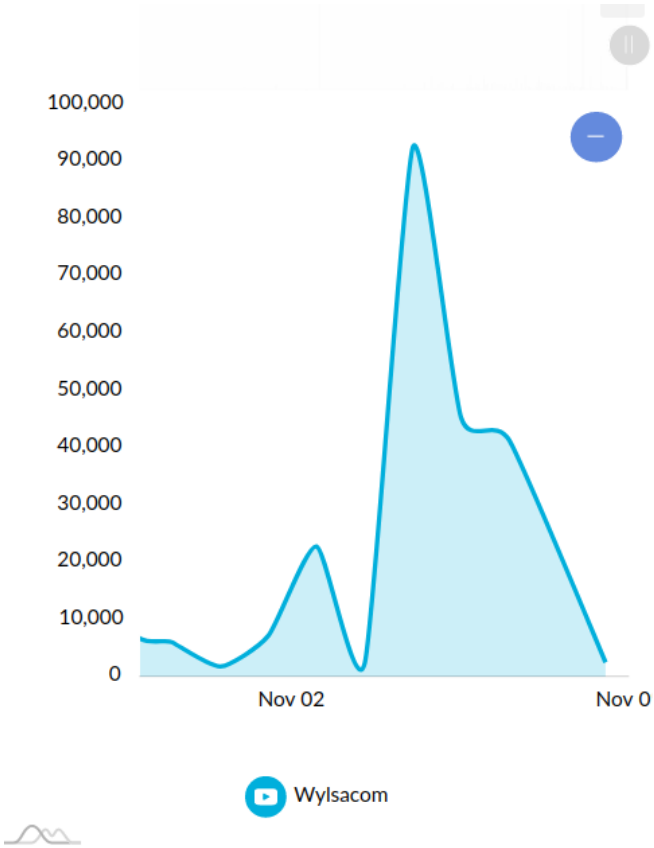

Рис. 12: Комметарии

1.4 Моделирование показательного распределения:

Пусть дана функция распределения случайной величины ξ-F_ξ(x), причем F_ξ(x) - непрерывна и монотонно возрастает. Тогда F_ξ^{-1}(y) существует и является обратной функцией, определенной на отрезке [0,1]. Пусть η будет являться случайной величиной с равномерным распределением, где f_η(x)$\in$ [0,1]. Тогда:
$P(\eta_i \leq F_\xi(x)) = P(F_\xi^{-1}(\eta_i \leq x) = P(x_i \leq x) = F_\xi(x)$
Из этого следует, что при помощи случайной величины η генерируется случайная величина, которая имеет распределение F_ξ(x). Тогда, при условии, что мы знаем янвный вид F_ξ^{-1}(y), то получается определенный алгоритм моделирования величины ξ.
Первый шаг: генерируется следующее значение η_i
Второй шаг: подставляем полученное значение в обратную функцию F_ξ^{-1}(y)
Третий шаг: получаем очереднок значение величины X

Найдем обратную функцию к F_ξ(x)=1-$e^-\lambda^x$ в области x $\geq$ 0. И реализуем алгоритм для показательного распределения:

$F_\xi(x) = 1 - e^(-\lambda)^x$
$y = 1 - e^{\lambda^x}$
$X = \frac{\ln(1-y)}{-\lambda}$

```
import random
import matplotlib.pyplot as plt
import math
import numpy as np
from statsmodels.distributions.empirical_distribution import ECDF
from scipy.stats import expon

def Pocaz():
    lx=[]
    ly=[]

    yi = lambda x: 1-math.e**(-x)
    xi = np.linspace(0, 9,100)
    def modal(x):
        return(-math.log(1-x)/1)
    for i in range(500):
        X=random.random()#сохранение значений
        Y=modal(X)
        lx.append(X)
        ly.append(Y)
    ecdfy=ECDF(ly)
    #print(ecdfy.x,ecdfy.y,sep='\n')
    plt.plot(ecdfy.x,ecdfy.y,'ro')
    plt.plot(xi,yi(xi))
    plt.show()

Pocaz()
```

Рис. 13: Код

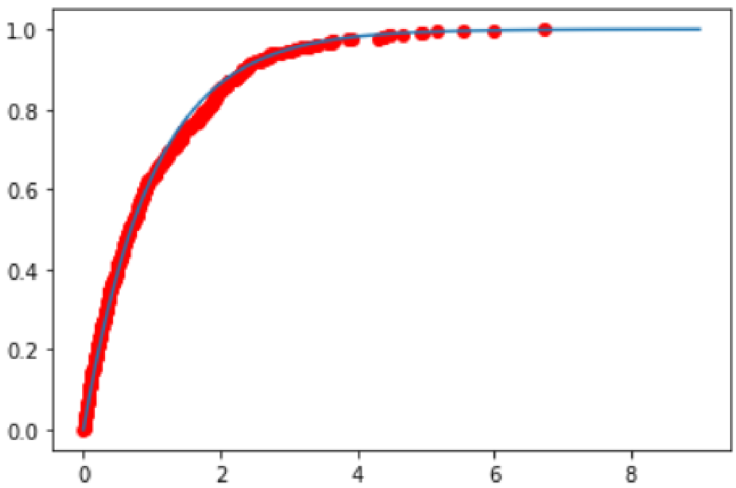

Рис. 14: График

2. Домашнее задание. Основные понятия математической статистики

2.1. Равномерное дискретное распределение

2.1.1. Моделирование выбранной случайной величины

На основе моделирования дискретного равномерного распределения, описанного в домашнем задании 1, создаем функцию, с помощью которой генерируем по 5 выборок каждого объема. Для примера напечатаны сгенерированные выборки объема n=5 и n=10.
Output:

-n=5
[5, 3, 3, 3, 4]
[4, 7, 7, 1, 2]
[6, 5, 6, 1, 5]
[3, 5, 5, 1, 5]
[2, 7, 7, 7, 6]

-n=10
[4, 7, 5, 1, 4, 4, 7, 1, 4, 3]
[4, 6, 4, 4, 7, 6, 7, 2, 5, 2]
[7, 6, 7, 6, 4, 6, 5, 4, 2, 4]
[2, 4, 2, 1, 2, 2, 6, 2, 1, 2]
[1, 5, 3, 1, 7, 1, 4, 4, 2, 3]

Код, с помощью которого генерируетсяпо 5 выборок каждого объема n=5, 10, 100, 1000, 10000. Итого 25 различных выборок для выбранного дискретного распределения.
Code:

```
import numpy
import random
from matplotlib import pyplot, axes

def generate(N=8,n=5):
    lx=[]
    dic = {}
    for i in range(N):
        dic[i+1]=(i/N,(i+1)/N)
    for i in range(n):
        rand_number = random.random()
        for i in dic:
            if dic[i][0] <= rand_number < dic[i][1]:
                x = i
        lx.append(x)
    return(lx)
generate()
```

Рис. 15: Код выборок n=5

```python
import numpy
import random
from matplotlib import pyplot, axes

def generate(N=8,n=10):
    lx=[]
    dic = {}
    for i in range(N):
        dic[i+1]=(i/N,(i+1)/N)
    for i in range(n):
        rand_number = random.random()
        for i in dic:
            if dic[i][0] <= rand_number < dic[i][1]:
                x = i
        lx.append(x)
    return(lx)
generate()
```

Рис. 16: Код выборок n=10

2.1.2. Построение эмпирической функции распределения

Для каждой выборки строим эмпирическую функцию распределения. В отчете продемонстрировано построение эмпирической функции распределения для выборок размера 5. На одном графике отображены эмпирические функции распределения для 5 выборок данного объема и график функции распределения.

Code:

```python
def ecdf(sample):
    ly=[]
    for i in range(7):
        ly.append(sample.count(i+1)/len(sample))
        ly = list(filter(lambda a: a != 0, ly))
        ly.insert(0,0)
    for i in range(1,len(ly)):
        ly[i]= ly[i]+ly[i-1]
    return(ly)

print(ecdf(samples_list[0]))
def gen_ecdf(samples_list):
    ecdf_list = []
    for i in range(25):
        ecdfx = list(set(samples_list[i]))
        ecdfx.insert(0,numpy.NINF)
        ecdfy = ecdf(samples_list[i])
        ecdf_list.append((ecdfx, ecdfy))
    return(ecdf_list)

ecdf_list = gen_ecdf(samples_list)
print(ecdf_list[0])
for i in range(5):
    sample_ecdf = ecdf_list[i]
    print('ECDF F(x) for sample', i+1, ':')
    for k in range(len(sample_ecdf[0])-1):
        print(sample_ecdf[1][k], ', ', sample_ecdf[0][k], '< x <=', sample_ecdf[0][k+1])
    print(sample_ecdf[1][-1], ', x >', sample_ecdf[0][-1], '\n')
```

Рис. 17: Код

Output:

ECDF F(x) for sample 1 :
0 , -inf < x <= 3
0.6 , 3 < x <= 4
0.8 , 4 < x <= 5
1.0 , x > 5

ECDF F(x) for sample 1 :
0 , -inf < x <= 3
0.6 , 3 < x <= 4
0.8 , 4 < x <= 5
1.0 , x > 5

ECDF F(x) for sample 2 :
0 , -inf < x <= 1
0.2 , 1 < x <= 2
0.4 , 2 < x <= 4
0.6000000000000001 , 4 < x <= 7
1.0 , x > 7

ECDF F(x) for sample 3 :
0 , -inf < x <= 1
0.2 , 1 < x <= 5
0.6000000000000001 , 5 < x <= 6
1.0 , x > 6

ECDF F(x) for sample 4 :
0 , -inf < x <= 1
0.2 , 1 < x <= 3
0.4 , 3 < x <= 5
1.0 , x > 5

ECDF F(x) for sample 5 :
0 , -inf < x <= 2
0.2 , 2 < x <= 6
0.4 , 6 < x <= 7
1.0 , x > 7

Code:

```
for k in range(0,5):
    sample_ecdf = ecdf_list[k]
    print(sample_ecdf)
    color = colors[k%5]
    pyplot.plot([0,sample_ecdf[0][1]],[0,0], c=color, alpha=0.7, linewidth=4)
    for i in range(1,len(sample_ecdf[0])-1):
        pyplot.plot([sample_ecdf[0][i],sample_ecdf[0][i+1]],[sample_ecdf[1][i],sample_ecdf[1][i]],
            c=color, alpha=0.7,linewidth=4)
        pyplot.plot([sample_ecdf[0][-1], sample_ecdf[0][-1]+1],[1,1],c=color, alpha=0.7, linewidth=4)
pyplot.xlabel('numbers')
pyplot.ylabel('probability')
pyplot.title('ECDF')
x = numpy.linspace(1,7,100)
pyplot.plot(x, y:=numpy.floor(x)/7,'b')
pyplot.plot(numpy.linspace(0,1,100), y:=x*0, 'b')
pyplot.plot(numpy.linspace(7,8,100), y:=(0*x)+1, 'b')
pyplot.show()
```

Рис. 18: Код

Output:

-n=5

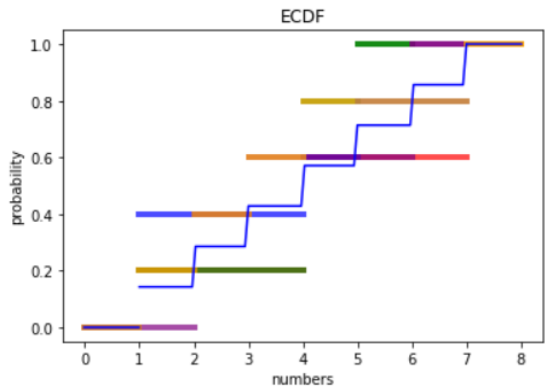

Рис. 19: n=5

-n=10

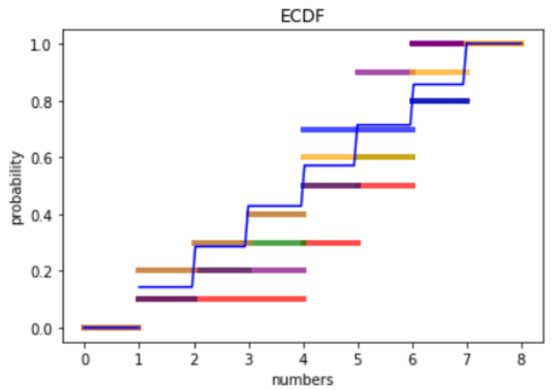

Рис. 20: n=10

-n=100

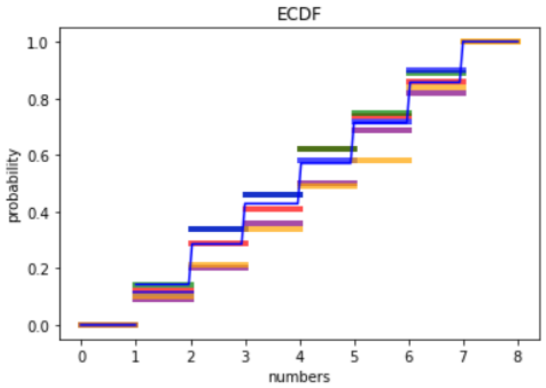

Рис. 21: n=100

-n=1000

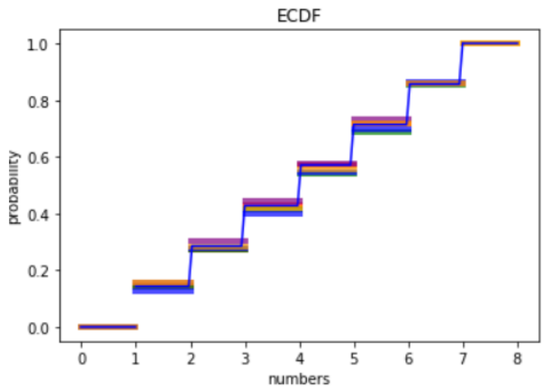

Рис. 22: n=1000

-n=10000

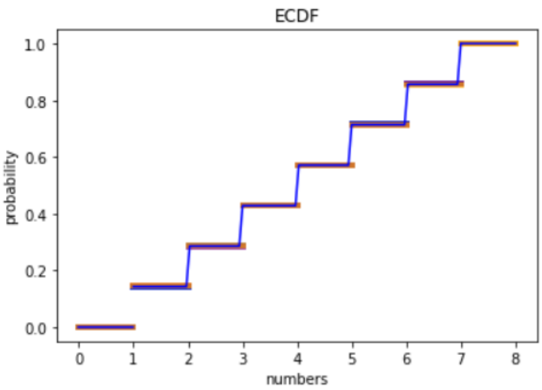

Рис. 23: n=10000

Существует училенный закон больших чисел, в котором эмпирическая функция почти наверное к теоретической функции: $\hat{F}x) \to F(x)$ почти наверное при $n \to \infty$. По данным, которые представлены на графике видно, что совпадение графика эмпирической функции с графиком теоретической функции распределения зависит от объема выборки n.Чем оно больше, тем больше совпадают графики.
Для каждого n найдем максимальную точную верхнюю границу разности каждой пары эмпирических функций распределения среди всех возможных пар. Для Каждой пары рассмотрим определенное количество вариантов, равное C_5^2
Точная верхняя граница разности - это наибольшее число разности значений $ECDF$ функций при одинаковых аргументах. Данное число можно вычислить при помощи кода представленного ниже.

