Шаплавский Леонид Павлович СКБ182

Оглавление

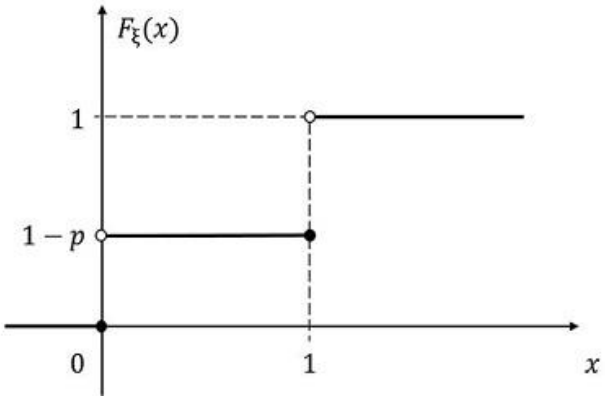
Шаплавский Леонид Павлович СКБ182 1
Распределение Бернулли 2
Функция распределение 2
Определение понятий 2
Математическое ожидание 2
Дисперсия 3
Мода распределения 3
Медиана распределения 3
Производящая функция для дискретных неотрицательных случайных величин 3
Характеристическая функция 3
Моделирование(поменял) 5
Подключаем необходимые модули: 6
Моделирование: 6
Создаём выборку: 6
Генерация пяти выборок: 7
Создание вариационного ряда: 8
Квантиль: 8
Построение эмпирической функции распределения: 9
Построение гистограммы и полигона частот 13
Распределение Эрланга 16
Функция распределения 16
Математическое ожидание 16
Дисперсия: 16
Мода: 17
Описание моделирование 18
Функции: 18
Моделируем: 19
Генерация пяти выборок: 22
Создание вариационного ряда: 22
Построение эмпирической функции распределения: 23
Квантиль: 26
Построение гистограммы и полигона частот 28

Распределение Бернулли

$$P(x) = p^x \cdot (1-P)^{1-x}, x \in \{0,1\}, 0 < p < 1$$

Функция распределение

$$F_\xi(x) = P(\xi < x) = \begin{cases} 0, & x < 0; \\ 1-p, & 0 < x < 1; \\ 1, & x \geq 1. \end{cases}$$



На графике видно, что во всех точках x, кроме точек $x = 0$ и $x = 1$, функция $F(x)$, непрерывна справа и слева, однако в $x = 0$, $x = 1$ функция непрерывна только справа, поскольку $F(0) = F(0+0)$ и $F(1) = F(1+0)$, но $F(0) \neq F(0-0), F(1) \neq F(1-0)$, от куда следует, что функция распределения $F(x)$ непрерывна справа, но не непрерывна слева.

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

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

Определение понятий

Математическое ожидание

Математическим ожиданием случайной величины ξ (средним значением) называется интеграл Лебега от неё мере P:

$$\mathrm{M}\xi = \mathrm{E}\xi = \int_\Omega \xi P(d\omega)$$

Если расписать интеграл Лебега по определению получим:

$$\mathrm{M}\xi = \mathrm{E}\xi = \sum_{i=1} x_i p_i$$

Математическое ожидание для распределения Бернулли:

$$\mathrm{M}\xi = \mathrm{E}\xi = \sum_{i=1} x_i p_i = 1 \cdot p + 0 \cdot q = p$$

Дисперсия

Центральный момент второго порядка называется дисперсией и обозначается :

$$\mathrm{D}\xi = \sigma^2(\xi) = M(\xi - M\xi)^2 = \mathrm{M}\xi^2 - (M\xi)^2$$

$$\mathrm{M}\xi^2 = 1 \cdot p + 0 \cdot q = p$$

$$\mathrm{D}\xi = p - p^2 = p(1-p) = pq$$

Мода распределения

Модой абсолютно непрерывного распределения называют любую точку локального максимума плотности распределения. Для дискретных распределений модой считают любое значение $\mathrm{a_i}$, вероятность которого больше, чем вероятности соседних значений (соседнего, если таковое одно).

При $\mathrm{p} > \mathrm{q}, \xi = 1$ – мода

При $\mathrm{p} < \mathrm{q}, \xi = 0$ - мода

При $\mathrm{p} = \mathrm{q}, \xi = 1, \xi = 0$ – две моды

Медиана распределения

Медианой распределения случайной величины ξ является значение, для которого вероятность меньшего значений, равна вероятности больших значений.

Медианы нет, поскольку случайные величины имеет только два значения.

Производящая функция для дискретных неотрицательных случайных величин

Производящей функцией $f_\xi(\mathrm{s})$ распределения случайной величины ξ, принимающей целые неотрицательные значения, называется степенной ряд:

$$f_\xi(\mathrm{s}) = \mathrm{M} \cdot \mathrm{s}^\xi = \sum_{\mathrm{r}=0}^{\infty} \mathrm{P}(\xi = k) \cdot \mathrm{s}^\mathrm{r}$$

$$f_\xi(\mathrm{s}) = qs + ps' = q + ps$$

Характеристическая функция

Характеристической функцией дискретной случайной величиной ξ , принимающей действительные значения, называется функция:

$$f(t) = \sum_{k=1} e^{itx_k} P(\xi = x_k)$$

$$f(t) = pe^{i\cdot 1\cdot t} + qe^{i\cdot 0\cdot t} = pe^{it} + q$$

По свойству производящей функции:

$$f_\xi^{(k)}(1) = M\left(\xi^{[k]}\right), f_\xi\ \ (s) = q + ps$$

$f_\xi'(1) = p$ – *первый факториальный момент*

$f_\xi''(1) = 0$ – *второй факториальный момент*

$f_\xi'''(1) = 0$ – *третий факториальный момент*

Благодаря характеристической функции найдём первые три момента случайной величины:

$$f^{(k)}(0) = i^k M\left(\xi^{[k]}\right), \text{где } k \in \overline{1,n}$$

$$f(t) = q + pe^{it}$$

$$f'(t) = ipe^{it}$$

$$f''(t) = -pe^{it}$$

$$f'''(t) = -pie^{it}$$

$f_\xi'(0) = i^1 M_{\xi^1}, M_{\xi^1} = \frac{f'(0)}{i} = \frac{pi}{i} = p$ – *первый момент*

$f_\xi''(0) = i^2 M_{\xi^2}, M_{\xi^2} = \frac{f''(0)}{i^2} = -\frac{p}{-1} = p$ – *второй момент*

$f_\xi'''(0) = i^3 M_{\xi^3}, M_{\xi^3} = \frac{f'''(0)}{i^3} = -\frac{pi}{-i} = p$ – *третий момент*

Второй центральный момент случайной величины есть дисперсия:

$$D\xi = f''(1) + f'(1) - f'(1)^2 = 0 + p - p^2 = p(1-p) = pq$$

$$P\{\xi = k\} = \frac{f_\xi^{(k)}(0)}{k!}$$

$$P\{\xi = 0\} = \frac{q}{0!} = q$$

$P\{\xi = 3\}$ как и $P\{\xi \geq 3\} = 0$, поскольку случайная величина ξ может принимать только значения 0 и 1

1.4 Примеров событий, которые могут быть описаны выбранными случайными величинами

1) Пример интерпретации распределения Рассмотрим пример для распределения Бернулли.

Примером интерпретации для него является следующая ситуация. Двое одновременно стреляют по мишени. Вероятность попадания по мишени у первого стрелка равна 0,6, у второго - 0,7. Какова вероятность того, что в мишени будет только одна пробоина? Нам подойдут два исхода, когда первый попадет, а второй промажет и, когда первый промахнется, а второй попадет. По условию p1=0,6, p2=0,7, значит q1=1−p1=0,4, q2=1−p2=0,3. Получим: P=p1·q2+q1·p2=0,6·0,3+0,4·0,7=0,46. Каждый стрелок может попасть или промазать, всего два варианта у каждого. Поэтому случайная величина (попадание или промах), имеет Распределение Бернулли.

2) Соотношения, связывающие распределение Бернулли с другими
Биноминальное распределение

$$P(x) = \binom{n}{x} p^x q^{n-x}, x \in \{0,1,\dots,n\}, n \geq 1, 0 < p < 1, q = 1 - p$$

Если все события независимы, с одинаково распределёнными случайными величинами и все испытания Бернулли с вероятностью успеха p, тогда их сумма распределяется в соответствии с биномиальным распределением с параметрами n и p:

$$\sum_{k=1}^{n} X_k \sim B(n, p)$$

В распределение Бернулли n=1: B(1, p)

Геометрическое распределение

$$P(x) = pq^x, x \in \mathbb{N} \cup \{0\}, 0 < p < 1, q = 1 - p$$

В геометрическом распределении моделей числа независимых и идентичных испытания Бернулли необходимо, чтобы получить один успех.

Моделирование(поменял)

Алгоритм: Исправлено

1. Установить $x \longleftarrow 0, x \longleftarrow 0$
2. Цикл пока k = n
    - Генерируется $U(0,1), k \longleftarrow k + 1$
    - Если $U \leq p,\ x \longleftarrow x + 1$
3. Возвращаем x

Доказательство:

Алгоритм построен на основе испытания Бернулли, что и является распределением Бернулли

Код:

До Эрланга всё исправлял

Подключаем необходимые модули:

```
import random
import matplotlib.pyplot as plt
import numpy as np
```

Моделирование:

```
def Bernoulli_discription(p):
    u = random.random()
    if u <= p:
        r = 1
    else:
        r = 0
    return r

def PMF(p,n,t):
    q = 1-p
    y=[]
    for i in range(n):
        if t[i]==0:
            y.append(q)
        else:
            y.append(p)
    return y
```

Создаём выборку:

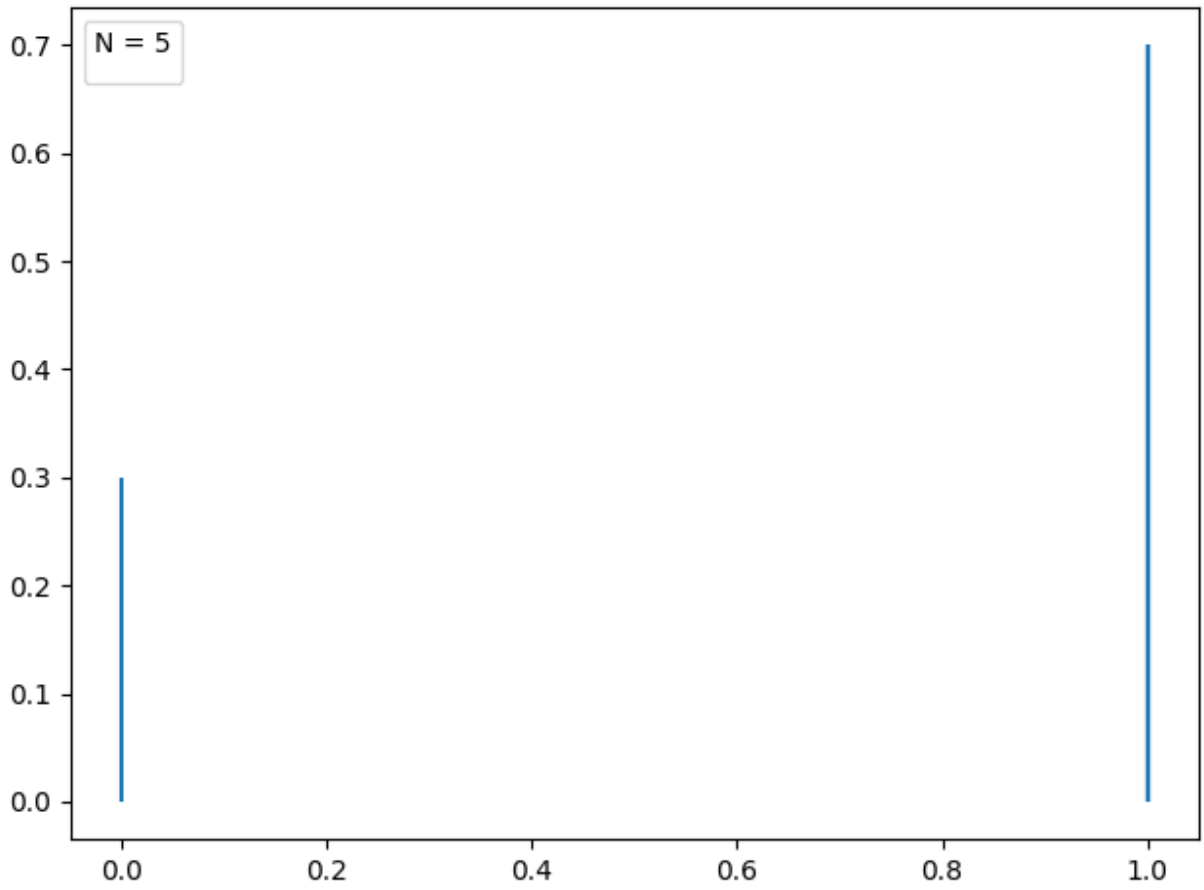
```
n = 5
p = 0.7

x=[]
for i in range(n):
    x.append(Bernoulli_discription(p))
x.sort()
print(x)
```

Вывод:

[0, 1, 1, 1, 1]

```
y = PMF(p,n,x)
print(y)

plt.vlines(x,0,y)
plt.legend(title='N = {}'.format(n),loc="upper left")
plt.show()
```

[0.30000000000000004, 0.7, 0.7, 0.7, 0.7]



Генерация пяти выборок:

```
def sampaling (N,p,t):
    sampl=[]
    for i in range(t):
        sampl.append(0)
    for i in range(len(sampl)):
        sampl[i] = []
    # print(sampl)
    for i in range(len(sampl)):
        for j in range(N):
            sampl[i].append(0)
    # print(sampl)
    for i in range(t):
        for j in range(N):
            sampl[i][j] = Bernoulli_discription(p)
    return sampl


sampl = sampaling(n,p,5)
```

Вывод:

[[1, 0, 0, 1, 1], [1, 0, 0, 1, 1], [1, 1, 1, 1, 0], [0, 1, 1, 0, 1], [0, 1, 1, 1, 1]]

Создание вариационного ряда:

Вариационный ряд – последовательность $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$, полученная в результате расположения в порядке неубывания исходной последовательности независимых одинаково распределенных случайных величин X1, . . . ,Xn.

```
for i in range(5):
    sampl[i].sort()
print(sampl)
```

[[0, 0, 1, 1, 1], [0, 0, 1, 1, 1], [0, 1, 1, 1, 1], [0, 0, 1, 1, 1], [0, 1, 1, 1, 1]]

Квантиль:

α-квантиль случайной величины σ с функцией распределения $F(x) = P\{\sigma < x\}$ - это любое число xα, удовлетворяющее двум условиям:

1. $F(x_\alpha) \leq \alpha$

2. $F(x + \alpha + 0) \geq \alpha$

```
print('_____________')
kvantilX=[]
for v in [0.1,0.5,0.7]:
    kvan = ceil(v*(n-1))
    if (kvan+1) < (v*n):
        kvantilX.append(sampl[0][kvan+1])
    elif (kvan + 1) == (v * n):
        kvantilX.append((sampl[0][kvan]+sampl[0][kvan+1])/2)
    elif (kvan+1) > (v*n):
        kvantilX.append(sampl[0][kvan])

gg=[0.1,0.5,0.7]
for i in range(3):
    print(kvantilX[i]," {}".format(gg[i]))
```

N=5

0 0.1

1 0.5

1 0.7

N=10

0 0.1

1 0.5

1 0.7

N=100

0 0.1

1 0.5

1 0.7

N=1000

0 0.1

1 0.5

1 0.7

N=10000

0 0.1

1 0.5

1 0.7

Теоретический:

0 0.1

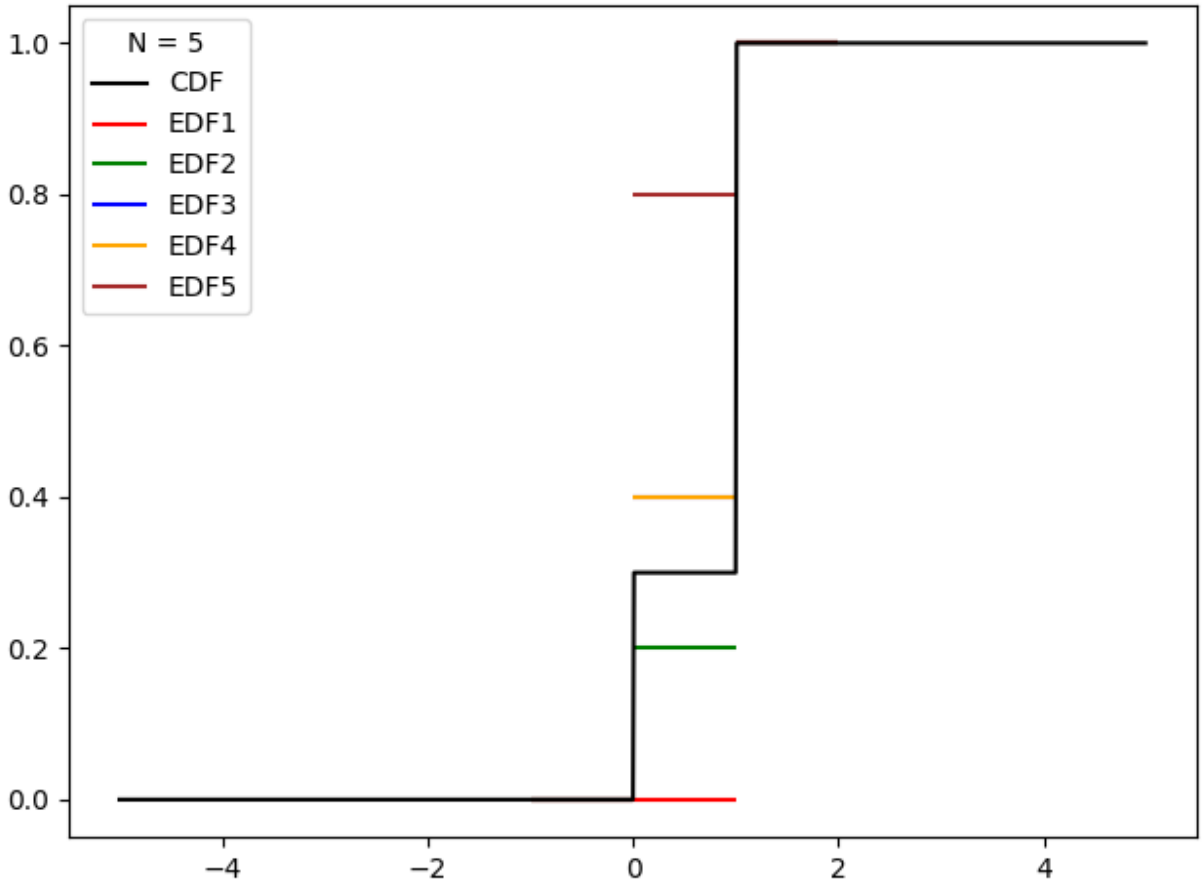
1 0.5

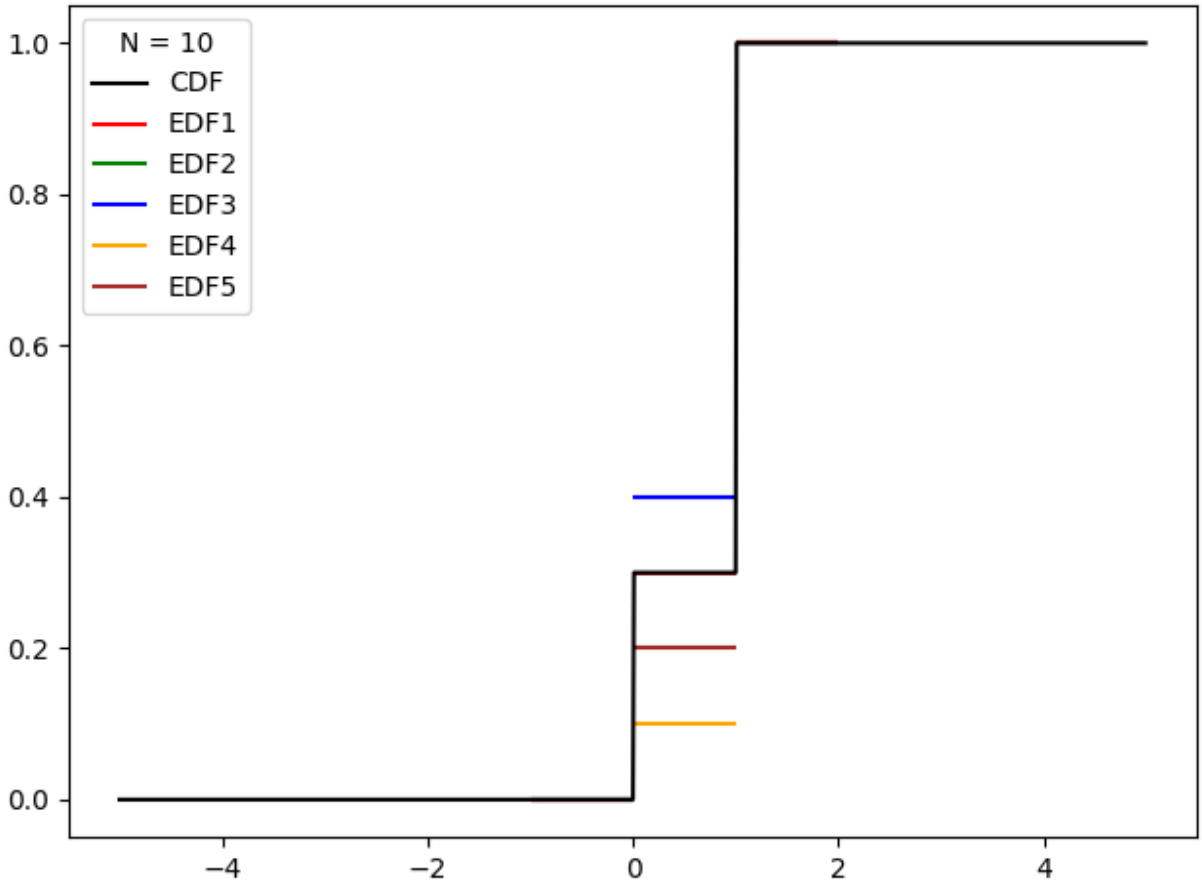
1 0.7

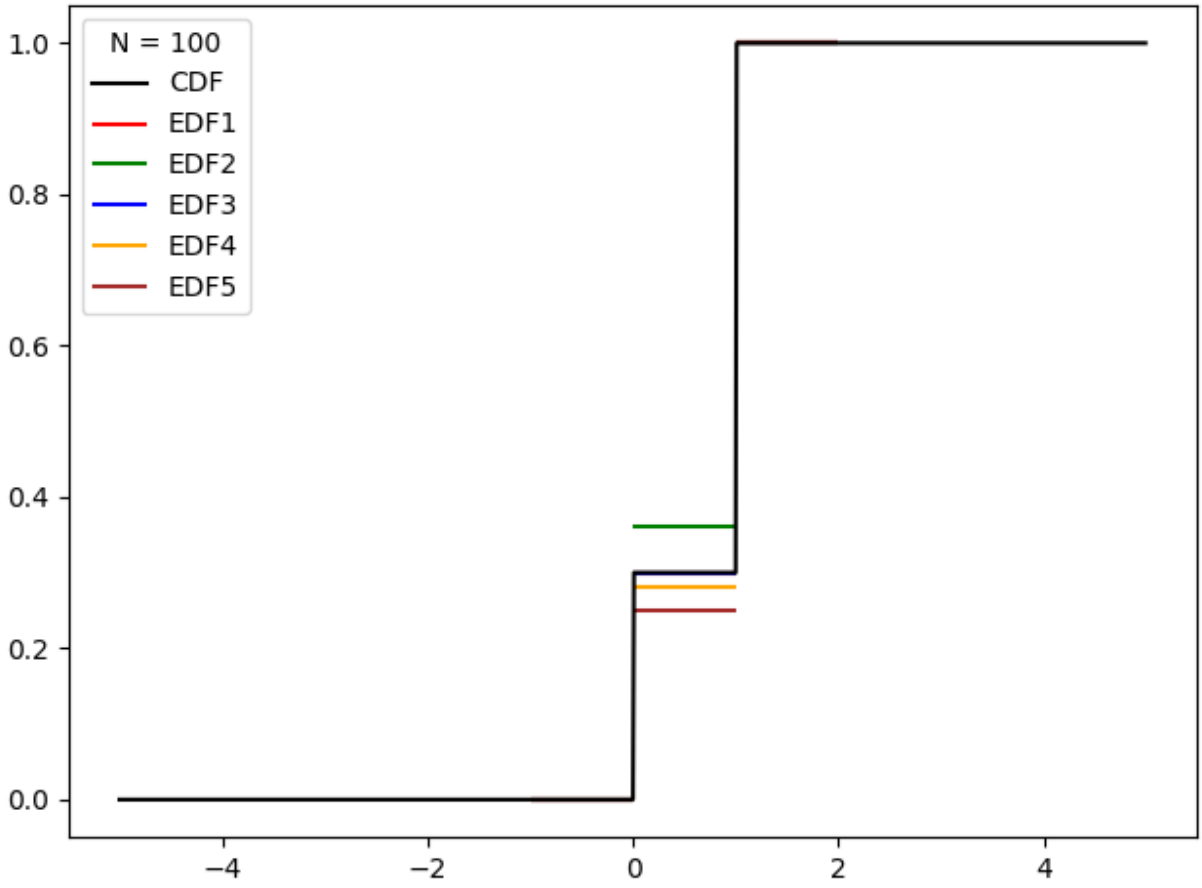
Построение эмпирической функции распределения:

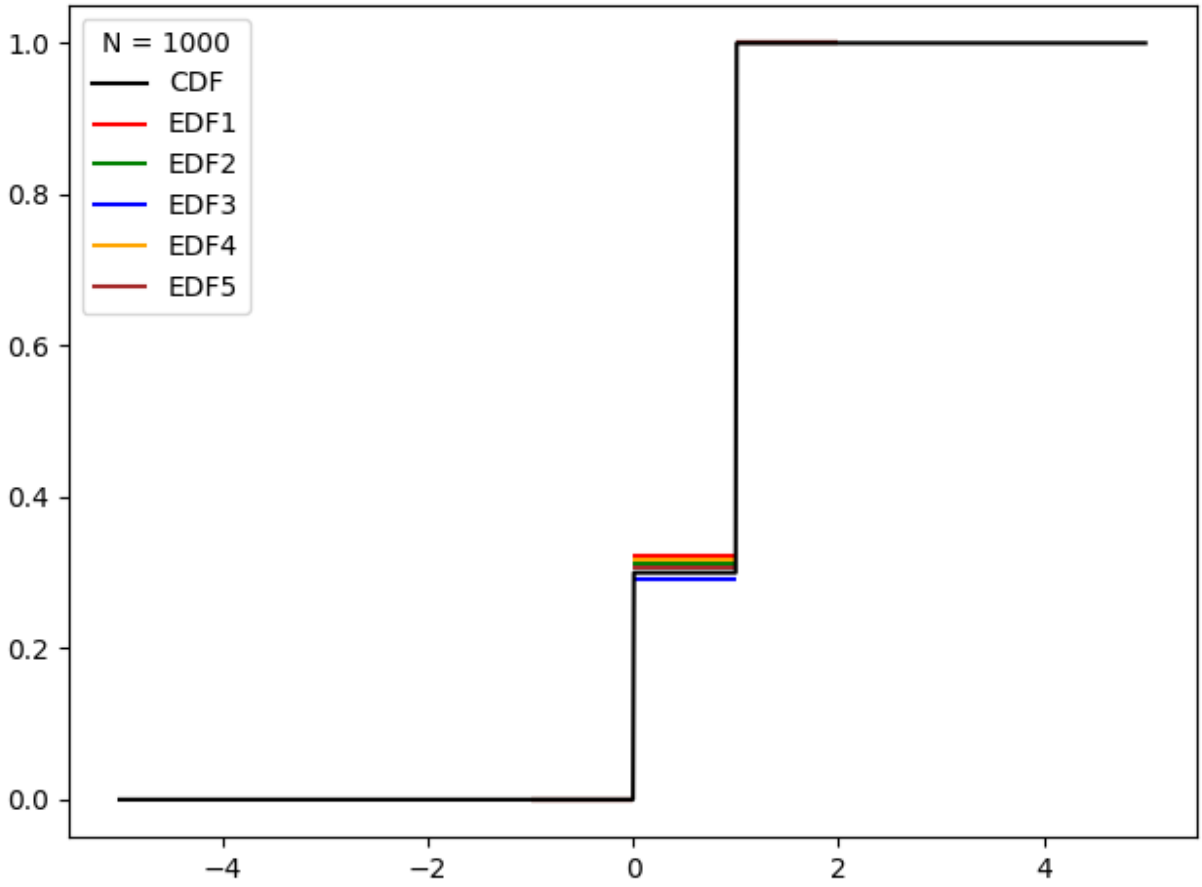
```
def CDF(p,c):
    if c < 0:
        g = 0
    elif 0 <= c and c < 1:
        g = 1-p
    elif c >= 1:
        g = 1
    return g


o=[]
a=[-1,0,1]
b=[0,1,2]
ss1=[]
cc=["red","green","blue","orange","brown"]



for i in range(5):
    sa = sampl[i]
    lenn = len(sa)
    t=sa.index(1)/lenn
    tt=[0,t,1]
    plt.hlines(tt,a,b,label='EDF{}'.format(i+1),colors=cc[i])
```