Code:

```
diff_list=[]
for i in range(4):
    for k in range(i+1,10):
        sample1=ecdf_list[i+5*n]
        sample2=ecdf_list[k+5*n]
        for j in range(1,len(sample1[0])-1):
            pair1 = [sample1[0][j],sample1[0][j+1]]
            for p in range(1,len(sample2[0])-1):
                pair2=[sample2[0][p],sample2[0][p+1]]
                if ((pair2[0]<=pair1[0] and pair2[1] >= pair1[0]) or (pair2[0]<=pair1[1] and pair2[1]>=pair1[1])):
                    diff_list.append(abs(sample1[1][j]-sample2[1][p]))
print(max(diff_list))
```

Рис. 24: Код

Output:
n=5–0.6000000000000001
n=10–0.6000000000000001
n=100–0.33
n=1000–0.16599999999999998
n=100000–0.05281999999999994
Точные верхнии границы разности будут уменьшаться при приближении эмпирической функции к теоретической с ростом n.

2.1.3. Построение вариационного ряда выборки

Вариационный ряд - набор элементов X_1,...,X_n, упорядоченный по возрастанию. Ниже представлен код, с помощью которого строятся вариационные ряды для каждой сгенерированной выборки. В отчете представлены вариационные ряды для выборок объема $n = 5$, $n = 10$.

Code:

```
var_list = []

for sample in samples_list:
    var_sample = sorted(sample)
    var_list.append(var_sample)

for i in range(10):
    print(var_list[i])
```

Рис. 25: Код

Output:

-$n = 5$:
[3, 3, 3, 4, 5]
[1, 2, 4, 7, 7]
[1, 5, 5, 6, 6]
[1, 3, 5, 5, 5]
[2, 6, 7, 7, 7]

-$n = 10$:
[1, 1, 3, 4, 4, 4, 4, 5, 7, 7]
[2, 2, 4, 4, 4, 5, 6, 6, 7, 7]
[2, 4, 4, 4, 5, 6, 6, 6, 7, 7]
[1, 1, 2, 2, 2, 2, 2, 2, 4, 6]
[1, 1, 1, 2, 3, 3, 4, 4, 5, 7]

α-квантиль случайной величины ξ с функцией распределения $F_\xi(x)$=$P(\xi < x)$-любое число x_α, которое удовлетворяет двум условиям:
-$F_\xi(x_\alpha) \leq \alpha$
-$F_\xi(x_\alpha + 0) \geq \alpha$

Code:

```
quantile_list=[0.1,0.5,0.7]

for element in ecdf_list:
    for quantile in quantile_list:
        if quantile in element[1]:
            q = element[1].index(quantile)
            print('q=', quantile, 'in', len(element[1])-1, 'x=', element[0][q+1])
```

Рис. 26: n=10000

Output:
q= 0.5 in 5 x= 5
q= 0.1 in 5 x= 4
q= 0.5 in 5 x= 6
q= 0.1 in 7 x= 2

При больших э.ф.р. стремятся к теоритической функции распределения, эмпирические квантили так же стремятся к теоритическим по определению.

2.1.4. Построение гистрограммы и полигона частот

Строим полигоны частот и гистрограммы частот, на одном графике размещая функции 5 выборок данного объема и теоретическую функцию.
При увеличении n полигоны частот и гистограммы стремятся к теоретической функции плотности. Как можно наблюдать ниже на графиках.
-n=5

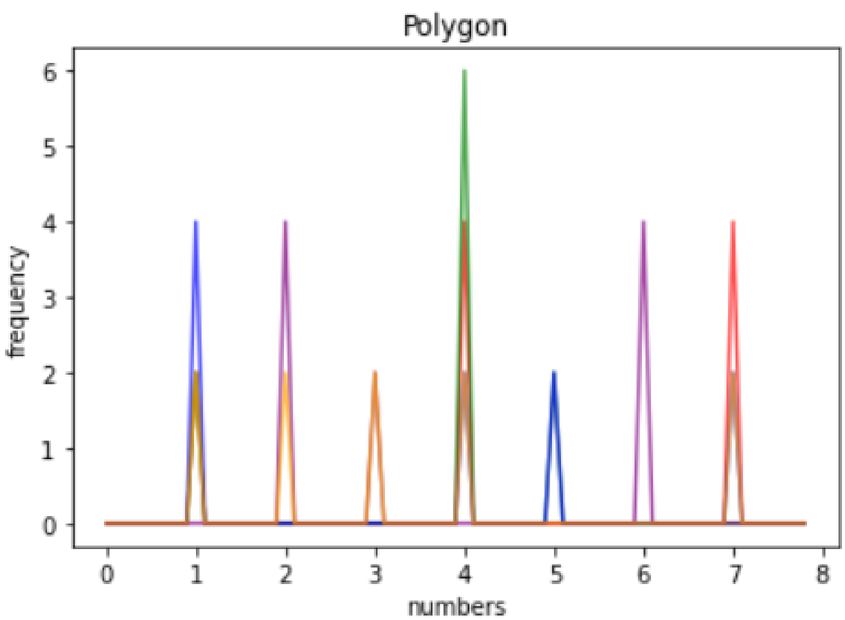

Рис. 27: n=5

-n=10

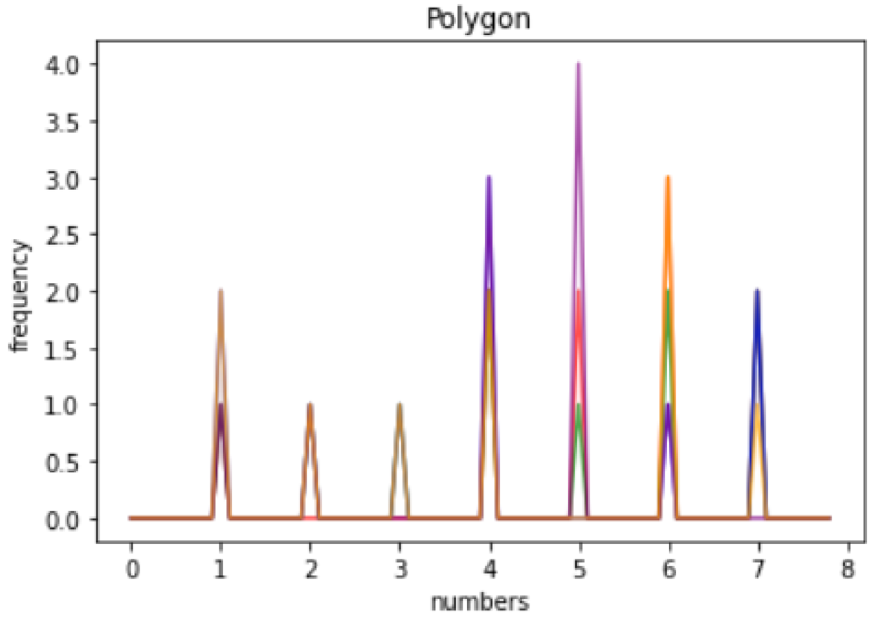

Рис. 28: n=10

-n=100

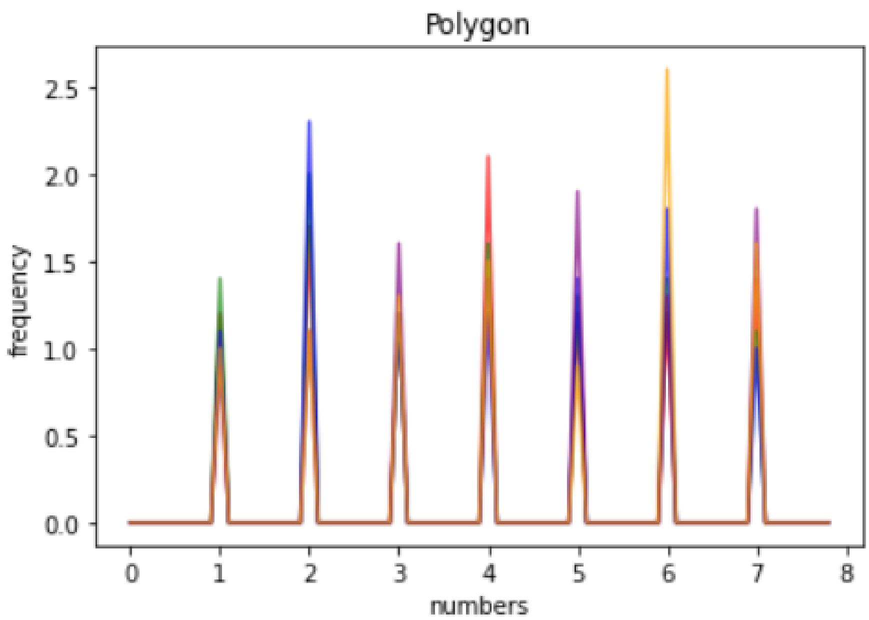

Рис. 29: n=100

-n=1000

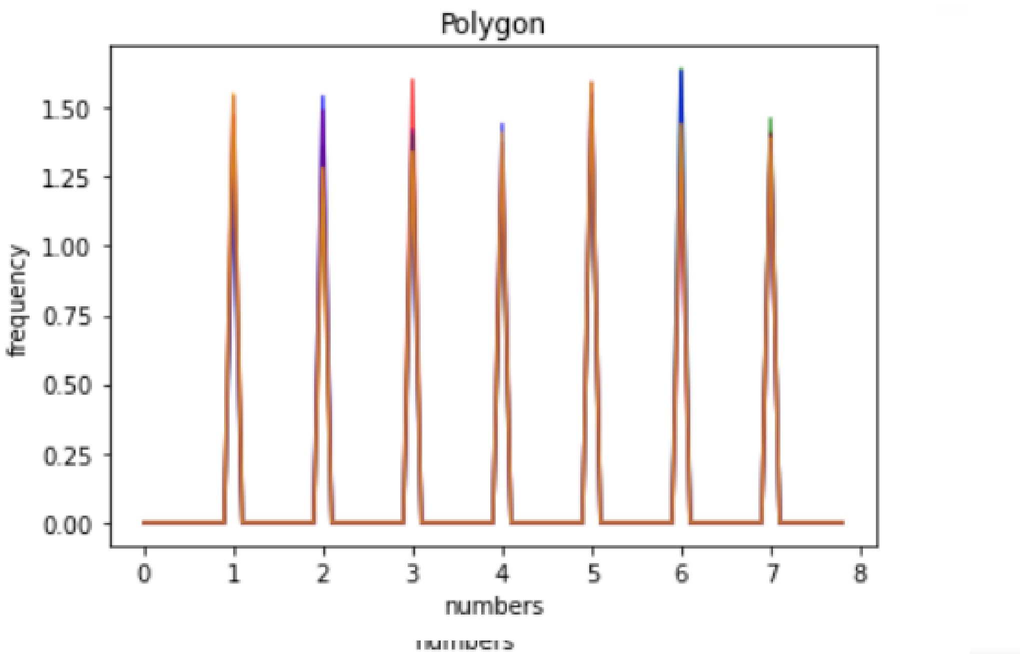

Рис. 30: n=1000

-n=10000
-n=5

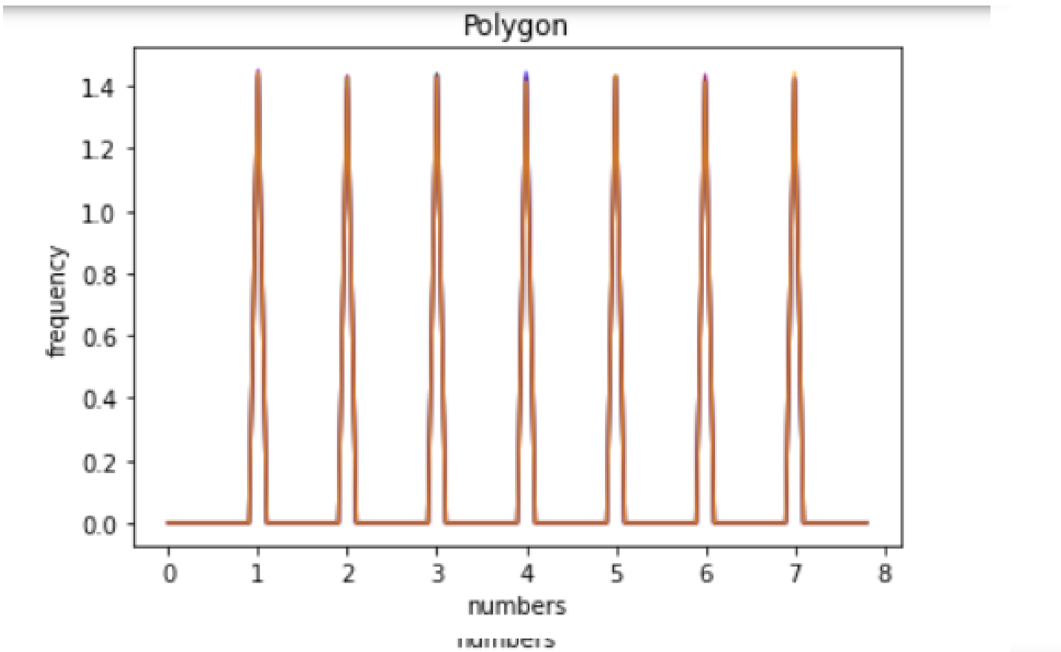

Рис. 31: n=10000

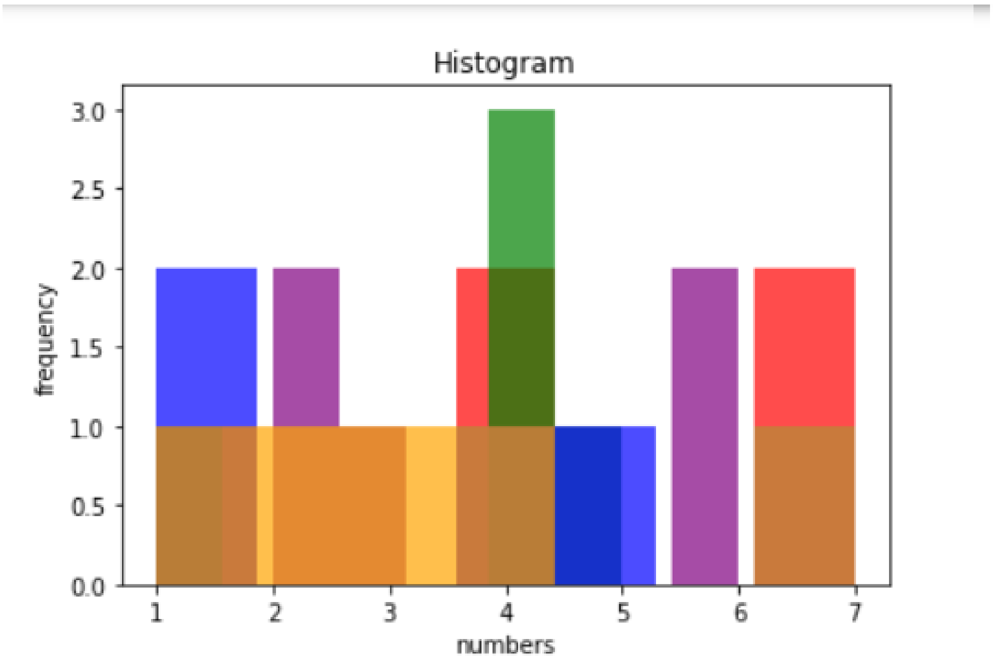

Рис. 32: n=5

-n=10

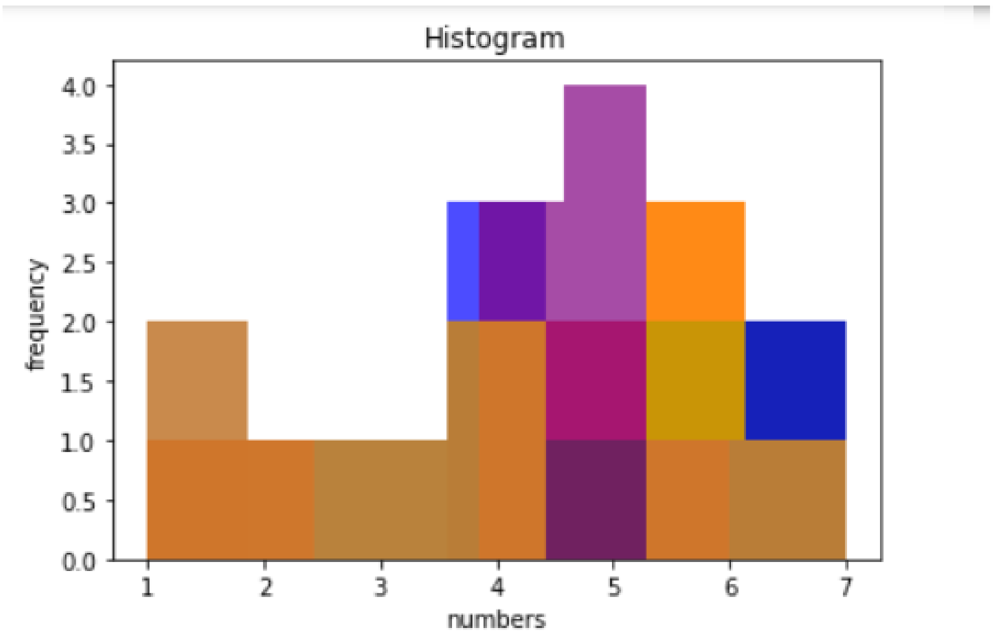

Рис. 33: n=10

-n=100

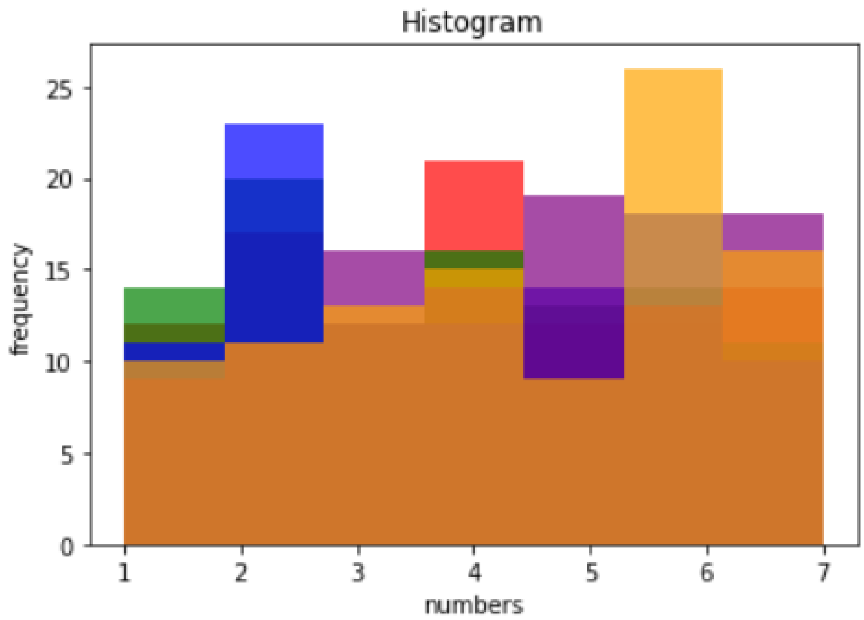

Рис. 34: n=100

-n=1000

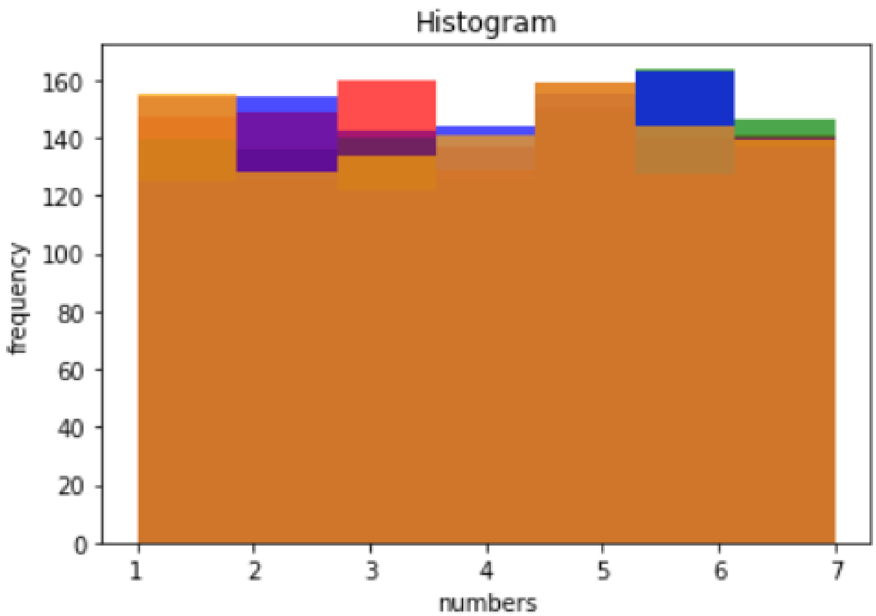

Рис. 35: n=1000

-n=10000

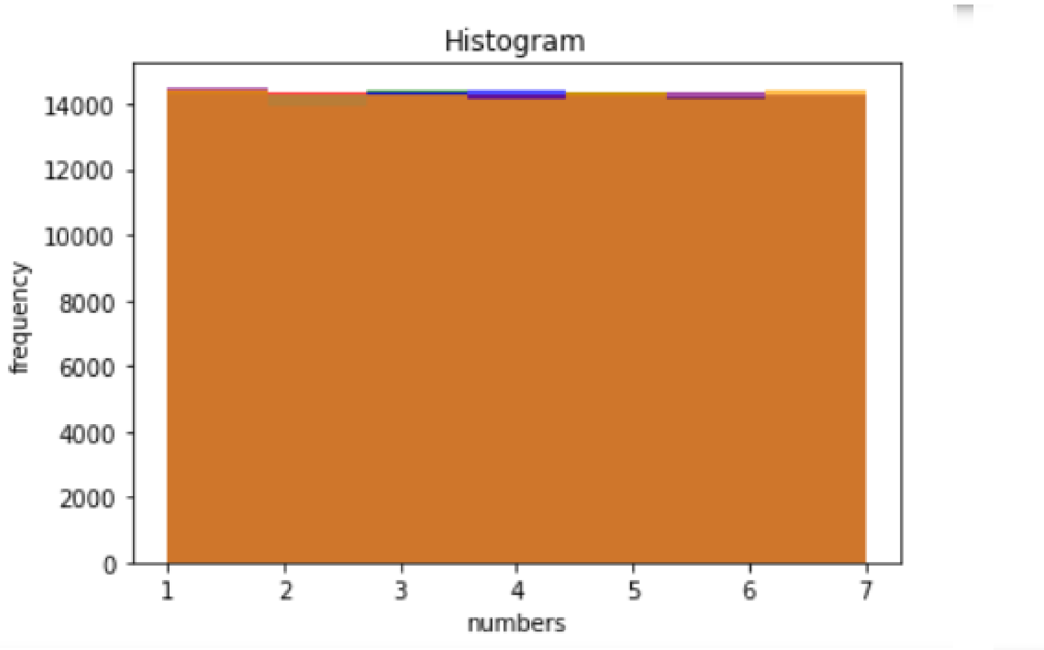

Рис. 36: n=10000

Code:

```
for i in range(0,5):
    color=colors[i%5]
    counts, bins = numpy.histogram(samples_list[i], bins=numpy.arange(0,8,0.1), density=True)
    pyplot.plot(bins[:-1], counts, color=color, alpha=0.7)
pyplot.xlabel('numbers')
pyplot.ylabel('frequency')
pyplot.title('Polygon')
pyplot.show()
```

Рис. 37: Код

```
for i in range(0,5):
    color=colors[i%5]
    counts, bins = numpy.histogram(samples_list[i], bins=numpy.arange(0,8,0.1), density=True)
    pyplot.plot(bins[:-1], counts, color=color, alpha=0.7)
pyplot.xlabel('numbers')
pyplot.ylabel('frequency')
pyplot.title('Polygon')
pyplot.show()
```

Рис. 38: Код

2.2. Экспоненциальное распределение

2.2.1. Моделирование выбранной случайной величины

На основе моделирования дискретного равномерного распределения, описанного в домашнем задании 1, создаем функцию, с помощью которой генерируем по 5 выборок каждого объема. Для примера напечатаны сгенерированные выборки объема n=5 b n=10.
Output:

-n=5
[0.16434301457281045, 0.6172450408927916, 0.6883199816464598, 2.7371544197817026, 5.673749863108269]
[0.04666232802146302, 0.426946266051896, 0.5528180465210837, 0.8334028818704013, 1.4603498342455614]
[0.2829500277473982, 0.2847671919813394, 0.7145741624776435, 0.8140387406630023, 1.374262248737471]
[0.7405392649830047, 0.9704575012125974, 1.0476765340252499, 1.19687704861772, 1.5401575489450705]
[0.602507792380593, 0.7592083198125391, 1.3862009444108174, 1.5361005698043353, 2.4003764038702555]