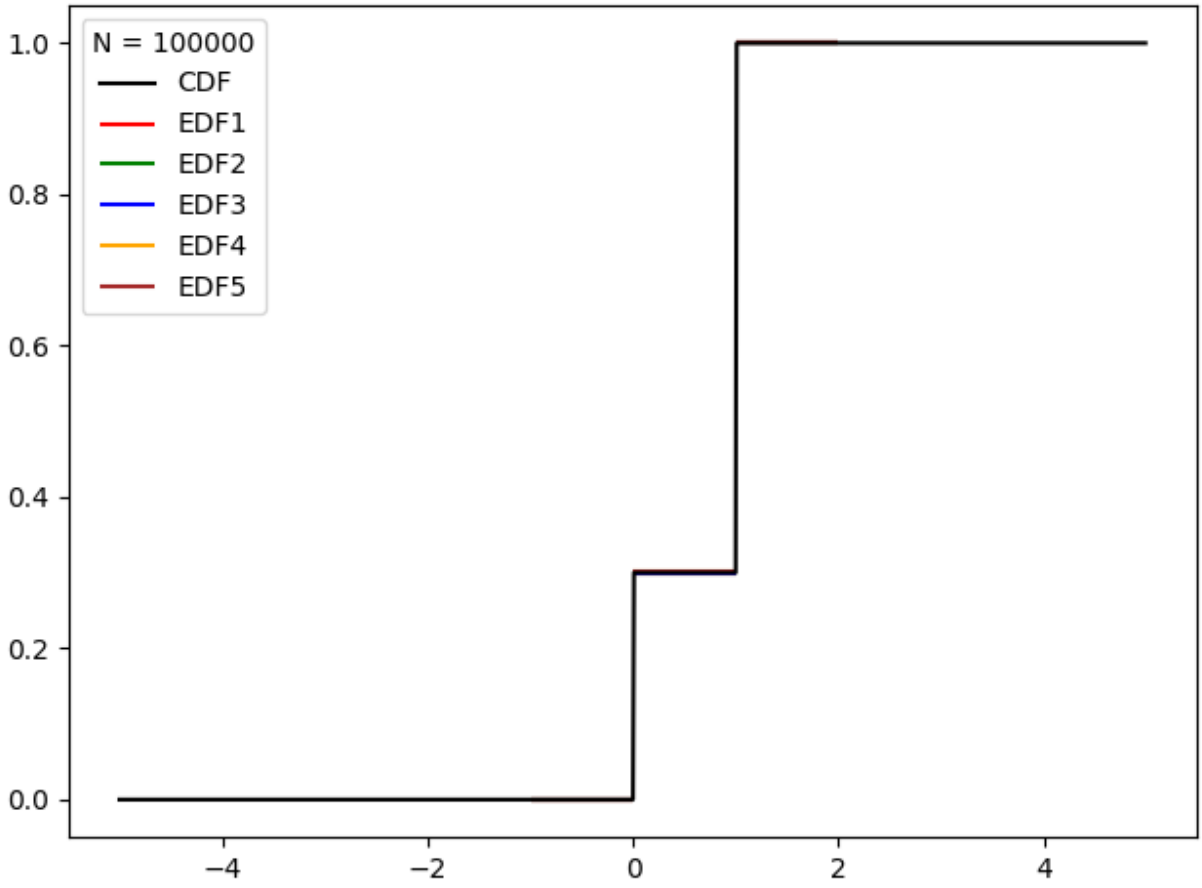
```
xCDF=np.arange(-5,5,0.01)
yCDF=[]

for i in range(len(xCDF)):
    yCDF.append(CDF(p,xCDF[i]))
```




N = 10
CDF
EDF1
EDF2
EDF3
EDF4
EDF5
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

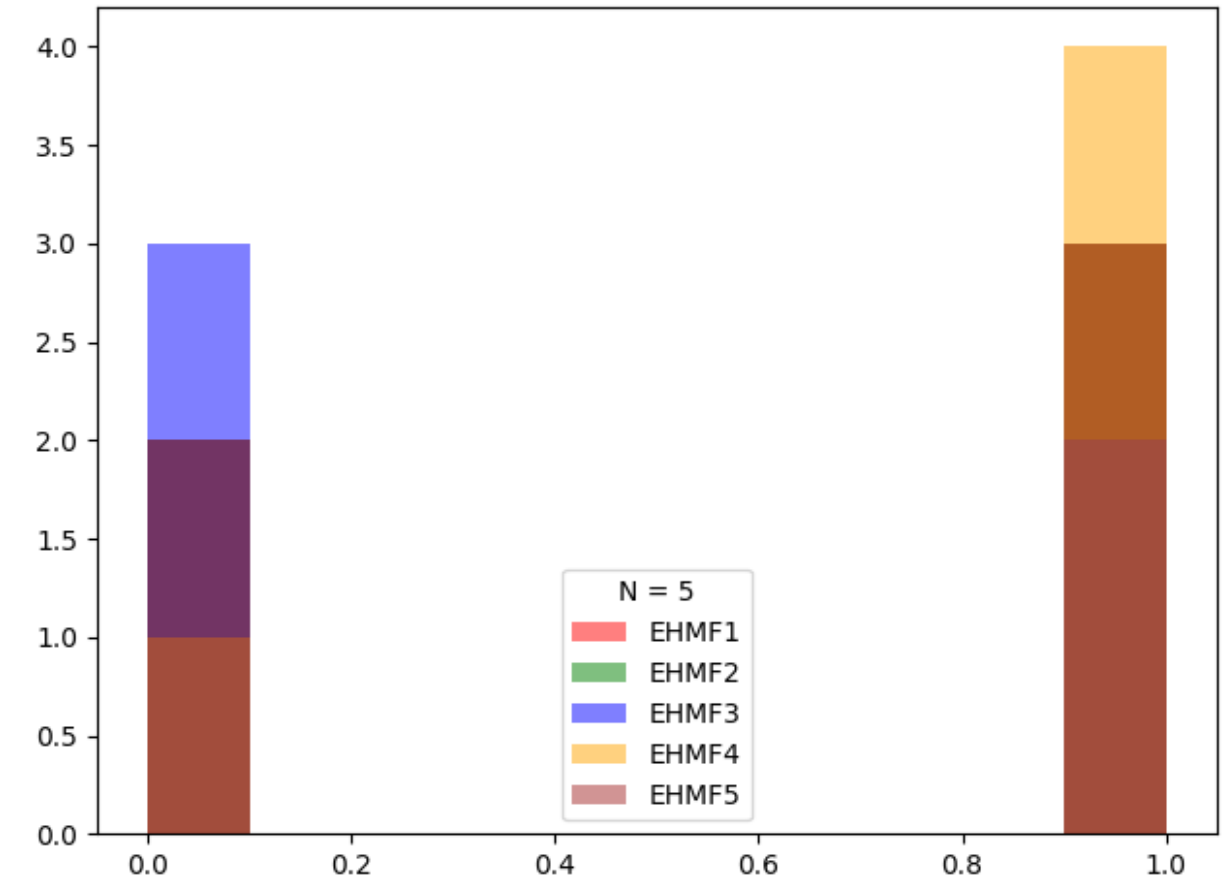

N = 100
CDF
EDF1
EDF2
EDF3
EDF4
EDF5
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

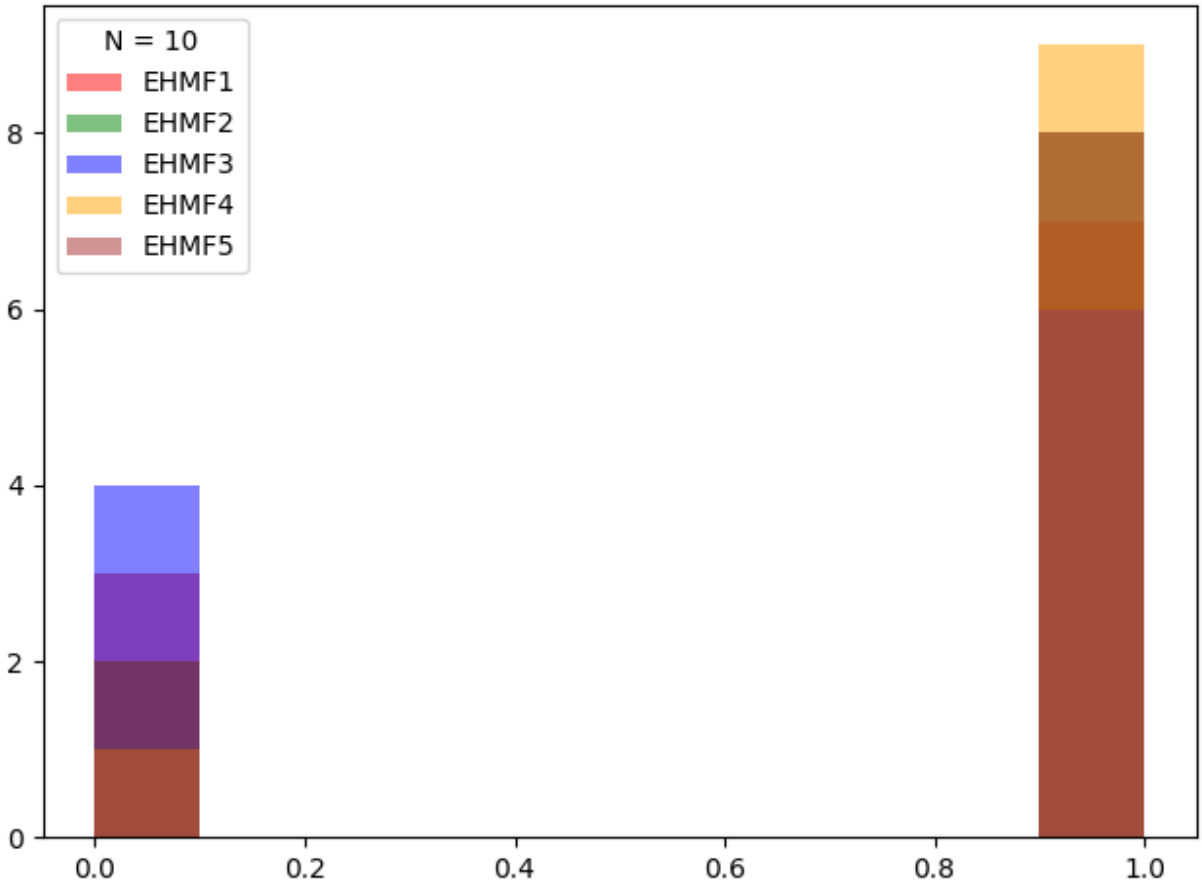

N = 1000
CDF
EDF1
EDF2
EDF3
EDF4
EDF5
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

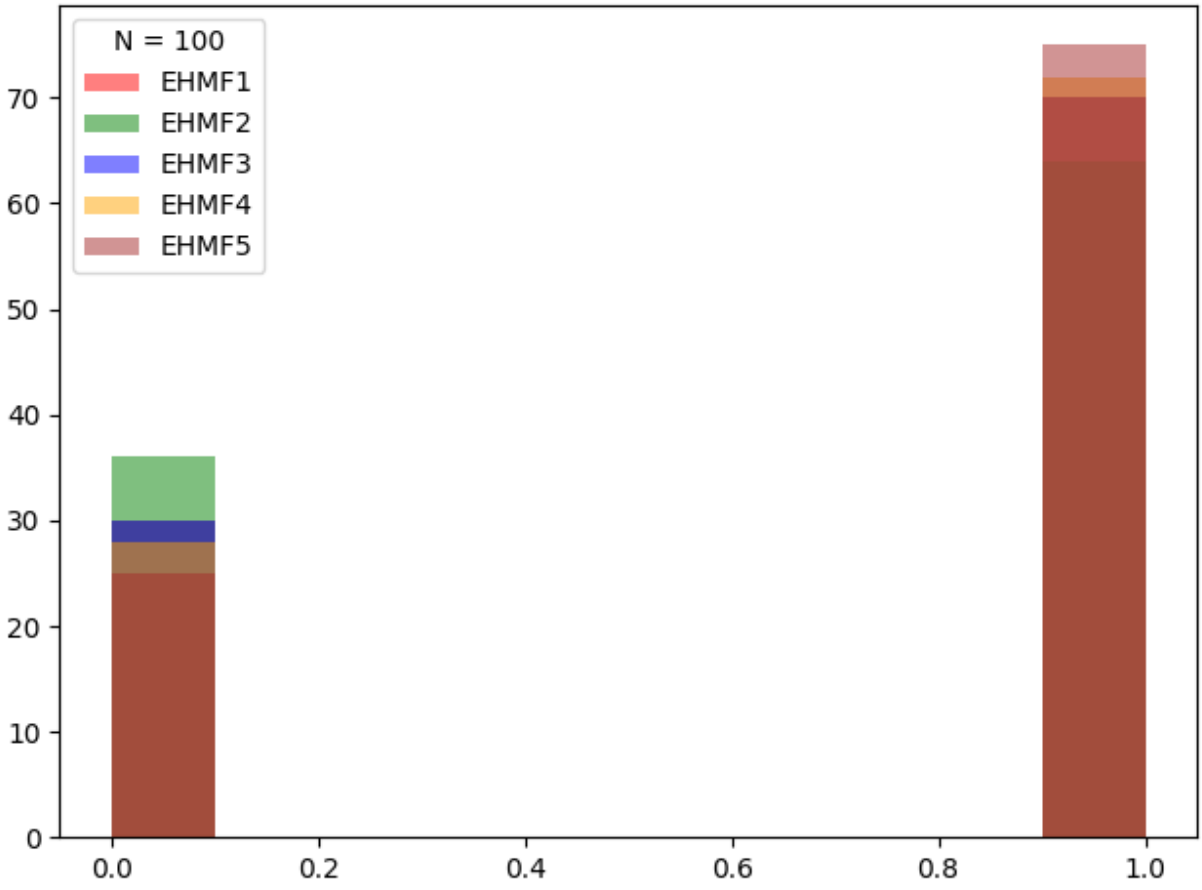

N = 100000
CDF
EDF1
EDF2
EDF3
EDF4
EDF5
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