-n=10
[[0.017400241396125234, 0.24743305378966501, 0.30518229114622886, 0.38623221599747204, 0.7108156828644356, 0.7650492552734278, 1.1704413373311817, 1.1960967845087855, 1.7374010777243183, 2.710884863378893]
[0.2018400497477813, 0.3344536166156225, 0.36029107196831084, 0.473291146028304, 0.7962614045780094, 0.8546230897791269, 0.934767835867592, 1.3675060041415124, 1.7079823954725326, 2.468894921619211]
[0.05746105376568163, 0.06209887456824994, 0.4727269575729011, 0.4742272339308097, 0.691099822405431, 0.8582509004218759, 1.159334248258738, 1.5271811749149165, 1.64485827684547, 2.1699870785164004]
[0.033281864405029817, 0.06841803989290686, 0.5523037426270578, 1.2267826029047861, 1.3259955356400717, 1.3343707850394242, 1.4194334218672617, 1.5112018846427622, 2.089906092898997, 2.409332449393737]
[0.13312206471118385, 0.23808417916430138, 0.2916973253887163, 0.31876171786448915, 0.8331820947219716, 1.0959642763443254, 1.3737520761689035, 2.3992930609655176, 2.600199328472886, 3.713027024780754]

Код, с помощью которого генерируетсяпо 5 выборок каждого объема n=5, 10, 100, 1000, 10000. Итого 25 различных выборок для выбранного дискретного распределения.
Code:

```python
def gen_samples():
    volume_list = [5,10,100,1000,100000]
    samples_list = []
    for v in volume_list:
        for i in range(5):
            samples_list.append(generate(n=v))


    return(samples_list)

samples_list = gen_samples()

for i in range(10):
    print(samples_list[i])
```

Рис. 39: Код выборок n=5

```python
def gen_samples():
    volume_list = [5,10,100,1000,100000]
    samples_list = []
    for v in volume_list:
        for i in range(5):
            samples_list.append(generate(n=v))


    return(samples_list)

samples_list = gen_samples()

for i in range(10):
    print(samples_list[i])
```

Рис. 40: Код выборок n=10

2.2.2. Построение эмпирической функции распределения

Для каждой выборки строим эмпирическую функцию распределения. В отчете продемонстрировано построение эмпирической функции распределения для выборок размера 5. На одном графике отображены эмпирические функции распределения для 5 выборок данного объема и график функции распределения.

Code:

```python
def gen_ecdf(samples_list):
    ecdf_list = []
    for sample in range(25):
        ecdf = ECDF(samples_list[sample])
        ecdf_list.append((ecdf.x, ecdf.y))
    return(ecdf_list)

ecdf_list = gen_ecdf(samples_list)

for i in range(5):
    sample_ecdf = ecdf_list[i]
    print('ECDF F(x) for sample', i+1, ':')
    for k in range(len(sample_ecdf[0])-1):
        print(sample_ecdf[1][k], ', ', sample_ecdf[0][k], '< x <=', sample_ecdf[0][k+1])
    print(sample_ecdf[1][-1], ', x >', sample_ecdf[0][-1], '\n')
```

Рис. 41: Код

Output:

EECDF F(x) for sample 1 :
0.0 , -inf < x <= 0.16434301457281045
0.2 , 0.16434301457281045 < x <= 0.6172450408927916
0.4 , 0.6172450408927916 < x <= 0.6883199816464598
0.6000000000000001 , 0.6883199816464598 < x <= 2.7371544197817026
0.8 , 2.7371544197817026 < x <= 5.673749863108269
1.0 , x > 5.673749863108269

ECDF F(x) for sample 2 :
0.0 , -inf < x <= 0.04666232802146302
0.2 , 0.04666232802146302 < x <= 0.426946266051896
0.4 , 0.426946266051896 < x <= 0.5528180465210837
0.6000000000000001 , 0.5528180465210837 < x <= 0.8334028818704013
0.8 , 0.8334028818704013 < x <= 1.4603498342455614
1.0 , x > 1.4603498342455614

ECDF F(x) for sample 3 :
0.0 , -inf < x <= 0.2829500277473982
0.2 , 0.2829500277473982 < x <= 0.2847671919813394

0.4 , 0.2847671919813394 < x <= 0.7145741624776435
0.6000000000000001 , 0.7145741624776435 < x <= 0.8140387406630023
0.8 , 0.8140387406630023 < x <= 1.374262248737471
1.0 , x > 1.374262248737471

ECDF F(x) for sample 4 :
0.0 , -inf < x <= 0.7405392649830047
0.2 , 0.7405392649830047 < x <= 0.9704575012125974
0.4 , 0.9704575012125974 < x <= 1.0476765340252499
0.6000000000000001 , 1.0476765340252499 < x <= 1.19687704861772
0.8 , 1.19687704861772 < x <= 1.5401575489450705
1.0 , x > 1.5401575489450705

ECDF F(x) for sample 5 :
0.0 , -inf < x <= 0.602507792380593
0.2 , 0.602507792380593 < x <= 0.7592083198125391
0.4 , 0.7592083198125391 < x <= 1.3862009444108174
0.6000000000000001 , 1.3862009444108174 < x <= 1.5361005698043353
0.8 , 1.5361005698043353 < x <= 2.4003764038702555
1.0 , x > 2.4003764038702555

Code:

```
colors = ['red', 'green', 'blue', 'purple', 'orange']

for k in range(0,5):
    sample_ecdf = ecdf_list[k]
    print(sample_ecdf)
    color = colors[k%5]
    pyplot.plot([0,sample_ecdf[0][1]],[0,0], c=color, alpha=0.7, linewidth=4)
    for i in range(1,len(sample_ecdf[0])-1):
        pyplot.plot([sample_ecdf[0][i],sample_ecdf[0][i+1]],[sample_ecdf[1][i],sample_ecdf[1][i]],c=color, alpha=0.7,linewidth=4)
    pyplot.plot([sample_ecdf[0][-1], sample_ecdf[0][-1]+1],[1,1],c=color, alpha=0.7, linewidth=4)

x = numpy.linspace(0,4,100)
pyplot.plot(x, y:=1-numpy.exp(-1*x),'b')
pyplot.show()
(x, y:=numpy.exp(-1*x),'b')
pyplot.show()
```

Рис. 42: Код

Output:

-n=5

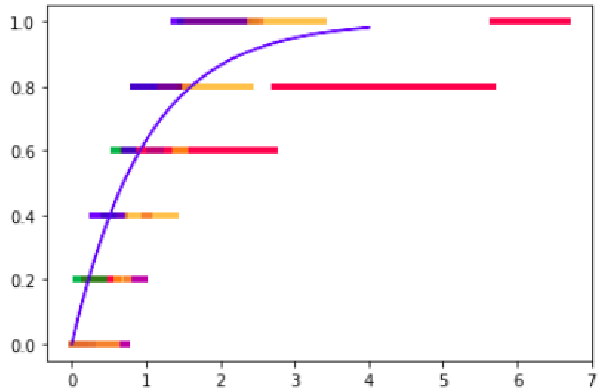

Рис. 43: n=5

-n=10

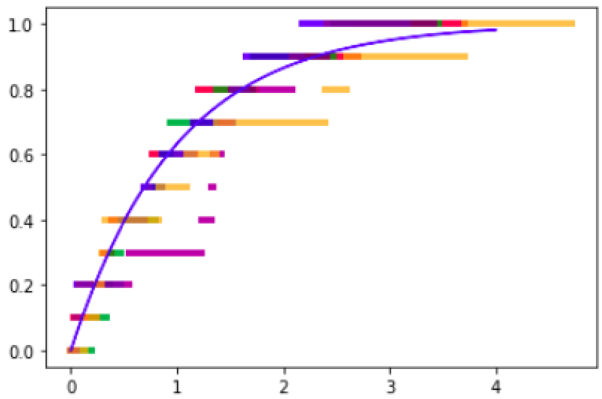

Рис. 44: n=10

-n=100

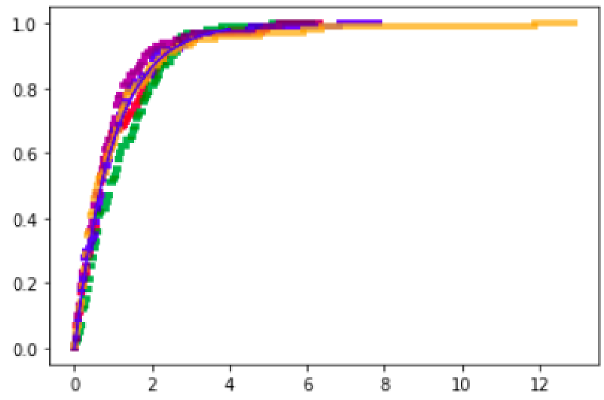

Рис. 45: n=100

-n=1000

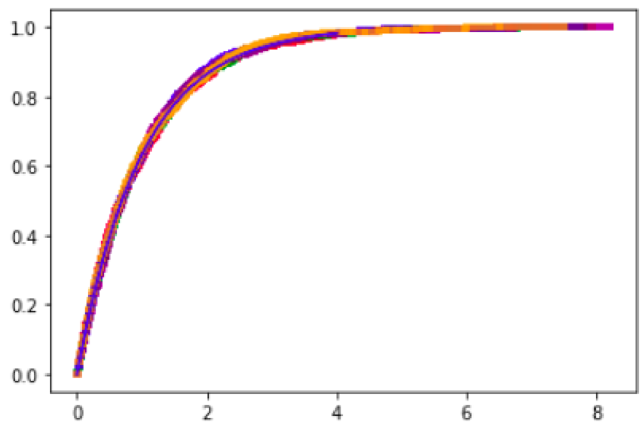

Рис. 46: n=1000

-n=10000

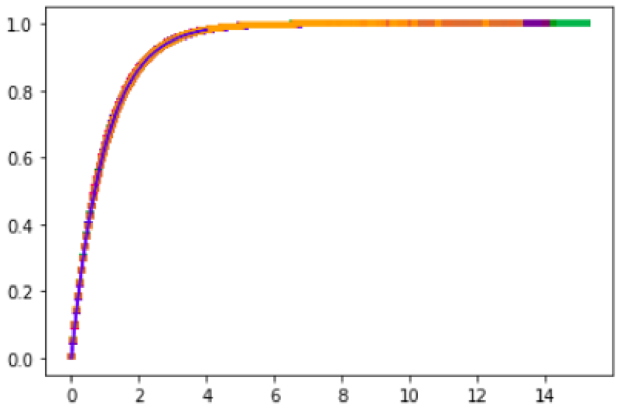

Рис. 47: n=10000

Существует училенный закон больших чисел, в котором эмпирическая функция почти наверное к теоретической функции: $\hat{F}x) \to F(x)$ почти наверное при $n \to \infty$. По данным, которые представлены на графике видно, что совпадение графика эмпирической функции с графиком теоретической функции распределения зависит от объема выборки n.Чем оно больше, тем больше совпадают графики.
Для каждого n найдем максимальную точную верхнюю границу разности каждой пары эмпирических функций распределения среди всех возможных пар. Для Каждой пары рассмотрим определенное количество вариантов, равное C_5^2
Точная верхняя граница разности - это наибольшее число разности значений $ECDF$ функций при одинаковых аргументах. Данное число можно вычислить при помощи кода представленного ниже.

Code:

```
diff_list=[]
for i in range(4):
    for k in range(i+1,10):
        sample1=ecdf_list[i+5*n]
        sample2=ecdf_list[k+5*n]
        for j in range(1,len(sample1[0])-1):
            pair1 = [sample1[0][j],sample1[0][j+1]]
            for p in range(1,len(sample2[0])-1):
                pair2=[sample2[0][p],sample2[0][p+1]]
                if ((pair2[0]<=pair1[0] and pair2[1] >= pair1[0]) or (pair2[0]<=pair1[1] and pair2[1]>=pair1[1])):
                    diff_list.append(abs(sample1[1][j]-sample2[1][p]))
print(max(diff_list))

var_list = []
```

Рис. 48: Код

Output:
n=5–0.6000000000000001
n=10–0.6000000000000001
n=100–0.33
n=1000–0.16599999999999998
n=100000–0.14641999999999994
Точные верхнии границы разности будут уменьшаться при приближении эмпирической функции к теоретической с ростом n.

2.2.3. Построение вариационного ряда выборки

Вариационный ряд - набор элементов X_1,...,X_n, упорядоченный по возрастанию. Ниже представлен код, с помощью которого строятся вариационные ряды для каждой сгенерированной выборки. В отчете представлены вариационные ряды для выборок объема $n = 5$, $n = 10$.