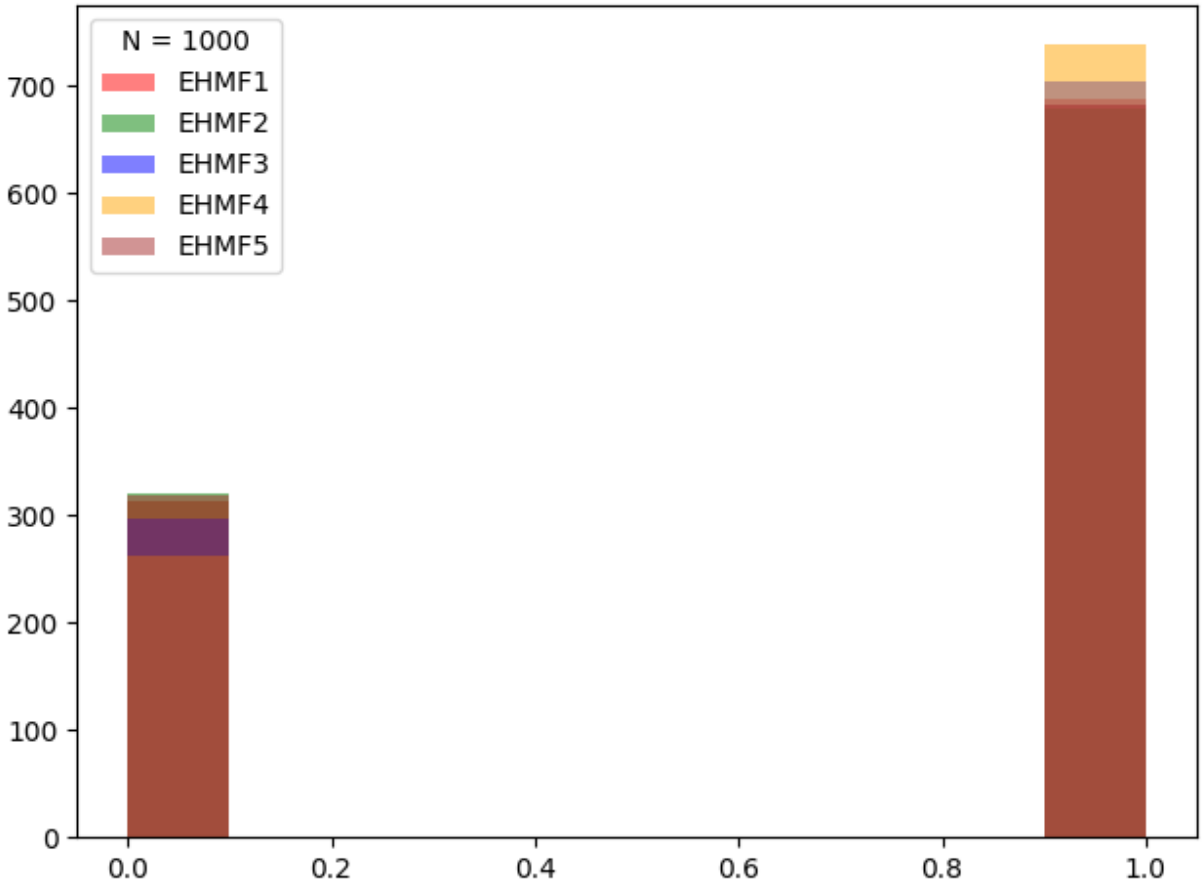
Построение гистограммы и полигона частот

```
for i in range(5):
    plt.hist(sampl[i],color=cc[i],label="EHMF{:d}".format(i+1),alpha=0.5)



plt.legend(title='N = {}'.format(n))
plt.show()
```




N = 10
EHMF1
EHMF2
EHMF3
EHMF4
EHMF5
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0


N = 100
EHMF1
EHMF2
EHMF3
EHMF4
EHMF5
0
10
20
30
40
50
60
70
0.0
0.2
0.4
0.6
0.8
1.0


N = 1000
EHMF1
EHMF2
EHMF3
EHMF4
EHMF5
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0

Распределение Эрланга

$$f(x) = \frac{(\lambda^x)}{(m-1)!} x^{m-1} e^{-\lambda x}, x, \lambda \in \mathrm{R}, x, \lambda > 0, m \in N$$

Функция распределения

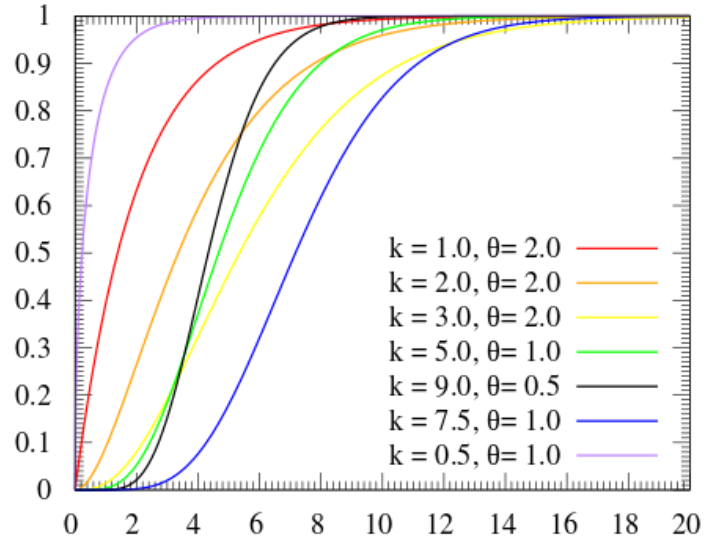
$$F(x) = \int_0^x f(x)dx =$$

$$= \int_0^x \frac{(\lambda^x)}{(m-1)!} x^{m-1} e^{-\lambda x} dx = \int_0^x \lambda \frac{(\lambda x)^{m-1}}{(m-1)!} e^{-\lambda x} dx =$$

$$= \int_0^x \frac{(\lambda x)^{m-1}}{(m-1)!} e^{-\lambda x} d\lambda x = \frac{\gamma(m, \lambda x)}{(m-1)!} =$$

$$= \sum_{n=0}^{\infty} \frac{(\lambda x)^{m|n} e^{-\lambda x}}{m(m+1)\ldots(m+n)(m-1)!} = 1 - \sum_{n=0}^{\infty} \frac{e^{-\lambda x}(\lambda x)^n}{n!}$$

$$F(x) = \begin{cases} 0, & x < 0 \\ 1 - \sum_{n=0}^{\infty} \frac{e^{-\lambda x}(\lambda x)^n}{n!}, & x \geq 0 \end{cases}$$



Математическое ожидание

$$E[x] = \int_{-\infty}^{+\infty} x f(x)dx = \int_0^{\infty} \frac{x\, \lambda^m\ x^{m-1} e^{-\lambda x}}{(m-1)!} dx = \int_0^{\infty} \frac{\lambda^m\ x^m e^{-\lambda x}}{(m-1)!} dx$$

$$= \frac{m}{\lambda} \int_0^{\infty} \frac{\lambda^{m+1}\ x^m e^{-\lambda x}}{m!} dx = \frac{m\, \Gamma(m+1)}{\lambda\, m!} = \frac{m}{\lambda}$$

Дисперсия:

$$D[x] = E[x^2] - (E[x])^2$$

$$E[x^2] = \int_{-\infty}^{\infty} x^2 f(x)dx = \int_0^{\infty} \frac{x^2\ \lambda^m\ x^{m-1}e^{-\lambda x}}{(m-1)!}dx =$$
$$= \frac{m(m+1)}{\lambda^2}\int_0^{\infty} \frac{\lambda^{m+2}\ x^{m+1}e^{-\lambda x}}{(m+1)!}dx = \frac{m(m+1)}{\lambda^2}$$

$$D[x] = \frac{m(m+1)}{\lambda^2} - \frac{m^2}{\lambda^2} = \frac{m}{\lambda^2}$$

Мода:

$$f'(x) = \frac{\lambda^m\ (m-1)x^{m-2}e^{-\lambda x}}{(m-1)!} - \frac{\lambda^m\ x^{m-1}e^{-\lambda x}}{(m-1)!}$$

$$\frac{\lambda^m\ (m-1)x^{m-2}e^{-\lambda x}}{(m-1)!} - \frac{\lambda^m\ x^{m-1}e^{-\lambda x}}{(m-1)!} = 0$$

$$\frac{\lambda^m\ (m-1)x^{m-2}e^{-\lambda x}}{(m-1)!} = \frac{\lambda^m\ x^{m-1}e^{-\lambda x}}{(m-1)!}$$

$$\frac{m-1}{\lambda} = x$$

Производящая функция для дискретных неотрецательных случайных велличин

Нету, поскольку у нас непрерывное распределение

Связь с другими распределениями

Наше распределение является частным решение гаммы, принимая в k только натуральные значения.

Описание моделирование

Как мы выяснили, выше Эрланг является частным решением гаммы, соответственно можно описать наше распределение через функцию плотности Гамма распределения:

$$f(x) = \frac{(e^{-x}x^{a-1})}{\Gamma(a)}, x \geq 0; a \geq 0$$

Если x и y есть независимые случайные величины гамма распределения с параметрами a и b соответственно, то x+y так же подчиняется гамма распределению с параметром a+b.

При a = 1 функция принимает вид $f(x) = e^{-x}$Стандартные гамма-переменные являются экспоненциальными и могут генерироваться как отрицательные $-\log(u)$ от равномерно распределенной переменной. Если a некоторое целочисленное, положительное число n, то утверждение 3.1 можно применить n-1 раз, получая $x \in (-\log(u1), -\log(u2), \ldots, -\log(un))$ в качестве допустимого преобразования.

Алгоритм:

1. Инициализируем $i \leftarrow p \leftarrow 1$
2. Сгенерировать u и сформировать p
3. Если i=n, вернуть $-\log(p) \rightarrow x$, иначе $i + 1 \rightarrow i$ и вернуться к шагу 2

Дальше до конца всё исправлял

```
import random
from math import exp
from math import e
from math import log
from math import factorial
from math import ceil
import matplotlib.pyplot as plt
from scipy.special import gamma as gm
```

Функции:

```
def erlang(a,l):
    i=1
    p=1
    while(i!=a):
        i=i+1
        u=random.random()
        p=p*u
    return -log(p)/l
```

```
def PDF(x,l,k):
    j = (l**(k) * x**(k-1) * e**(-x*l))/(factorial(k-1))
    return j
```