Code:

```
var_list = []

for sample in samples_list:
    var_sample = sorted(sample)
    var_list.append(var_sample)

for i in range(10):
    print(var_list[i])

```

Рис. 49: n=10000

Output:

-$n = 5$:
[3, 3, 3, 4, 5]
[1, 2, 4, 7, 7]
[1, 5, 5, 6, 6]
[1, 3, 5, 5, 5]
[2, 6, 7, 7, 7]

-$n = 10$:
[1, 1, 3, 4, 4, 4, 4, 5, 7, 7]
[2, 2, 4, 4, 4, 5, 6, 6, 7, 7]
[2, 4, 4, 4, 5, 6, 6, 6, 7, 7]
[1, 1, 2, 2, 2, 2, 2, 2, 4, 6]
[1, 1, 1, 2, 3, 3, 4, 4, 5, 7]

α-квантиль случайной величины ξ с функцией распределения $F_\xi(x)$=$P(\xi < x)$-любое число x_α, которое удовлетворяет двум условиям:
$F_\xi(x_\alpha) \leq \alpha$
$F_\xi(x_\alpha + 0) \geq \alpha$

Code:

```
quantile_list=[0.1,0.5,0.7]

for element in ecdf_list:
    for quantile in quantile_list:
        if quantile in element[1]:
            q = element[1].index(quantile)
            print('q=', quantile, 'in', len(element[1])-1, 'x=', element[0][q+1])
```

Рис. 50: n=10000

Output:
q= 0.1 in 10 x= 0.6413320745799311
q= 0.5 in 10 x= 0.910640008327763
q= 0.1 in 10 x= 0.12399528221711266
q= 0.5 in 10 x= 0.7906900254239998
q= 0.1 in 10 x= 0.3620208678055566
q= 0.5 in 10 x= 0.812727627733623
q= 0.1 in 10 x= 0.18279315356500137
q= 0.5 in 10 x= 1.1445950153100095
q= 0.1 in 10 x= 0.11689336921577874
q= 0.5 in 10 x= 1.3870231080405808
q= 0.5 in 100 x= 0.7934820621085901
q= 0.5 in 100 x= 0.793073541539919
q= 0.5 in 100 x= 0.8403326253990359
q= 0.5 in 100 x= 0.9128728846620282
q= 0.5 in 100 x= 0.6371726007486764
q= 0.1 in 1000 x= 0.10843256891167612
q= 0.5 in 1000 x= 0.7037936174682455
q= 0.1 in 1000 x= 0.10181855264122328
q= 0.5 in 1000 x= 0.6612883835095232
q= 0.1 in 1000 x= 0.09519690655230126
q= 0.5 in 1000 x= 0.6150171893581541
q= 0.1 in 1000 x= 0.1104187302789945
q= 0.5 in 1000 x= 0.7101891924275118
q= 0.1 in 1000 x= 0.10293627298446764
q= 0.5 in 1000 x= 0.6644350911354294
q= 0.1 in 100000 x= 0.10616135390540682
q= 0.5 in 100000 x= 0.6956843876851393
q= 0.1 in 100000 x= 0.10417649816429929
q= 0.5 in 100000 x= 0.6912307711709078
q= 0.1 in 100000 x= 0.10433782019034865
q= 0.5 in 100000 x= 0.69497424308664
q= 0.1 in 100000 x= 0.10565314554016611
q= 0.5 in 100000 x= 0.694595466157049
q= 0.1 in 100000 x= 0.1054678901206238
q= 0.5 in 100000 x= 0.6907414403728113

2.2.4. Построение гистрограммы и полигона частот

Строим полигоны частот и гистрограммы частот, на одном графике размещая функции 5 выборок данного объема и теоретическую функцию.
При увеличении n полигоны частот и гистограммы стремятся к теоретической функции плотности. Как можно наблюдать ниже на графиках.