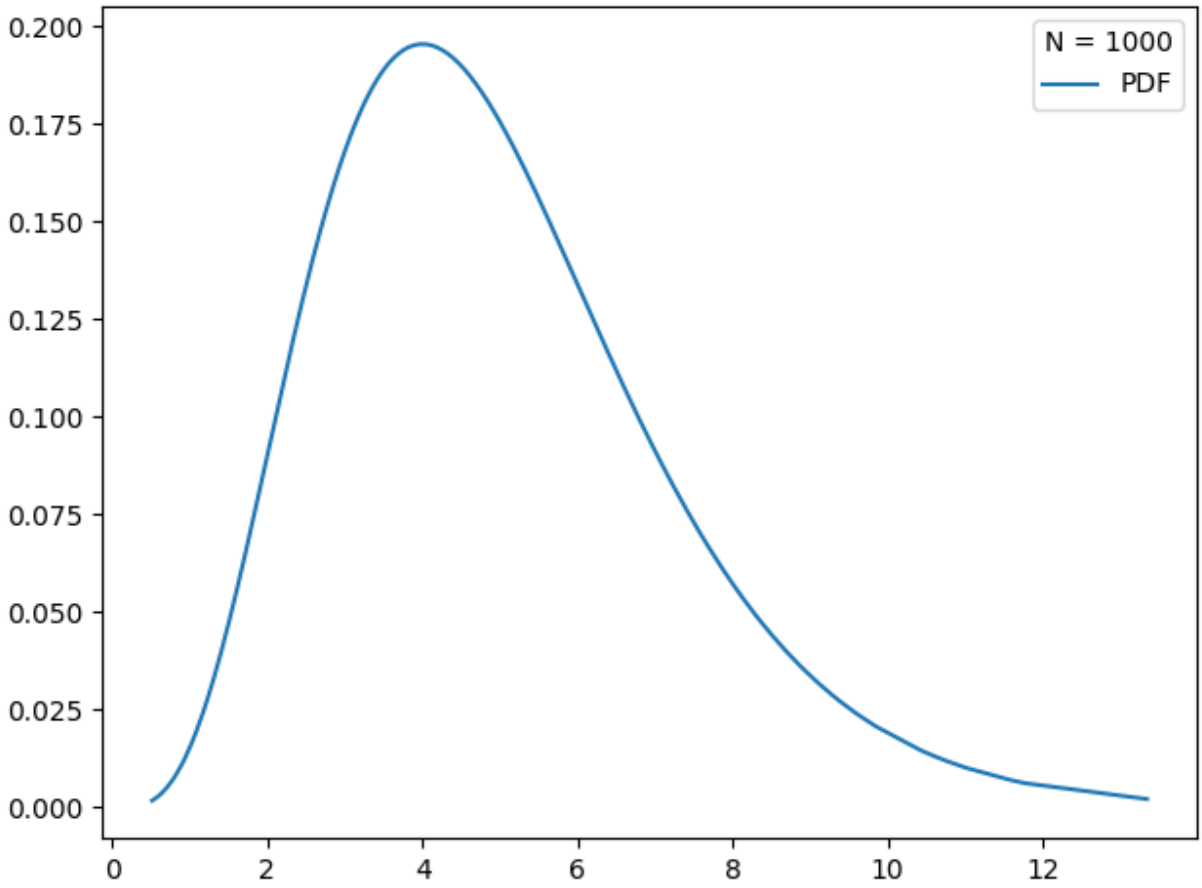
```

```

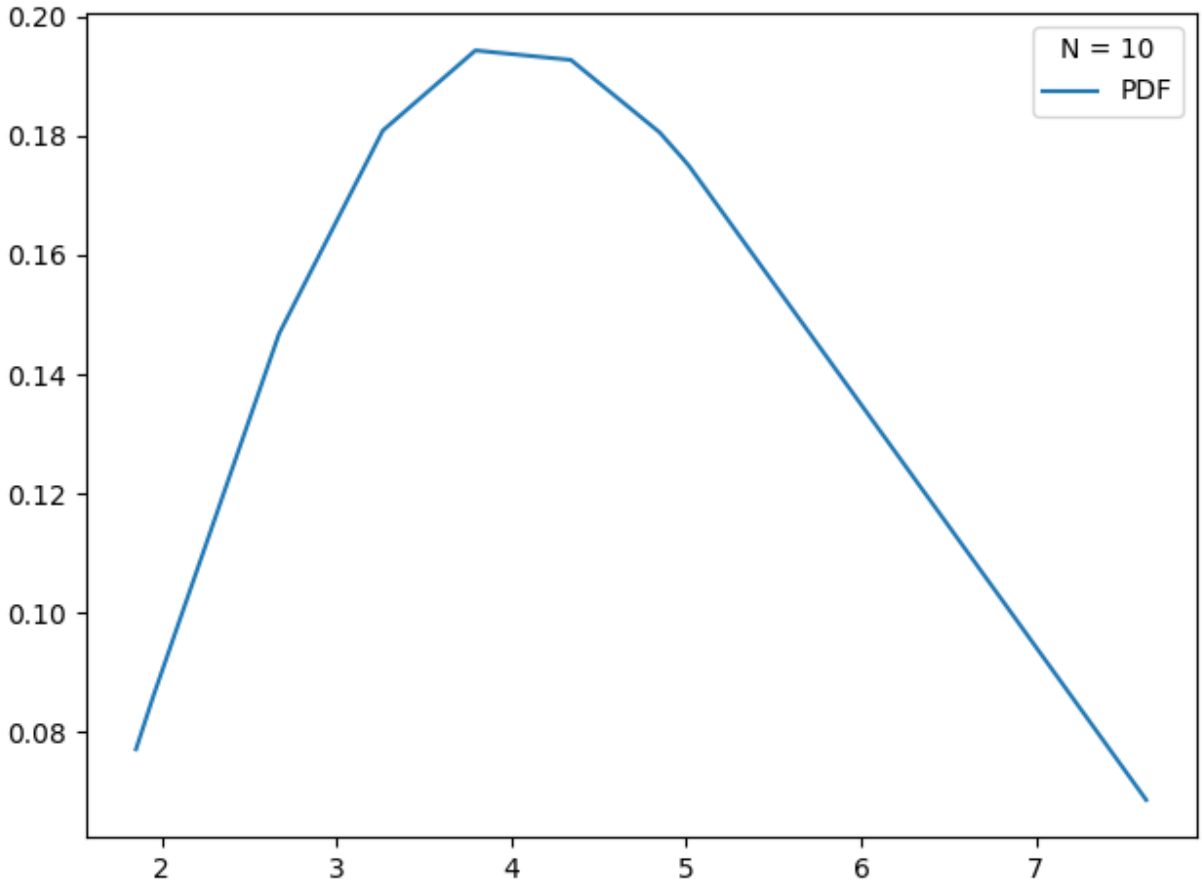
```
def CDF(i,l,k):
    sum1 = 0
    for j in range(0,k - 1):
        sum1 = sum1 + (((exp(-(l * i))) * ((l * i) ** j)) / (factorial(j)))
    return 1 - sum1
def sampaling (N,a,l,t):
    sampl=[]
    for i in range(t):
        sampl.append(0)
    for i in range(len(sampl)):
        sampl[i] = []
    for i in range(len(sampl)):
        for j in range(N):
            sampl[i].append(0)
    for i in range(t):
        for j in range(N):
            sampl[i][j] = erlang(a,l)
    return sampl
def arrey_creation (N,t):
    sampl=[]
    for i in range(t):
        sampl.append(0)
    for i in range(len(sampl)):
        sampl[i] = []
    for i in range(len(sampl)):
        for j in range(N):
            sampl[i].append(0)
    return sampl

def EDF (sampl):
    n =len(sampl[0])
    y_EDF = arrey_creation(n,len(sampl))
    work_samples = sampl

    for i in range(len(sampl)):
        for j in range(n):
            if work_samples[i][j] <= work_samples[i][0]:
                y_EDF[i][j]=0
            elif work_samples[i][j] > work_samples[i][0] and
work_samples[i][j] < work_samples[i][n-1]:
                y_EDF[i][j] = (j/n)
            else:
                y_EDF[i][j]=1
    return y_EDF
```

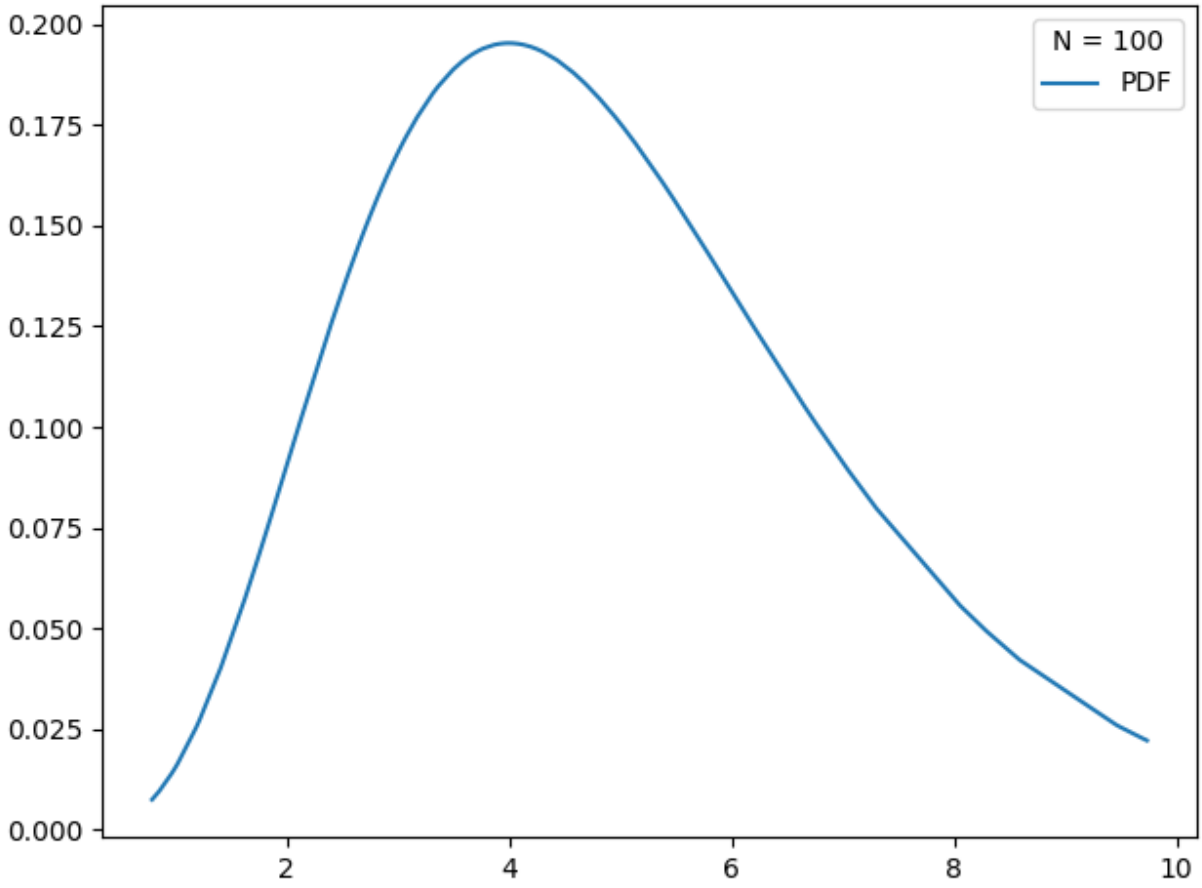
Моделируем:

```
k=5
l=1

x =[]
y=[]

for i in range(n):
    x.append(erlang(k,l))

x.sort()

for i in range(n):
    y.append(PDF(x[i],l,k))

print(x)
print(y)
```

```
plt.plot(x,y,label ='PDF')
plt.legend(title='N = {}'.format(n))
plt.show()
```




N = 10
PDF
0.20
0.18
0.16
0.14
0.12
0.10
0.08
2
3
4
5
6
7


N = 100
PDF
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
2
4
6
8
10

Генерация пяти выборок:

```
sampl = sampaling(n,k,1,5)
```

Вывод:

[[5.121471848400674, 4.27050061650272, 4.47506290677155, 5.045570904012902, 5.021750262322504],

[2.3566830697080414, 7.226178084586032, 2.444554901082047, 5.321604794795407, 5.098717752967133],

[6.824641770071372, 2.1334349228318095, 2.004325957275834, 4.131633654430441, 3.8046452456844686],

[2.3715464000883713, 6.927794500580919, 6.913325411437188, 2.3159459278622583, 3.538749698715397],

[4.143002197194301, 3.1374760629414107, 3.6895891565739554, 2.7252049659616886, 1.812506268735553]]

Создание вариационного ряда:

```
for i in range(len(sampl)):
    sampl[i].sort()
```

[[4.27050061650272, 4.47506290677155, 5.021750262322504, 5.045570904012902, 5.121471848400674],

[2.3566830697080414, 2.444554901082047, 5.098717752967133, 5.321604794795407, 7.226178084586032],

[2.004325957275834, 2.1334349228318095, 3.8046452456844686, 4.131633654430441, 6.824641770071372],

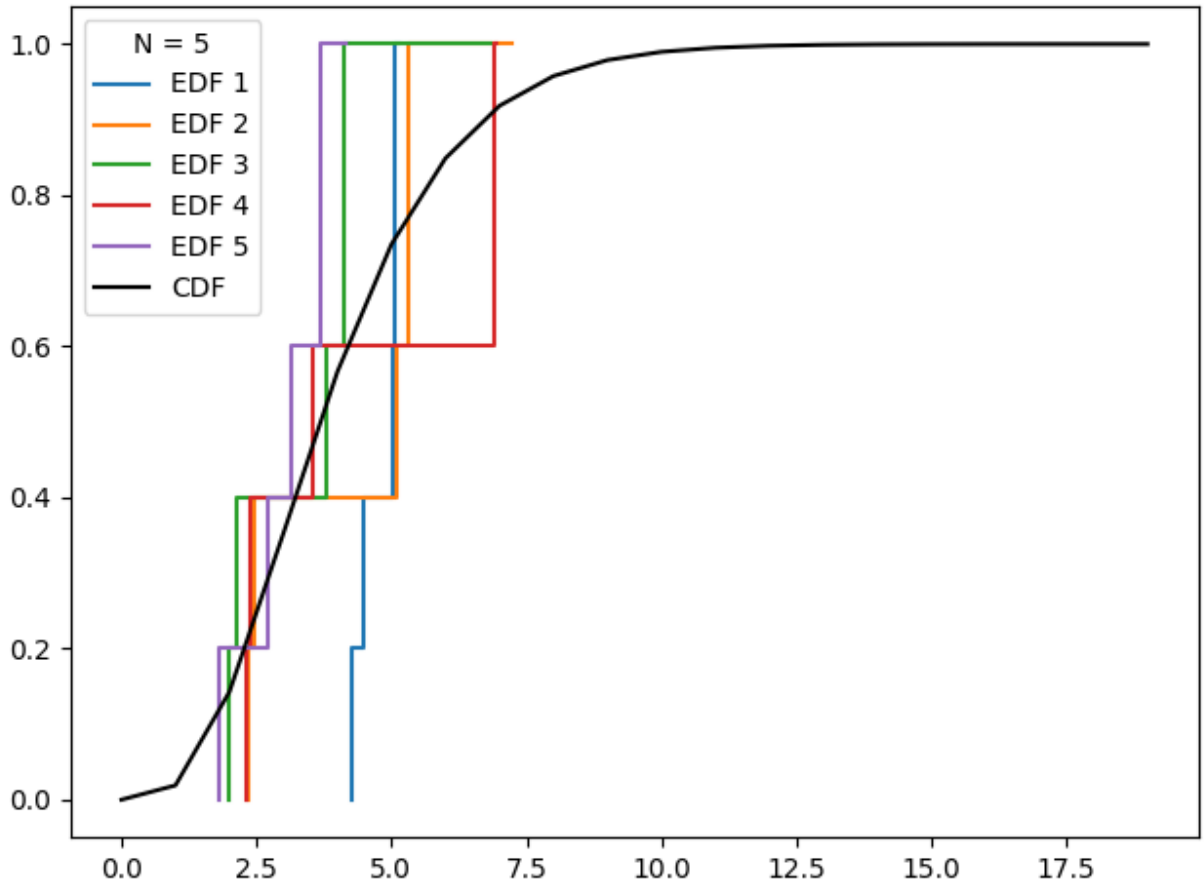
[2.3159459278622583, 2.3715464000883713, 3.538749698715397, 6.913325411437188, 6.927794500580919],

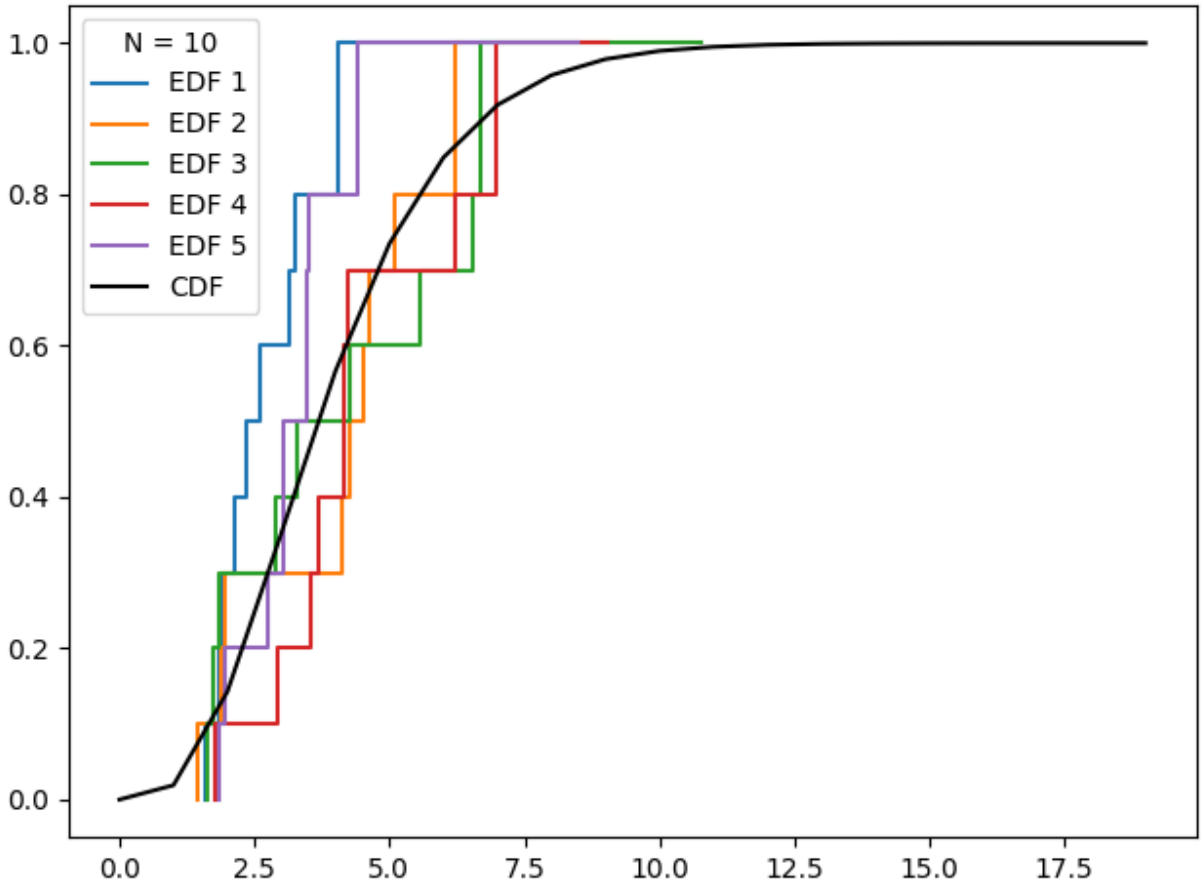
[1.812506268735553, 2.7252049659616886, 3.1374760629414107, 3.6895891565739554, 4.143002197194301]]

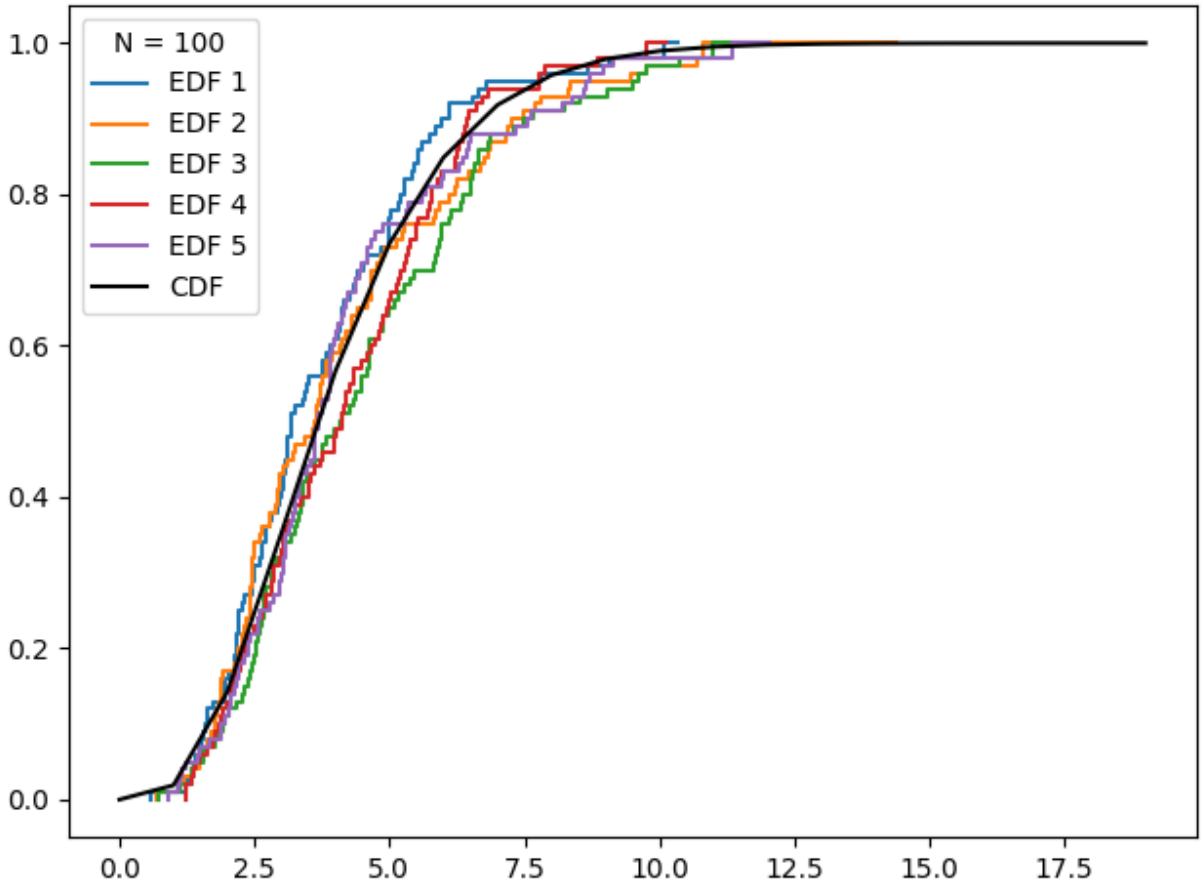
Построение эмпирической функции распределения:

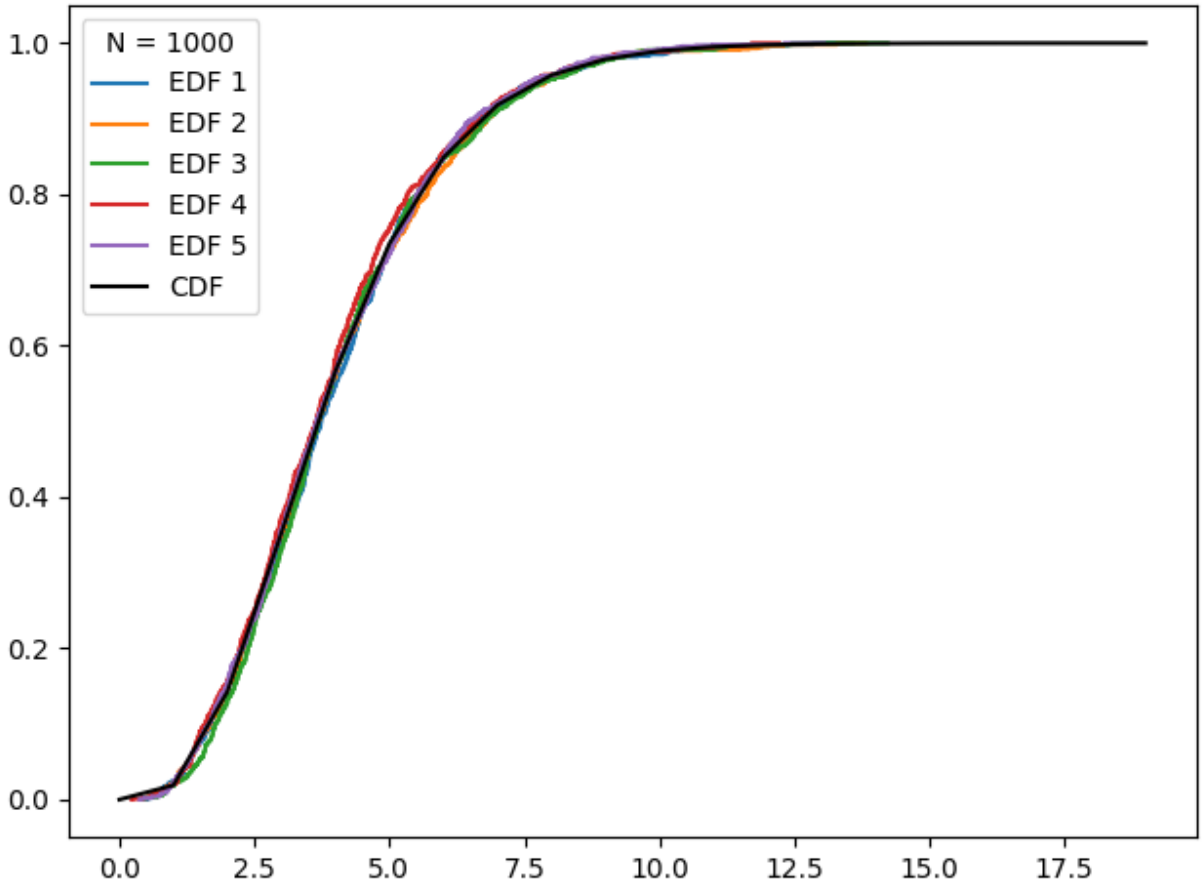
```
y_EDF = EDF(sampl)
print(y_EDF)
xCDF=[]
yCDF=[]
for i in range(20):
    xCDF.append(i)
xCDF.sort()


for i in range(len(xCDF)):
    yCDF.append(CDF(i,l,k))


for i in range(5):
    plt.step(sampl[i], y_EDF[i],label="EDF {}".format(i+1))

plt.plot(xCDF, yCDF,color='black',label ='CDF' )
plt.legend(loc="upper left",title='N = {}'.format(n))
plt.show()
```

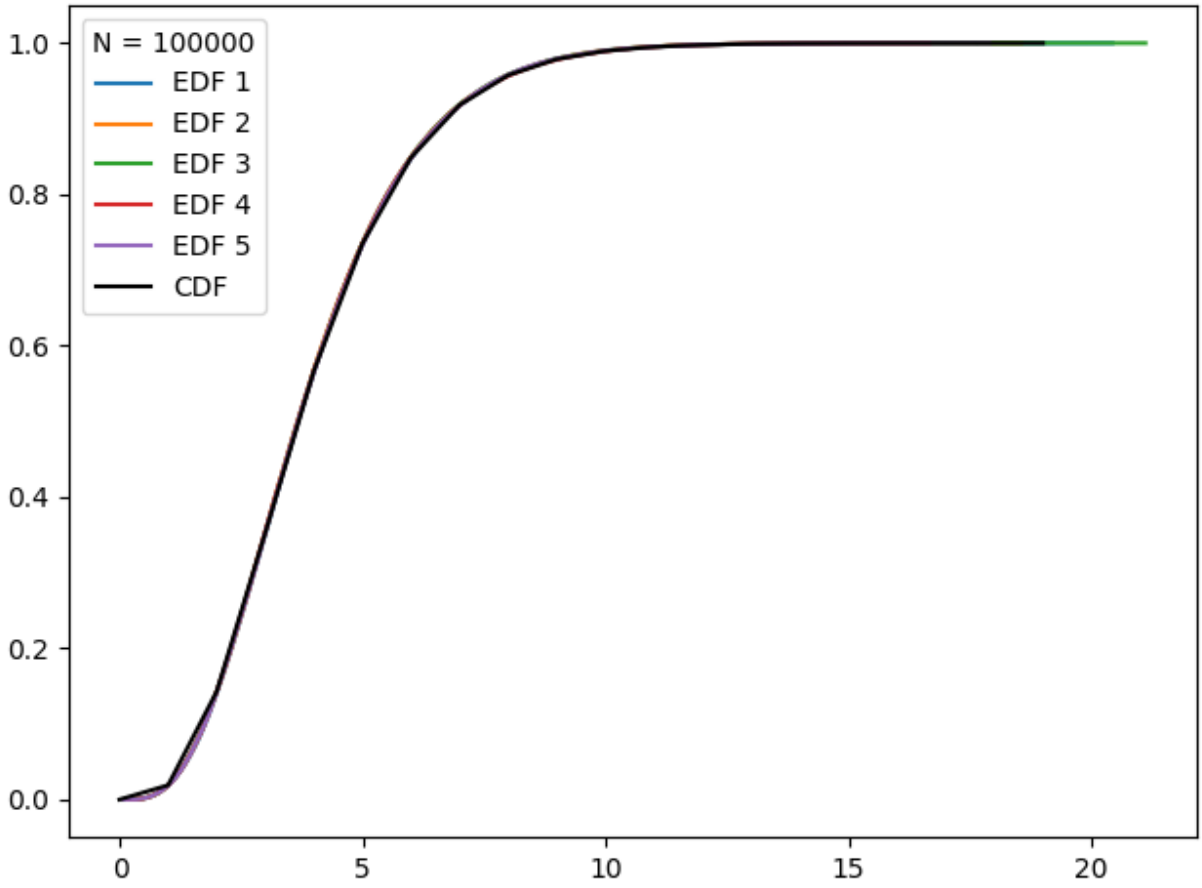
[[0, 0.2, 0.4, 0.6, 1], [0, 0.2, 0.4, 0.6, 1], [0, 0.2, 0.4, 0.6, 1], [0, 0.2, 0.4, 0.6, 1], [0, 0.2, 0.4, 0.6, 1]]




N = 10
EDF 1
EDF 2
EDF 3
EDF 4
EDF 5
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


N = 100
EDF 1
EDF 2
EDF 3
EDF 4
EDF 5
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


N = 1000
EDF 1
EDF 2
EDF 3
EDF 4
EDF 5
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


N = 100000
EDF 1
EDF 2
EDF 3
EDF 4
EDF 5
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20

Квантиль:

```
for v in [0.1,0.5,0.7]:
    kvan = ceil(v*(n-1))
    if (kvan+1) < (v*n):
        kvantilX.append(sampl[0][kvan+1])
    elif (kvan + 1) == (v * n):
        kvantilX.append((sampl[0][kvan]+sampl[0][kvan+1])/2)
    elif (kvan+1) > (v*n):
        kvantilX.append(sampl[0][kvan])

gg=[0.1,0.5,0.7]
for i in range(3):
    print(kvantilX[i]," {}".format(gg[i]))
```