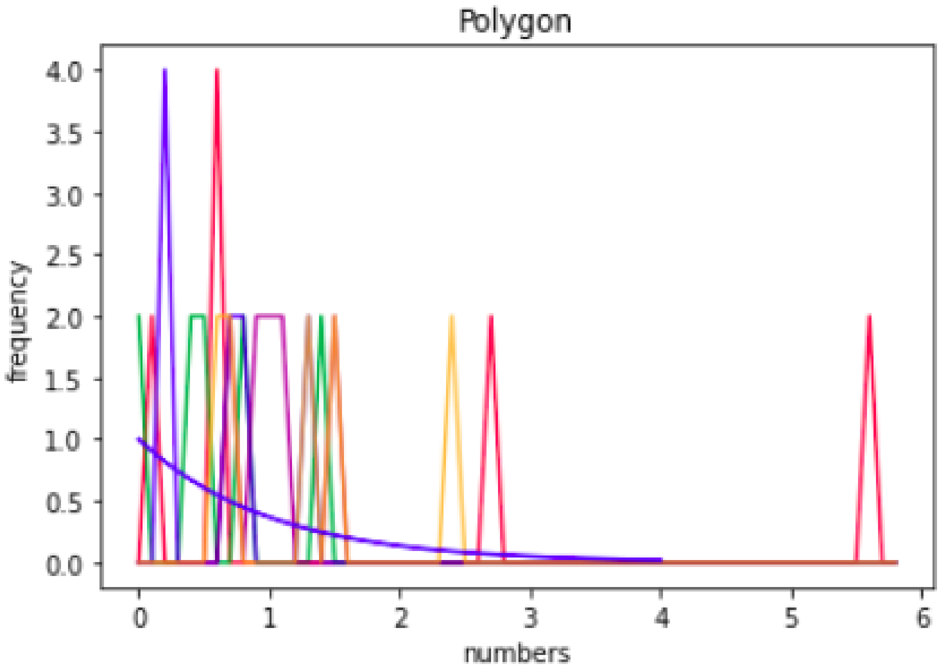

Рис. 51: n=5

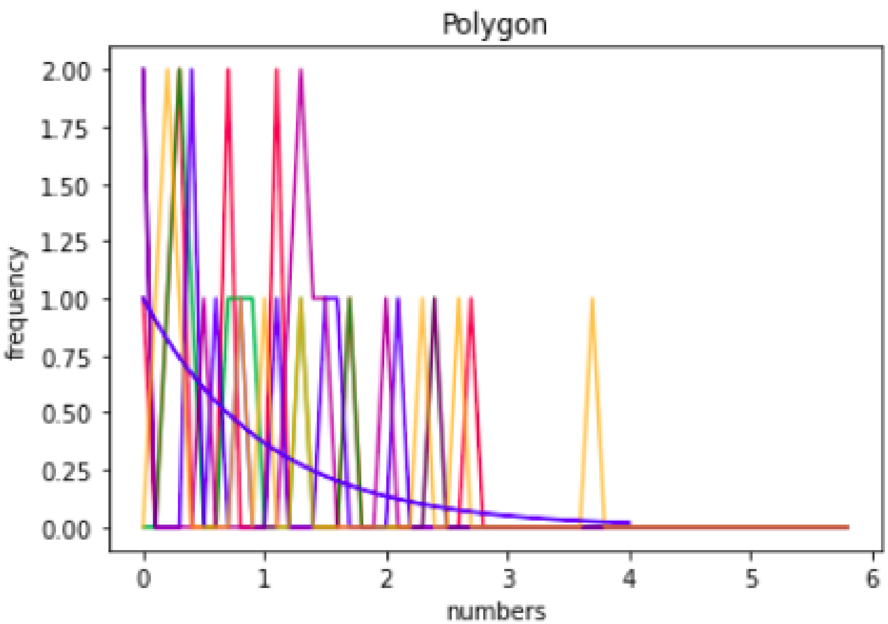

Рис. 52: n=10

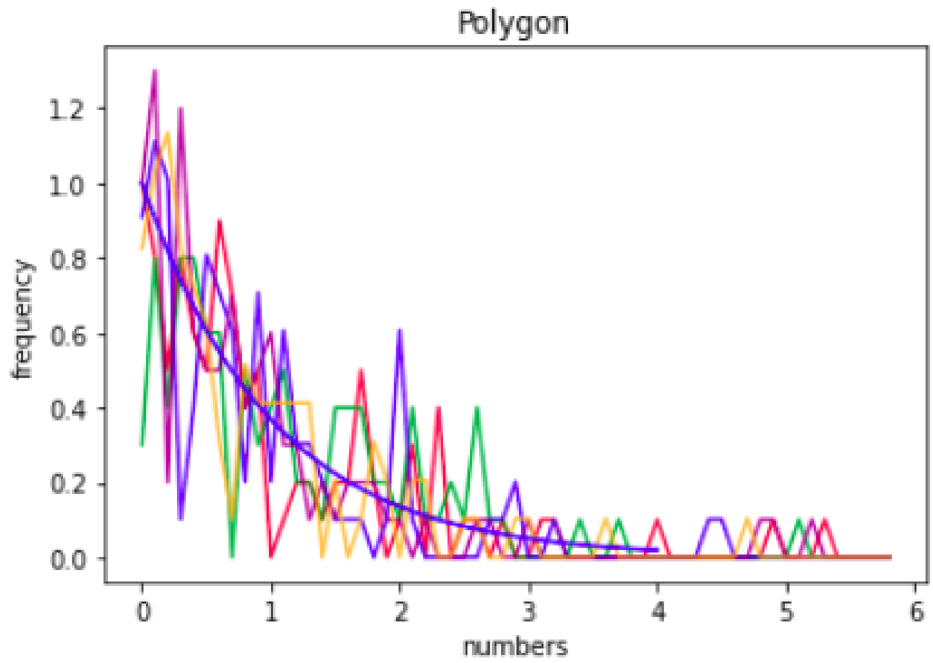

Рис. 53: n=100

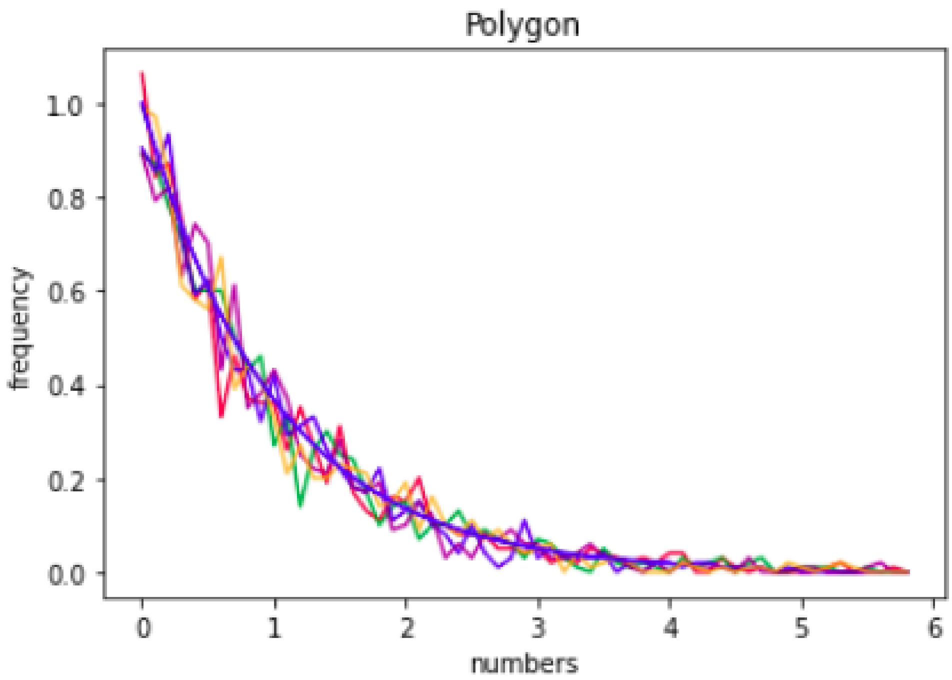

Рис. 54: n=1000

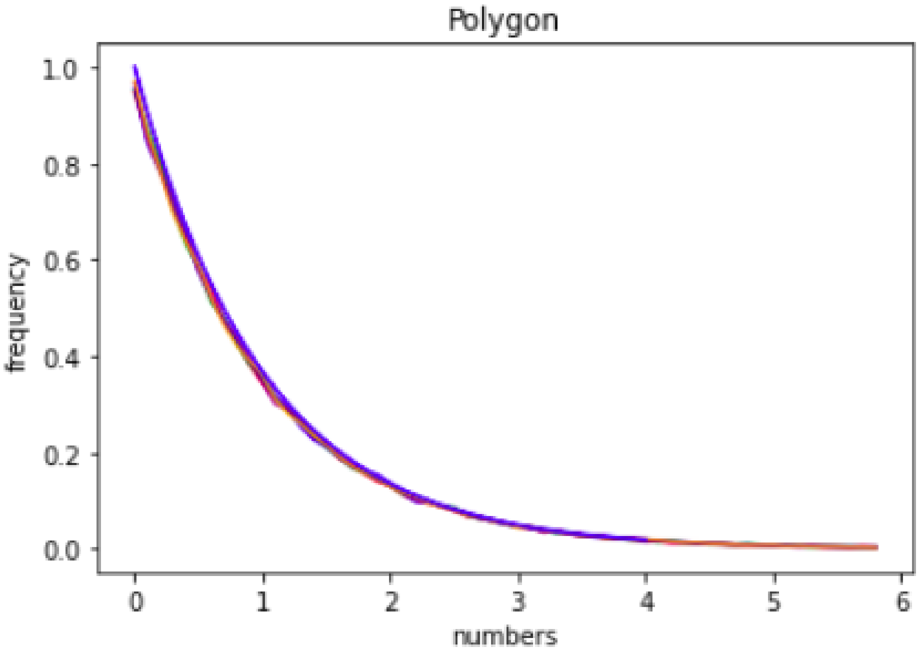

Рис. 55: n=10000

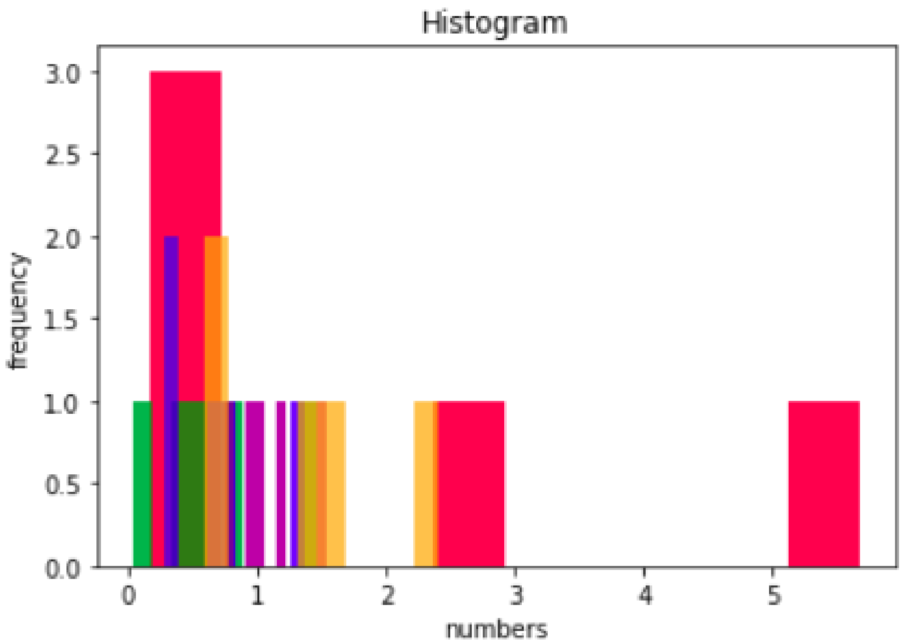

Рис. 56: n=5

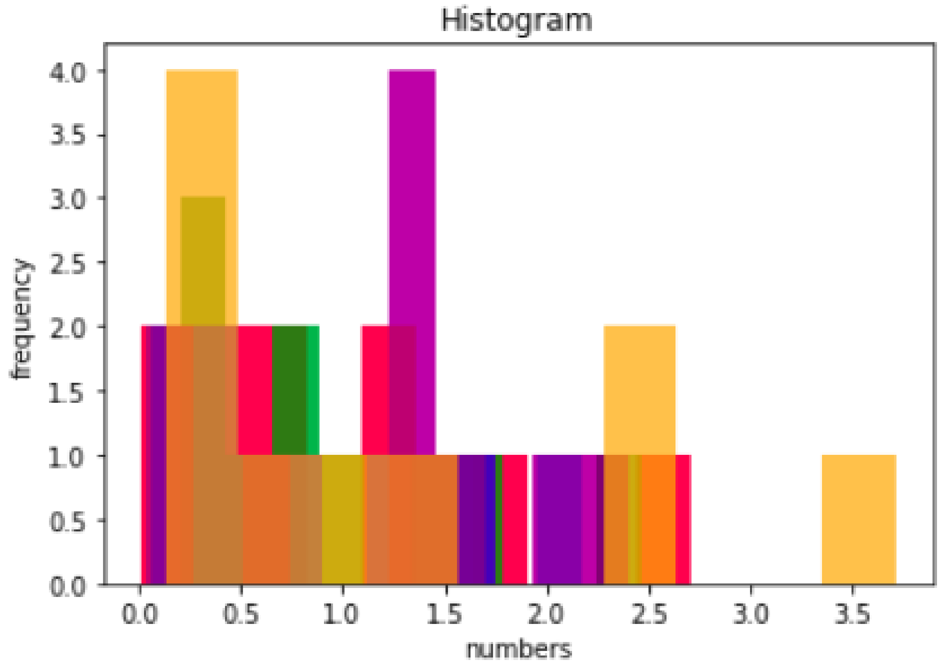

Рис. 57: n=10

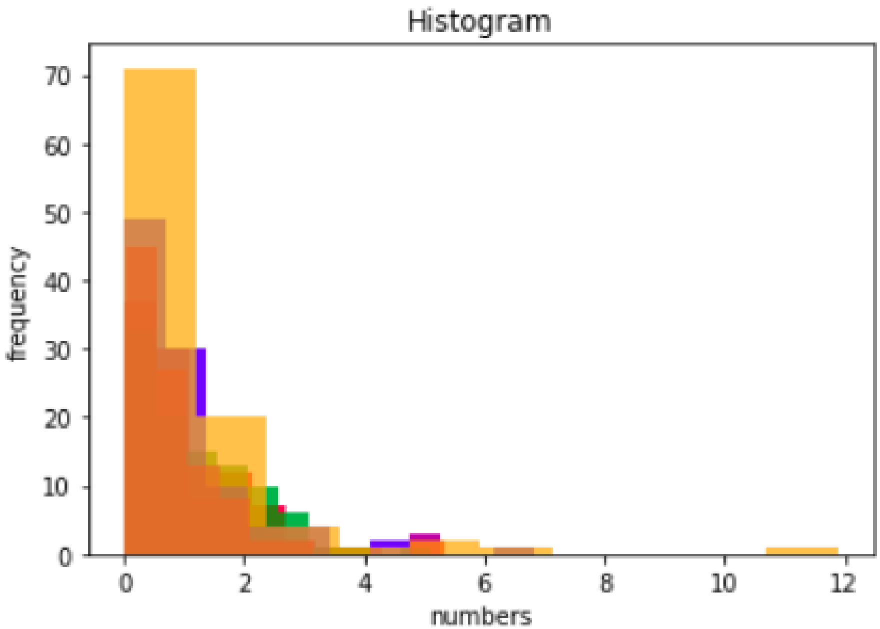

Рис. 58: n=100

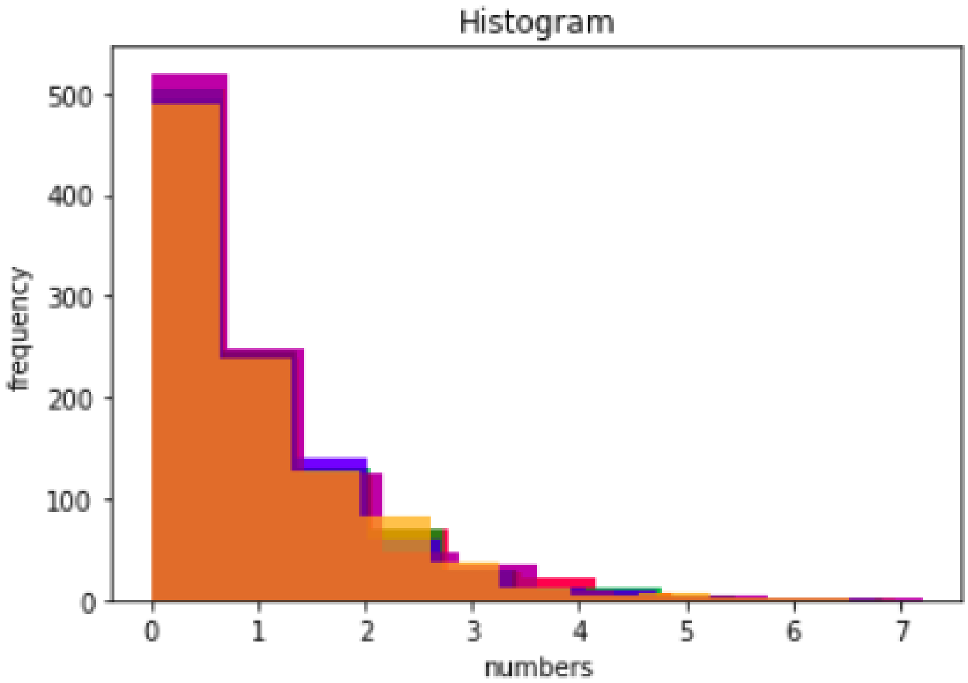

Рис. 59: n=1000

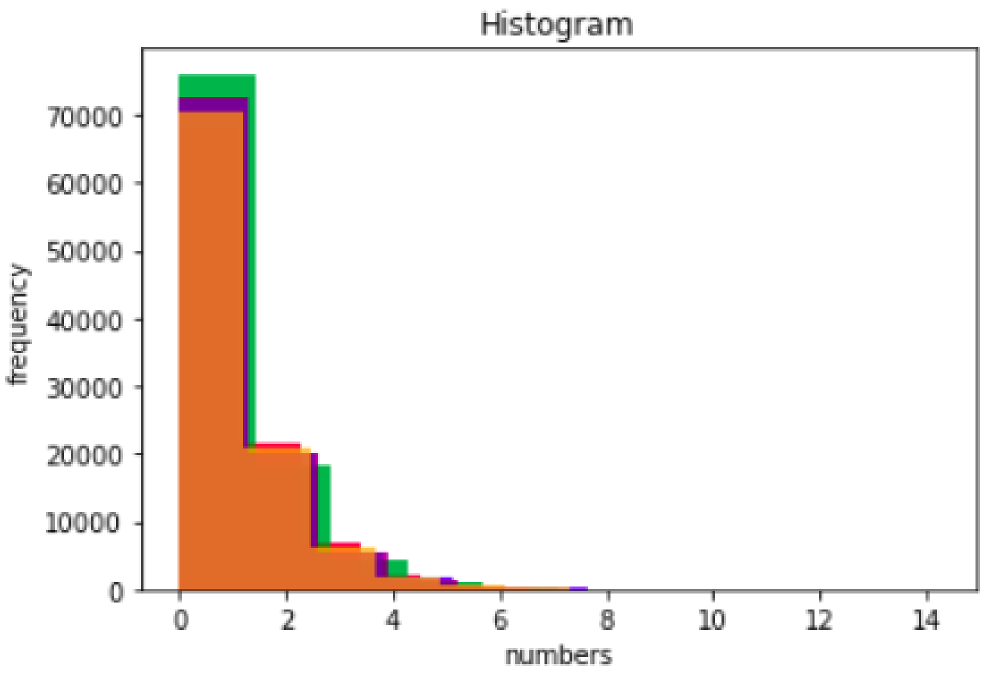

Рис. 60: n=10000

Code:

```
colors = ['red', 'green', 'blue', 'purple', 'orange']

for i in range(0,5):
    color = colors[i%5]
    sample = samples_list[i]
    pyplot.hist(sample,bins=10, color=color, alpha=0.7)
pyplot.xlabel('numbers')
pyplot.ylabel('frequency')
pyplot.title('Histogram')
pyplot.show()
```

Рис. 61: Код

```
for i in range(10,15):
    color=colors[i%5]
    counts, bins = numpy.histogram(samples_list[i], bins=numpy.arange(0,6,0.1), density=True)
    pyplot.plot(bins[:-1], counts, color=color, alpha=0.7)

    x = numpy.linspace(0,4,100)
pyplot.xlabel('numbers')
pyplot.ylabel('frequency')
pyplot.title('Polygon')
pyplot.plot(x, y:=numpy.exp(-1*x),'b')
pyplot.show()
```

Рис. 62: Код

3. Домашнее задание. Оценки

3.1. Дискретное равномерное распределение

3.1.1. Нахождение выборочного среднего и выборочной дисперсии

Наиболее важными характеристиками случайной величины ξ являются ее моменты $= E\,\xi^k$, а также центральные моменты $\mu^k = E(\xi - \alpha_1)^k$ (когда они существуют). Их статистическими аналогами, вычисляемыми по соответствующей выборки $X = (X_1, ..., X_n)$, являются выборочные моменты соответсвенно обычные:

$$\hat{\alpha}_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k$$

и центральные:

$$\hat{\mu}_k = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\alpha}_1)^k$$

$\hat{\alpha}_1$ (принято обозначать как $\bar{X}$) называется выборочным средним, $\hat{\mu}_2$ - выборочной дисперсией. Таким образом, выборочное среднее и выборочная дисперсия являются статистическими аналогами теоритических среднего (матетматического ожидания) $E\xi$ и дисперсии $D\xi$, когда они существуют.

Для каждой выработанной ранее выборки найдем выборочное среднее и выборочную дисперсию.

Приведем только выборки размера 5 и 10, для проверки правильности вычислений:

n=5
[4, 3, 7, 2, 2]
[2, 1, 7, 2, 6]
[2, 2, 5, 1, 6]
[5, 7, 5, 6, 1]
[4, 2, 7, 6, 1]

n=10
[5, 4, 3, 5, 4, 1, 1, 7, 6, 1]
[2, 2, 6, 4, 6, 6, 7, 1, 3, 1]
[3, 4, 1, 2, 1, 4, 5, 7, 5, 4]
[2, 6, 2, 6, 4, 3, 2, 4, 3, 7]
[5, 3, 7, 4, 3, 1, 4, 4, 6, 6]

Создаем функции для вычисления выборочных среднего и дисперсии и применяем их к сгенерированным выборкам распределения.