N=5

4.334258762414626 0.1

5.01640816845459 0.5

5.717660126645539 0.7

N=10

1.6391696494537693 0.1

3.4979072430354368 0.5

4.231038777572746 0.7

N=100

1.59193029143821 0.1

3.6673189535159576 0.5

4.932209163378797 0.7

N=1000

1.7874450972808178 0.1

3.8325288128741115 0.5

5.086382595207342 0.7

N=10000

1.74840916637723 0.1

3.664861936358644 0.5

4.751254123159008 0.7

Проверка:

```
print(gamma.ppf(0.1,4))
print(gamma.ppf(0.5,4))
print(gamma.ppf(0.7,4))
```

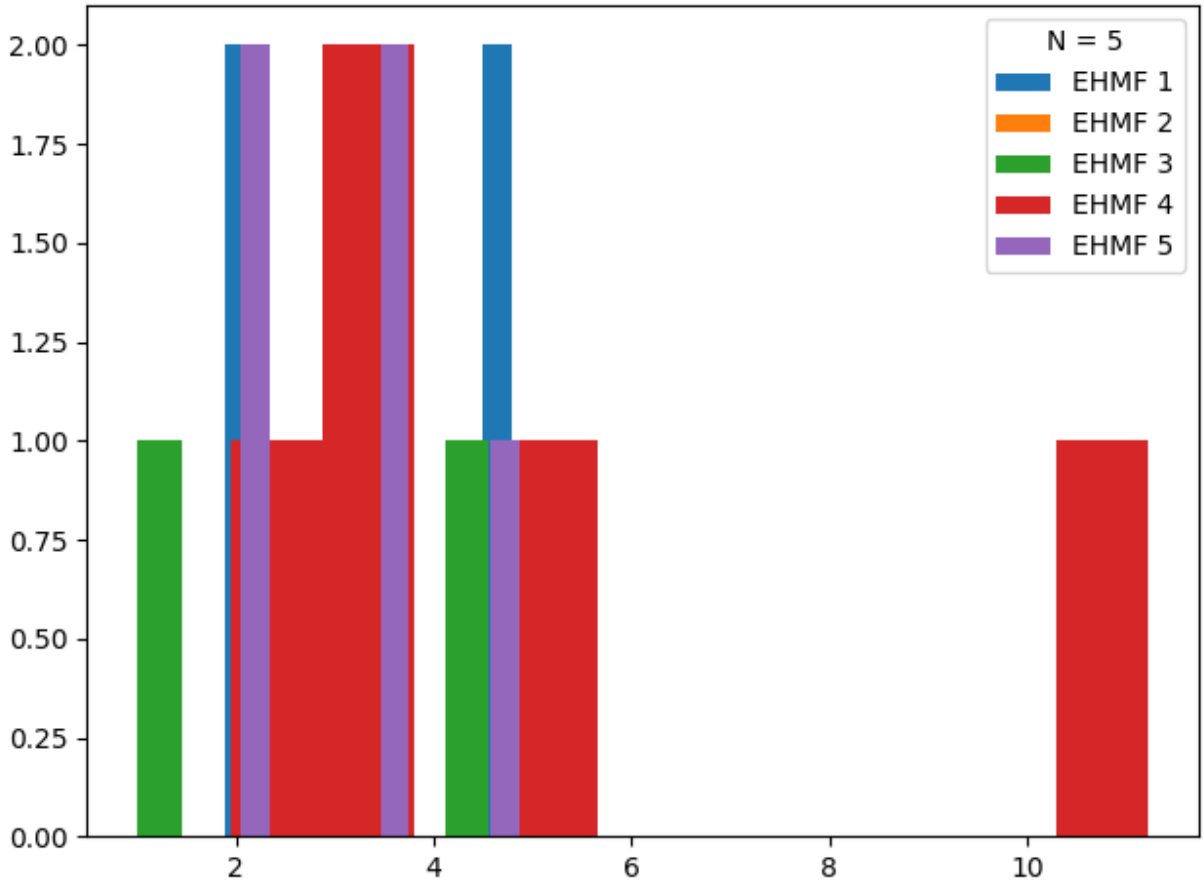
1.7447695628249114

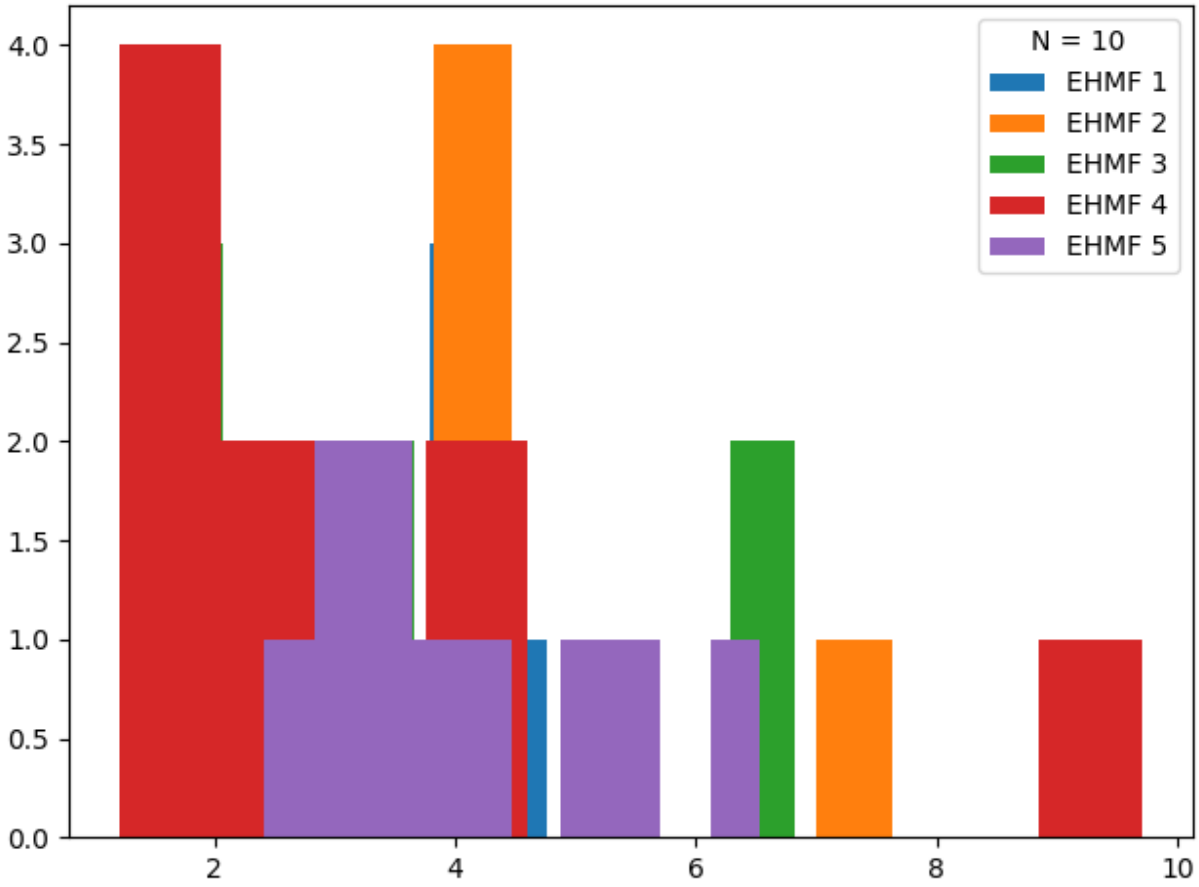
3.672060748850897

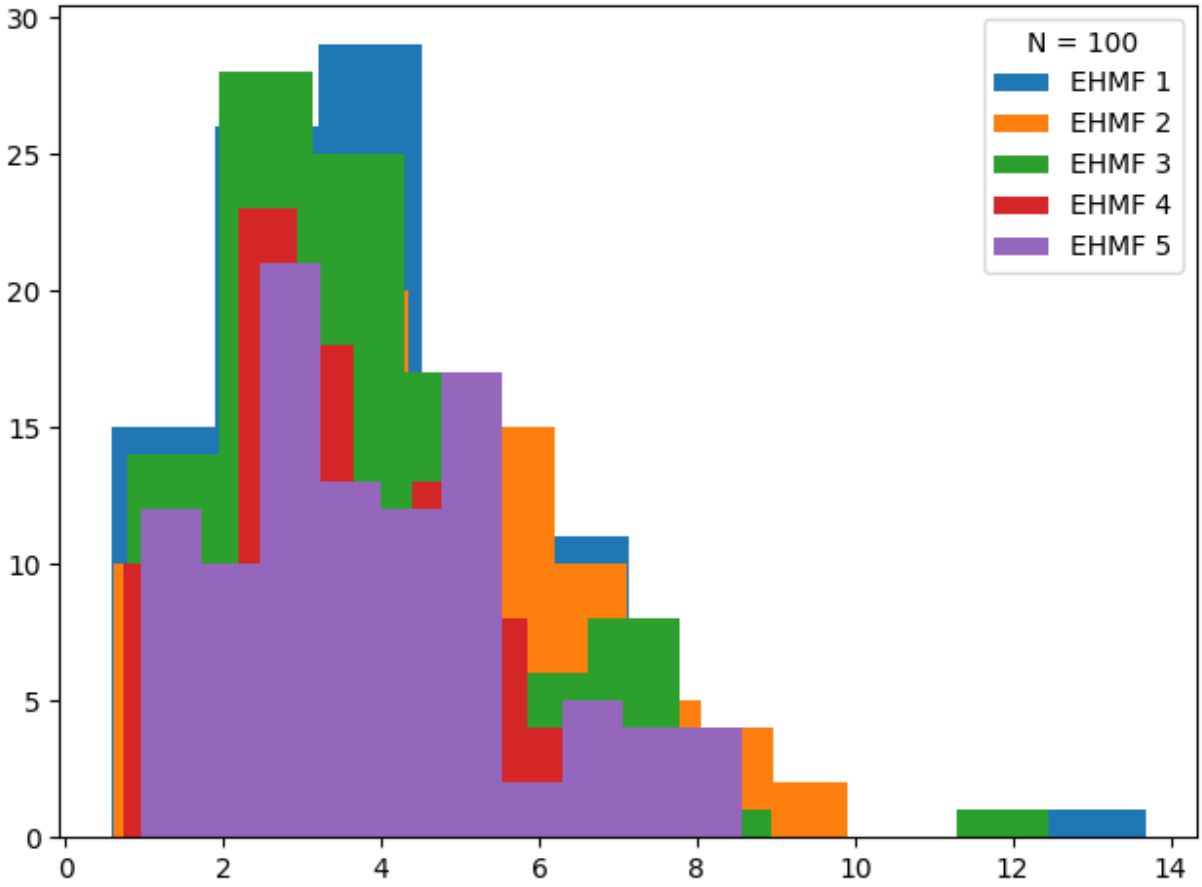
4.762229096535917

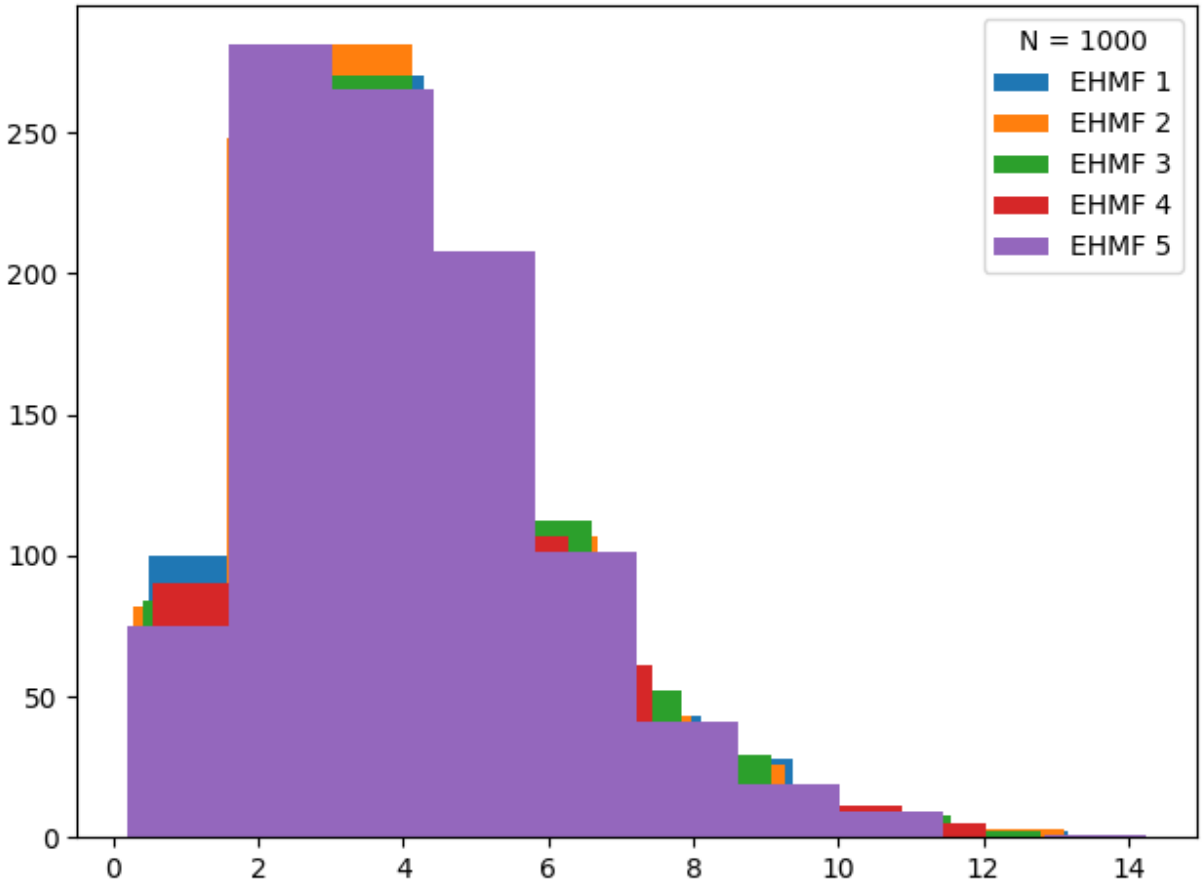
Построение гистограммы и полигона частот