Выборочное среднее:

```python
for i in range(15,20):
    sum_el = 0
    for item in var_list[i]:
        sum_el = sum_el+item
    raz_var = len(var_list[i])
    it_sr = sum_el/raz_var
    srvr_list.append(it_sr)
    print("Выборочное среднее - ", it_sr)
```

Рис. 63: Код

n=5
Выборочное среднее - 3.6
Выборочное среднее - 3.6
Выборочное среднее - 3.2
Выборочное среднее - 4.8
Выборочное среднее - 4.0

n=10
Выборочное среднее - 3.7
Выборочное среднее - 3.8
Выборочное среднее - 3.6
Выборочное среднее - 3.9
Выборочное среднее - 4.3

n=100
Выборочное среднее - 3.69
Выборочное среднее - 4.27
Выборочное среднее - 3.95
Выборочное среднее - 4.03
Выборочное среднее - 4.43

n=1000
Выборочное среднее - 3.959
Выборочное среднее - 3.95
Выборочное среднее - 4.016
Выборочное среднее - 3.993
Выборочное среднее - 4.014

n=10000
Выборочное среднее - 4.00641
Выборочное среднее - 4.00459
Выборочное среднее - 3.99936
Выборочное среднее - 4.00149

Выборочное среднее - 3.99684

Исправлено:

С помощью формул выведенных в первом домашнем задании посчитаем математическое ожидание для N=8:

$$M\xi = \frac{N+1}{2} = \frac{9}{2} = 4,5$$

Выборочное среднее стремится к истинному значению математического ожидания при увеличении объема выборки. И это обусловленно свойством состоятельности нашей оценки, которое в свою очередь обучловлено законом больших чисел. Покажем это при помощи закона больших чисел Хинчина:

$$\frac{\xi_1+\xi_2+...+\xi n-1+\xi n}{n} \xrightarrow{P} M\xi$$

Если существует $M|\xi_1|$ тогда:

$$\overline{X_n} \xrightarrow{P} M\xi, n \to \infty$$

Получается, что по закону больших чисел Хинчина оценка является состоятельной. Выборочное среднее обладает свойством несмещенности.Тогда покажем, что математическое ожидание от оценки будет равно истинному математическому ожиданию:

$$M\overline{X} = M(\frac{1}{n}\sum_{i=1}^{n} M(X_i)) = \frac{1}{n} \cdot n \cdot M\xi = M\xi$$

Мы показали, что эта оценка является несмещенной.

Выборочная дисперсия вычисляется с помощью кода представленного ниже:

```
for i in range(20,25):
    sum_el = 0
    sum_el_dic = 0
    for item in var_list[i]:
        sum_el = sum_el+item
    for item in var_list[i]:
        sum_el_dic = sum_el_dic+item*item
    raz_var = len(var_list[i])
    it_sr = sum_el/raz_var
    it_sr_dic = sum_el_dic/raz_var-it_sr*it_sr
    print("Выборочная дисперсия - ", it_sr_dic)

```

Рис. 64: Код

n=5
Выборочная дисперсия - 3.4399999999999977
Выборочная дисперсия - 5.84
Выборочная дисперсия - 3.7599999999999998
Выборочная дисперсия - 4.16
Выборочная дисперсия - 5.199999999999999

n=10
Выборочная дисперсия - 4.209999999999997
Выборочная дисперсия - 4.76
Выборочная дисперсия - 3.2399999999999984
Выборочная дисперсия - 3.0900000000000016
Выборочная дисперсия - 2.8100000000000023

n=100
Выборочная дисперсия - 4.0139
Выборочная дисперсия - 3.9771000000000036
Выборочная дисперсия - 4.407500000000001
Выборочная дисперсия - 4.1290999999999976
Выборочная дисперсия - 3.7051000000000016

n=1000
Выборочная дисперсия - 3.8253189999999986
Выборочная дисперсия - 3.707499999999998
Выборочная дисперсия - 4.015743999999998
Выборочная дисперсия - 4.200951
Выборочная дисперсия - 4.077804

n=10000
Выборочная дисперсия - 4.009068911900002
Выборочная дисперсия - 4.002488931899997
Выборочная дисперсия - 3.993259590400003
Выборочная дисперсия - 4.020787779899997
Выборочная дисперсия - 4.0198900144

Исправлено:

Для произведения сравнения значений параметров и их оценок, посчитаем численно дисперсию по формуле, выведенной в первом домашнем задании для N=8:

$$D\xi = \frac{N^2-1}{12} = \frac{64-1}{12} = 5,25$$

По полученным результатом выборочной дисперсии, мы можем видеть, что их значения стремятся к истинному значению $D\xi$ при росте объема выборки. Это обусловленно свойством состоятельности нашей оценки.

Воспользуемся доказанным свойством состоятельности выборочного среднего, как оценки и законом больших чисел.Выразим выборочную дисперсию.

$$\hat{\mu_2} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}^2 = \frac{1}{n}\sum_{i=1}^{n}X_i^2 - 2\bar{X}\frac{1}{n}\sum_{i=1}^{n}X_i + \frac{1}{n}\sum_{i=1}^{n}\bar{X}^2 = \bar{X^2} - \bar{X}^2$$

$$\overline{X^2} \xrightarrow{P} M\xi^2$$

$$\overline{X}^2 \xrightarrow{P} (M\xi)^2, n \to \infty$$

Выводы по закону больших чисел. Тогда:

$$\overline{X^2} - \overline{X}^2 \xrightarrow{P} M\xi^2 - (M\xi)^2 = D\xi, n \to \infty$$

Из этого следует, что оценка является состоятельной.

Проведем преобразования:

$$M\hat{\mu_2} = M(\overline{X^2} - \overline{X}^2) = M\overline{X^2} - M\overline{X}^2$$
$$M\xi^2 - M\overline{X}^2 = M\xi^2 - ((M\overline{X})^2 + D\overline{X})$$
$$M\xi^2 - (M\xi)^2 - D(\tfrac{1}{n}\sum_{i=1}^{n} X_i) = \mu_2 - \tfrac{1}{n^2}n\mu^2 = \tfrac{n-1}{n} \cdot \mu_2$$

Таким образом, мы показали что выборочная дисперсия не является несмещенной оценкой. Но путем преобразований $M\bar{\mu_2} = \frac{n-1}{n} \cdot \mu_2 \to M(\frac{n}{n-1} \cdot \hat{\mu_2}) = \mu_2$.Такая оценка будет несмещенной оценкой. Так как $\frac{n}{n-1} \to 1$ при $n \to \infty$, тогда данная оценка будет состоятельной.

3.1.2. Нахождение параметров распределений событий

Для дискретного равномерного распределения необходимо проверить условие регулярности, проверить пренадлежит ли оно экспоненциальному семейству, предложить оценку X для оцениваемого параметра θ и выполним следующее:

- Проверить, является предложенная оценка несмещенной
- Проверить, является ли предложенная оценка X состоятельной для оцениваемого параметра θ
- Проверить, является ли предложенная оценка оптимальной, эффективной. Если нет, то построить оптимлаьную оценку для параметра θ

Согласно условиям регулярности, дискретное равномерное распределение не относится к этой категории регулярных распределений.

Найдем оценку параметра θ^* методом моментов($M[x] = \bar{X}$). $M[x] = \frac{\theta+1}{2}$.

$$1 + \theta = 2M[x]$$

$$(M[x] = \bar{X}) -> \theta = 2\bar{X} - 1$$

Дискретное равномерное распределение не принадлежит экспоненциальному семейству, так распределение может быть распределено на разных границах.

Статистика $T = T(X)$ называется несмещенной оценкой для заданной параметрической функции, если она удовлетворяет условию:

$$M_\theta T(x) = \tau(\theta)$$

Которое называется уравнением несмещенности.

$$M[\hat{\theta}] = M[2\bar{X} - 1]$$

$$M[2\frac{1}{n}\sum_{i-1}^{n} X_i - 1] = \frac{2}{n}\sum_{i=1}^{n} MX_i - 1 = 2MX_i - 1 = 2(\frac{\theta+1}{2}) - 1 = \frac{2+2\theta-2}{2} = \frac{2\theta}{2} = \theta$$

Статистика $T_n = T_n(X)$ называется состоятельной оценкой для неизвестного параметра $\tau(\theta)$ случайной величины ξ, если при $n \to \infty$

$$T_n(x) \xrightarrow[n\to\infty]{P} \tau(\theta)$$

$$\hat{\theta} = 2\bar{X} - 1$$

$$2MX_1 - 1 = 2(\frac{1+\theta}{2} - 1) = \frac{2+2\theta-2}{2} = \frac{2\theta}{2} = \theta$$

Исправлено:

Построим оптимальную оценку. Так как максимальная порядковая статистика X_n, построим ее полоноту:

$$M_\varphi(X_n) = \sum_{i=1}^{\theta} \varphi(i)((\frac{i}{\theta})^n - (\frac{i-1}{\theta})^n)$$

Данное условие будет выполняться при условии, если $\varphi = 0$. Это показывает, что наша достаточная статистика будет являться полной. Тогда из этого следует, что функция φ от нашей достаточной полной статистики будет являться оптимальной оценкой для ее математического ожидания.

$$P(X_n \leq t) = (\frac{t}{\theta})^n$$

$$P(X_n = t) = P(X_n \leq t) - P(X_n \leq t-1) = (\frac{t}{\theta})^2 - (\frac{t-1}{\theta})^4$$

$$M_\varphi(X_n) = \sum_{i=1}^{\theta} \varphi(i)((\frac{i}{\theta})^n - (\frac{i-1}{\theta})^n)$$

Займемся тем, что подберем φ для данного выражения. Тогда получим оценку для параметра θ:

$$\frac{(X_{(n)}^{n+1} - (X_{(n)}-1)^{n+1})}{(X_{(n)}^{n} - (X_{(n)}-1)^{n})}$$

Эффективные оценки существуют только в моделях экспоненциального семейства распределений. Выше было доказано, что дискретное равномерное распределение не принадлежит экспоненциальному семейству. Из-за этого эффективную оценку дискретного равномерного распределения найти невозможно.

Рассмотрим значения оценки по нашим выборкам, которые мы моделировали выше. Применим код написанные на Python:

```
for sample in var_list:
  sum=0
  for element in sample:
    sum += element
  print("Объем выборки = ", len(sample),',','параметр = ', 2*(sum/len(sample))-1)
```

Рис. 65: Код

Output:
Объем выборки = 5 , параметр = 8.6
Объем выборки = 5 , параметр = 5.0
Объем выборки = 5 , параметр = 8.2
Объем выборки = 5 , параметр = 3.4000000000000004
Объем выборки = 5 , параметр = 5.8
Объем выборки = 10 , параметр = 7.4
Объем выборки = 10 , параметр = 6.8
Объем выборки = 10 , параметр = 7.4
Объем выборки = 10 , параметр = 8.0
Объем выборки = 10 , параметр = 8.6
Объем выборки = 100 , параметр = 7.58
Объем выборки = 100 , параметр = 7.02
Объем выборки = 100 , параметр = 6.8
Объем выборки = 100 , параметр = 6.38
Объем выборки = 100 , параметр = 6.16
Объем выборки = 1000 , параметр = 6.998
Объем выборки = 1000 , параметр = 6.956
Объем выборки = 1000 , параметр = 7.01
Объем выборки = 1000 , параметр = 7.052
Объем выборки = 1000 , параметр = 6.954
Объем выборки = 100000 , параметр = 7.015040000000001
Объем выборки = 100000 , параметр = 7.0138
Объем выборки = 100000 , параметр = 7.005599999999999
Объем выборки = 100000 , параметр = 6.99758
Объем выборки = 100000 , параметр = 7.010960000000001

Таким образом можно наблюдать свойста состоятельности оценки.

3.2. Экспоненциальное распределение

3.2.1. Нахождение выборочного среднего и выборочной дисперсии

Наиболее важными характеристиками случайной величины ξ являются ее моменты $= E$ ξ^k, а также центральные моменты $\mu^k = E(\xi - \alpha_1)^k$ (когда они существуют). Их статистическими аналогами, вычисляемыми по соответствующей выборки $X = (X_1, ..., X_n)$, являются выборочные моменты соответсвенно обычные:

$$\hat{\alpha}_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k$$

и центральные:

$$\hat{\mu}_k = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\alpha}_1)^k$$

$\hat{\alpha}_1$ (принято обозначать как $\bar{X}$) называется выборочным средним, $\hat{\mu}_2$ - выборочной дисперсией. Таким образом, выборочное среднее и выборочная дисперсия являются статистическими аналогами теоритических среднего (матетматического ожидания) $E\xi$ и дисперсии $D\xi$, когда они существуют.

Для каждой выработанной ранее выборки найдем выборочное среднее и выборочную дисперсию.

Приведем только выборки размера 5 и 10, для проверки правильности вычислений:

n=5
[1.0999431208979835, 1.0231721836232246, 0.7599196398482674,
0.6179271340547875, 1.61735966084932]
[1.4870532313630744, 0.033117325688593054, 0.23641077941854624,
0.17089954401185403, 0.062190229549720595]
[0.4972709252204324, 0.037431184675976774, 0.9491365972011903,
0.15766067799900646, 1.2195715156246898]
[1.2400162817536216, 0.1445476361917044, 5.379479459444616,
0.318237695566795, 0.42575782550672575]
[0.44189005438095397, 0.05047793994970985, 1.5964832838301517,
0.012592118419619332, 0.12387791263130578]

n=10
[1.6937133419053623, 0.007467431059613799, 1.1449982381443404, 1.2005527380539958,
0.03218896859764006, 0.15959701523876638, 0.36626936395276394, 0.5864571179397815,
0.35722408238000597, 0.2848703511429897]
[0.45870925795512446, 0.6322497620353889, 0.656663340949564, 0.5649698162567983,
2.749602197752879, 2.9663187768291603, 0.1919132588291146, 0.3971906607794735,
1.6180916064404363, 0.9069358465901778]
[2.839076635580519, 0.5496371206544617, 0.245077931049124, 1.6210540560099278,
0.5916769322254678, 2.0852042848173626, 0.23049060222403164, 0.5590253617299235,
2.3975833507441355, 1.0883281058257048]
[0.6466203034187537, 2.323258877168198, 0.7668944874902923, 0.3629591683828057,
1.4137436905247334, 0.2515237295013045, 0.29496782858080783, 0.2029496376522597,
0.3005683632463297, 0.7844055847597325]
[1.8490038172266217, 0.3170940637645764, 0.8657925833205204, 1.9785355801385411,
0.10443322558164524, 0.16628666017878238, 0.9002243289300105, 3.460661057880038,
0.5191491723221491, 0.16051527758322168]

Создаем функцию для вычисления выборочного среднего и дисперсии и применяем их к сгенерированным выборкам распределения.

Выборочное среднее:

n=5
Выборочное среднее - 1.0236643478547165
Выборочное среднее - 0.39793422200635764

```python
for i in range(15,20):
    sum_el = 0
    for item in var_list[i]:
        sum_el = sum_el+item
    raz_var = len(var_list[i])
    it_sr = sum_el/raz_var
    srvr_list.append(it_sr)
    print("Выборочное среднее - ", it_sr)
```

Рис. 66: Код

Выборочное среднее - 0.5722141801442591
Выборочное среднее - 1.5016077796926925
Выборочное среднее - 0.44506426184234804

n=10
Выборочное среднее - 0.583333864841526
Выборочное среднее - 1.114264452441812
Выборочное среднее - 1.2207154380860656
Выборочное среднее - 0.7347891670725218
Выборочное среднее - 1.032169576692611

n=100
Выборочное среднее - 1.0036409006207696
Выборочное среднее - 0.9125987903367597
Выборочное среднее - 1.0577573912983125
Выборочное среднее - 0.9222421767232302
Выборочное среднее - 0.8181101465079221

n=1000
Выборочное среднее - 1.0123789911482708
Выборочное среднее - 0.986841483203838
Выборочное среднее - 0.9257189485569894
Выборочное среднее - 1.0666065008112484
Выборочное среднее - 1.0247516902652096

n=10000
Выборочное среднее - 0.9973756573027538
Выборочное среднее - 1.0008651681994791
Выборочное среднее - 1.0006799620361455
Выборочное среднее - 0.9979827554494645
Выборочное среднее - 1.0039114657998032

Исправлено:

Как мы знаем, выборочное среднее является оценкой для математического ожидания случайной величины.С помощью формул выведенных в первом домашнем задании посчитаем математическое ожидание для λ=1:

$$M\xi = \frac{1}{\lambda}$$

Так как $\lambda = 1$ тогда:

$$M\xi = \frac{1}{1} = 1$$

Из полученных нами результатов, выборочное среднее стремится к истинному значению математического ожидания при увеличении объема выборки. И это обусловленно свойством состоятельности нашей оценки, которое в свою очередь обучловлено законом больших чисел. Покажем это при помощи закона больших чисел Хинчина:

$$\frac{\xi_1+\xi_2+...+\xi n-1+\xi n}{n} \xrightarrow{P} M\xi$$

Если существует $M|\xi_1|$ тогда:

$$\overline{X_n} \xrightarrow{P} M\xi, n \to \infty$$

Получается, что по закону больших чисел Хинчина оценка является состоятельной. Выборочное среднее обладает свойством несмещенности.Тогда покажем, что математическое ожидание от оценки будет равно истинному математическому ожиданию:

$$M\overline{X} = M(\frac{1}{n}\sum_{i=1}^{n} M(X_i)) = \frac{1}{n} \cdot n \cdot M\xi = M\xi$$

Мы показали, что эта оценка является несмещенной.

Выборочная дисперсия вычислдяется с помощью кода представленного ниже:

```
for i in range(20,25):
    sum_el = 0
    sum_el_dic = 0
    for item in var_list[i]:
        sum_el = sum_el+item
    for item in var_list[i]:
        sum_el_dic = sum_el_dic+item*item
    raz_var = len(var_list[i])
    it_sr = sum_el/raz_var
    it_sr_dic = sum_el_dic/raz_var-it_sr*it_sr
    print("Выборочная дисперсия - ", it_sr_dic)

```

Рис. 67: Код

n=5

Выборочная дисперсия - 0.11849535515558274
Выборочная дисперсия - 0.30192603607385826
Выборочная дисперсия - 0.20492119562437267
Выборочная дисперсия - 3.9011497978842025
Выборочная дисперсия - 0.3543334061938572

n=10

Выборочная дисперсия - 0.29316372076856007
Выборочная дисперсия - 0.8939402410046107

Выборочная дисперсия - 0.8145532508092472
Выборочная дисперсия - 0.40321283862274626
Выборочная дисперсия - 1.0650240619446356

n=100
Выборочная дисперсия - 0.7667482072721106
Выборочная дисперсия - 0.8706876520101826
Выборочная дисперсия - 1.0787992073691104
Выборочная дисперсия - 0.7173581077031459
Выборочная дисперсия - 0.7758940584735567

n=1000
Выборочная дисперсия - 1.0008941353650054
Выборочная дисперсия - 0.9290919423560127
Выборочная дисперсия - 0.8353360117857244
Выборочная дисперсия - 1.2901945963977102
Выборочная дисперсия - 1.0064003922894733

n=10000
Выборочная дисперсия - 0.9892977386306037
Выборочная дисперсия - 1.0057685247547912
Выборочная дисперсия - 1.0043502699720284
Выборочная дисперсия - 0.9987937557921877
Выборочная дисперсия - 1.0060029252580207

Исправлено:

произведения сравнения значений параметров и их оценок, посчитаем численно математическое ожидание и дисперсию по формулам, выведенным в первом домашнем задании:

Как видно из полученных значений, при больших объмах выборки оценки выборочная дисперсия стремится к истинной дисперсии. Для произведения сравнения значений параметров и их оценок, посчитаем численно дисперсию по формуле, выведенной в первом домашнем задании для λ=1:

$$D\xi = \frac{1}{\lambda^2}$$

Так как $\lambda = 1$ тогда:

$$D\xi = \frac{1}{1^2} = 1$$

По полученным результатом выборочной дисперсии, мы можем видеть, что их значения стремятся к истинному значению $D\xi$ при росте объема выборки. Это обусловленно свойством состоятельности нашей оценки.

Воспользуемся доказанным свойством состоятельности выборочного среднего, как оценки и законом больших чисел.Выразим выборочную дисперсию.

$$\hat{\mu_2} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}^2 = \frac{1}{n}\sum_{i=1}^{n}X_i^2 - 2\bar{X}\frac{1}{n}\sum_{i=1}^{n}X_i + \frac{1}{n}\sum_{i=1}^{n}\bar{X}^2 = \bar{X^2} - \bar{X}^2$$

$$\overline{X^2} \xrightarrow{P} M\xi^2$$

$$\overline{X}^2 \xrightarrow{P} (M\xi)^2, n \to \infty$$

Выводы по закону больших чисел. Тогда:

$$\overline{X^2} - \overline{X}^2 \xrightarrow{P} M\xi^2 - (M\xi)^2 = D\xi, n \to \infty$$

Из этого следует, что оценка является состоятельной.

Проведем преобразования:

$$M\hat{\mu_2} = M(\overline{X^2} - \overline{X}^2) = M\overline{X^2} - M\overline{X}^2$$
$$M\xi^2 - M\overline{X}^2 = M\xi^2 - ((M\overline{X})^2 + D\overline{X})$$
$$M\xi^2 - (M\xi)^2 - D(\tfrac{1}{n}\sum_{i=1}^{n} X_i) = \mu_2 - \tfrac{1}{n^2}n\mu^2 = \tfrac{n-1}{n} \cdot \mu_2$$

Таким образом, мы показали что выборочная дисперсия не является несмещенной оценкой. Но путем преобразований $M\bar{\mu_2} = \frac{n-1}{n} \cdot \mu_2 \to M(\frac{n}{n-1} \cdot \hat{\mu_2}) = \mu_2$.Такая оценка будет несмещенной оценкой. Так как $\frac{n}{n-1} \to 1$ при $n \to \infty$, тогда данная оценка будет состоятельной.

3.2.2. Нахождение параметров распределений событий

Для показательного распределения необходимо проверить условие регулярности, проверить пренадлежит ли оно экспоненциальному семейству, предложить оценку X для оцениваемого параметра θ и выполним следующее:

- Проверить, является предложенная оценка несмещенной
- Проверить, является ли предложенная оценка X состоятельной для оцениваемого параметра θ
- Проверить, является ли предложенная оценка оптимальной, эффективной. Если нет, то построить оптимлаьную оценку для параметра θ

Два условия принято назывань условиями регулярности:ю.

- Существует такой носитель семейства распределений F_θ, что при каждом $x \in C$ функция $\sqrt{f_\theta(y)}$ непрерывно дифференцируема по θ всюду в области Θ
- Информация Фишера $I(\theta) = E(\frac{\partial}{\partial\theta} ln f_\theta(X_1))^2$ существует, положительна и непрерывна по θ во всех точках $\theta \in \Theta$

Статистика $T = T(X)$ называется несмещенной в среднем (или просто несмещенная) оценкой для для заданной параметрической функции $\tau(\theta)$, если она удовлетворяет условию

$$M_\theta T(X) = \tau(\theta), \forall\theta \in \Theta$$

котороые называется уравнением несмещенности.

$$M\hat{\theta} = M\bar{X} = MX_1 = \tfrac{1}{\lambda}$$

Найдем оценку параметра θ^* методом максимального правдоподобия.

$$L(\theta, x) = \prod_{n=1}^{n} \theta e^{-\alpha x_i} = \theta^n e^{-\theta \sum_{i=1}^{n} x_i}$$

Воспользуемся условием максимума $\frac{dL(\theta,x)}{d\theta} = 0$. И получим следующее уравнение:

$$n\theta^{n-1}e^{-\theta\sum_{i=1}^{n}x_i} - \theta^n\sum_{i=1}^{n}x_ie^{-\theta\sum_{i=1}^{n}x_i} = 0$$

Отсюда следует $\theta^* = \frac{1}{\frac{1}{n}\sum_{i=1}^{n}x_i}$. Откуда $\tau(\theta) = \hat{\theta} = \bar{X}$

Найдем теперь оценку параметра θ^* методом моментов.

$\mu_1(\theta) = M[x] = \int_0^\infty x\theta e^{-\theta x}dx = \frac{1}{\theta}$ Приравниваем к $m_1 = \frac{1}{n}\sum_{i=1}^{n}x_i$. Отсюда получаем $\theta^* = \frac{1}{m_1} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n}x_i}$. Таким образом, оценки параметра θ полученные методом максимального правдоподобия и методом моментов совпали.

Статистика $T_n = T_n(X)$ называется состоятельной оценкой для неизвестного параметра $\tau(\theta)$ случайной величины ξ, если при $n->\infty$

$$T_n(x) \xrightarrow[n->\infty]{P} \tau(\theta)$$

$$\bar{X} \xrightarrow[n->\infty]{P} MX_1$$

$$\hat{\theta} = \bar{X}->MX_1$$

$$\hat{X}->MX_1 = \frac{1}{\lambda}$$

Эффективные оценки существуют только в моделях экспоненциального семейства распределелний. А именно, если

$$L(x,\theta) = h(x)\cdot exp[A(\theta)\cdot U(x) + B(\theta)]$$

то существует эффективная оценка $T(x) = U(x)$ параметрической функции

$$\tau(\theta) = -\frac{B'(\theta)}{A'(\theta)}$$

$$L(x,\theta) = 1\cdot[-\theta n\cdot\bar{X} + nln\theta]$$

Откуда следует $U(x) = \bar{X}$

$$\tau(\theta) = -\frac{\frac{n}{\theta}}{-n} = \frac{1}{\theta}$$

Мы показали, что оценка является эффективной, а значит оптимальная и единственная.

Рассмотрим значения оценки по нашим выборкам, которые мы моделировали выше. Применим код написанные на Python:

Output:

Объем выборки = 5 , параметр = 4.4

```python
for sample in samples_list:
    sum=0
    for element in sample:
        sum += element
    print("Объем выборки = ", len(sample),',','параметр = ', sum/len(sample)-1)
```

Рис. 68: Код

Объем выборки = 5 , параметр = 1.7999999999999998
Объем выборки = 5 , параметр = 4.2
Объем выборки = 5 , параметр = 2.4
Объем выборки = 5 , параметр = 4.8
Объем выборки = 10 , параметр = 1.9
Объем выборки = 10 , параметр = 3.0999999999999996
Объем выборки = 10 , параметр = 3.4000000000000004
Объем выборки = 10 , параметр = 3.2
Объем выборки = 10 , параметр = 2.6
Объем выборки = 100 , параметр = 3.2300000000000004
Объем выборки = 100 , параметр = 3.3499999999999996
Объем выборки = 100 , параметр = 3.4400000000000004
Объем выборки = 100 , параметр = 3.58
Объем выборки = 100 , параметр = 3.2
Объем выборки = 1000 , параметр = 3.4989999999999997
Объем выборки = 1000 , параметр = 3.5570000000000004
Объем выборки = 1000 , параметр = 3.458
Объем выборки = 1000 , параметр = 3.5780000000000003
Объем выборки = 1000 , параметр = 3.4749999999999996
Объем выборки = 10000 , параметр = 3.492
Объем выборки = 10000 , параметр = 3.5092999999999996
Объем выборки = 10000 , параметр = 3.4665999999999997
Объем выборки = 10000 , параметр = 3.5395000000000003
Объем выборки = 10000 , параметр = 3.5204000000000004

Таким образом можно наблюдать свойста состоятельности оценки.