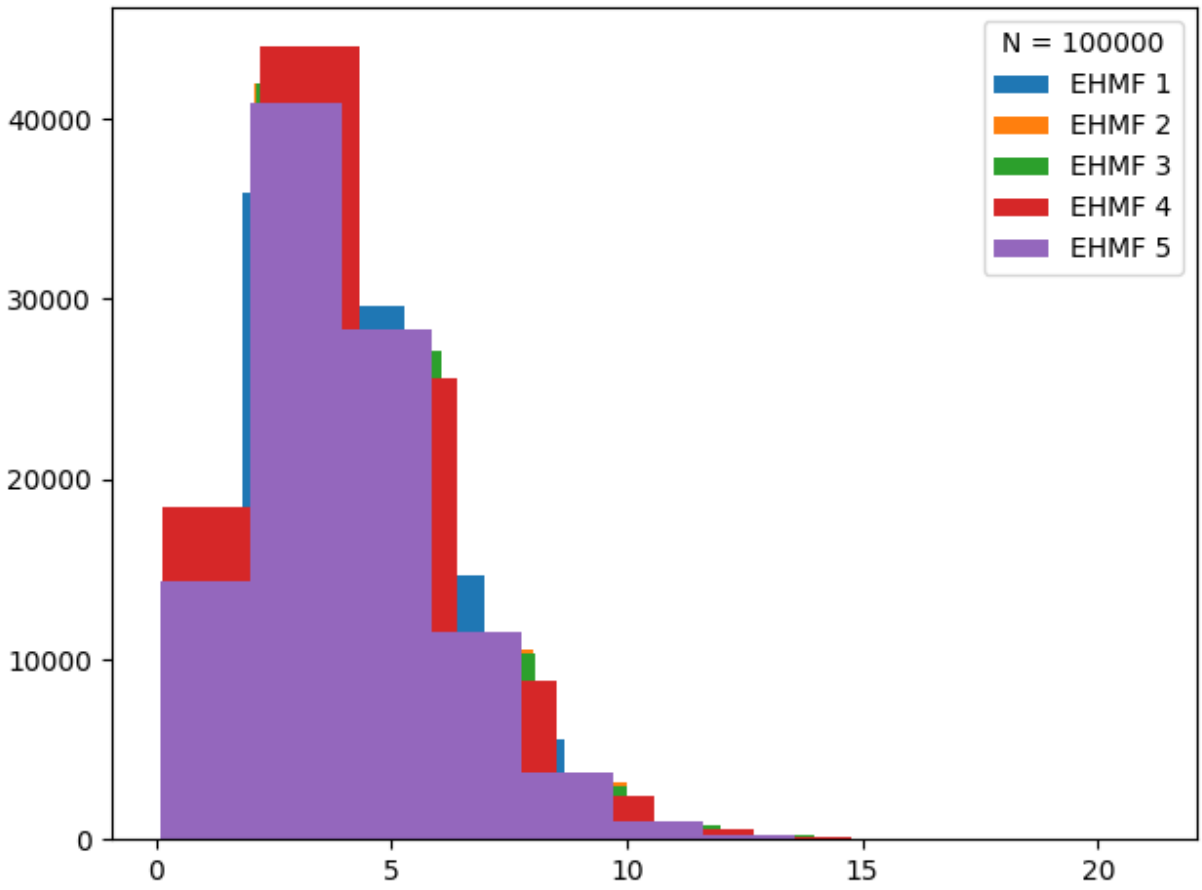
```
fgfg = sampaling(n,k,l,5)
for i in range(5):
    plt.hist(fgfg[i],label='EHMF {}'.format(i+1))
plt.legend(title='N = {}'.format(n))
plt.show()
```



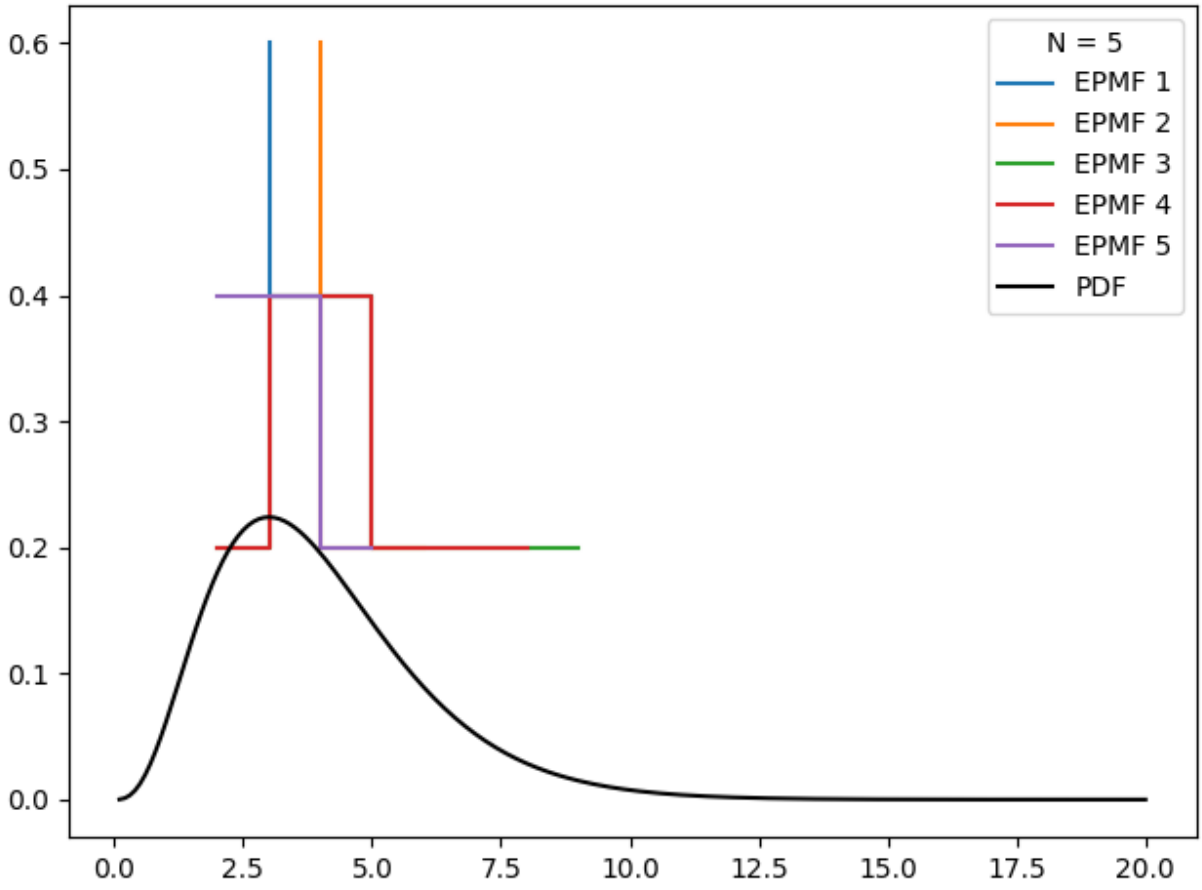

N = 10
EHMF 1
EHMF 2
EHMF 3
EHMF 4
EHMF 5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2
4
6
8
10

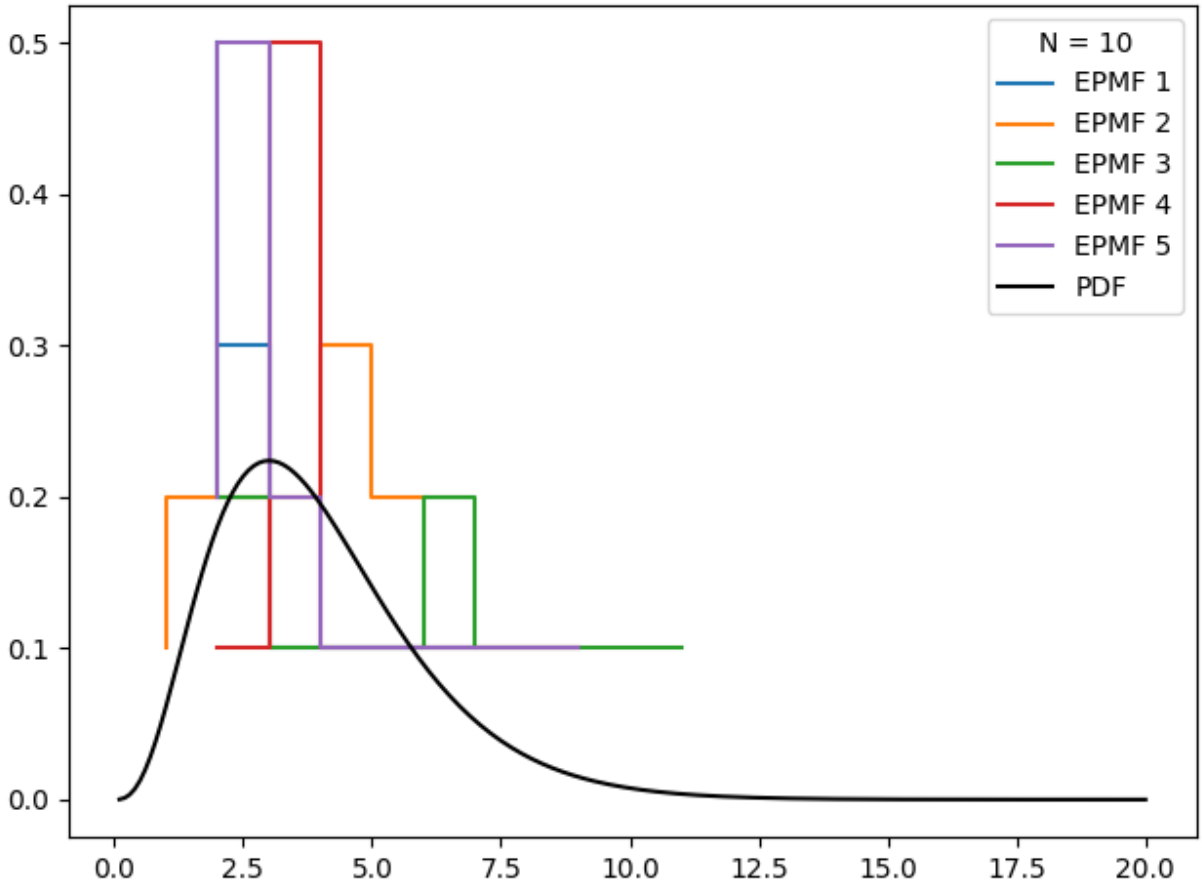

N = 100
EHMF 1
EHMF 2
EHMF 3
EHMF 4
EHMF 5
0
5
10
15
20
25
30
0
2
4
6
8
10
12
14

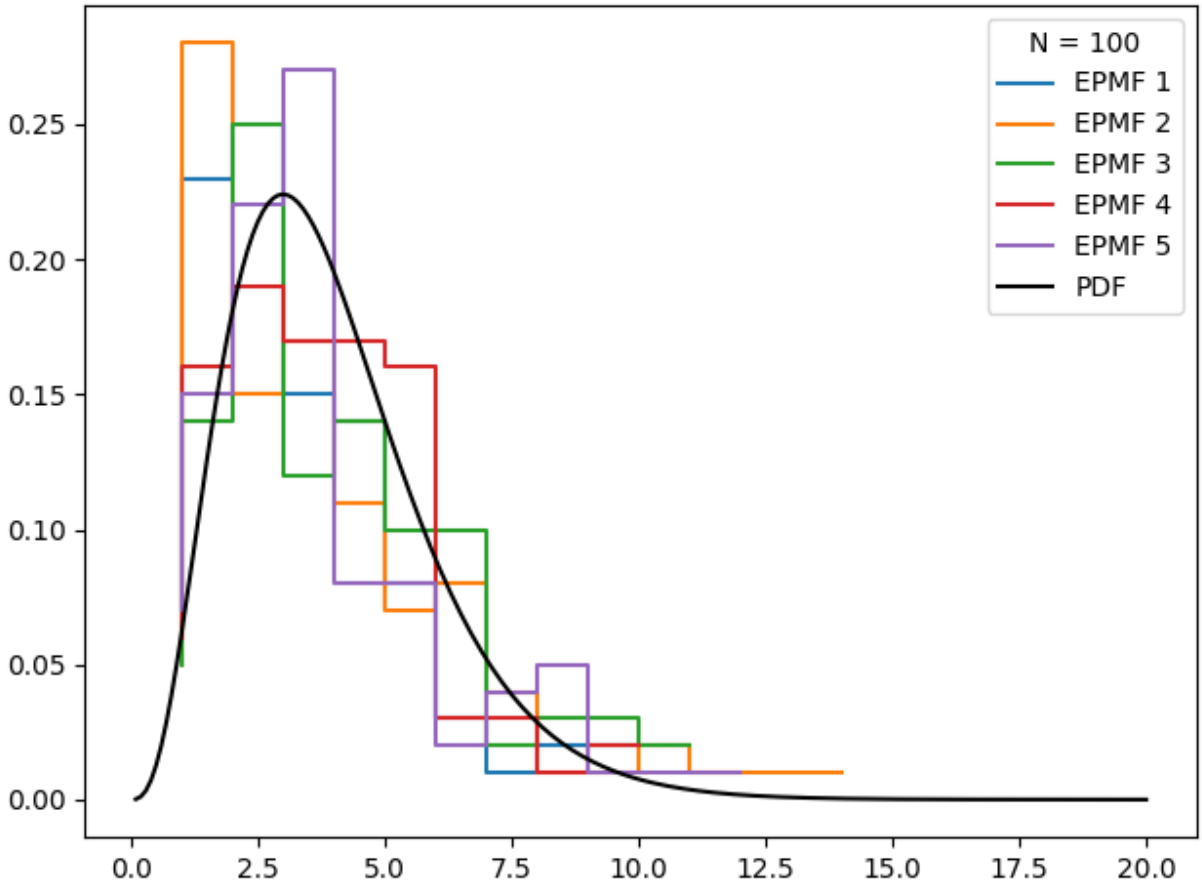

N = 1000
EHMF 1
EHMF 2
EHMF 3
EHMF 4
EHMF 5
0
50
100
150
200
250
0
2
4
6
8
10
12
14

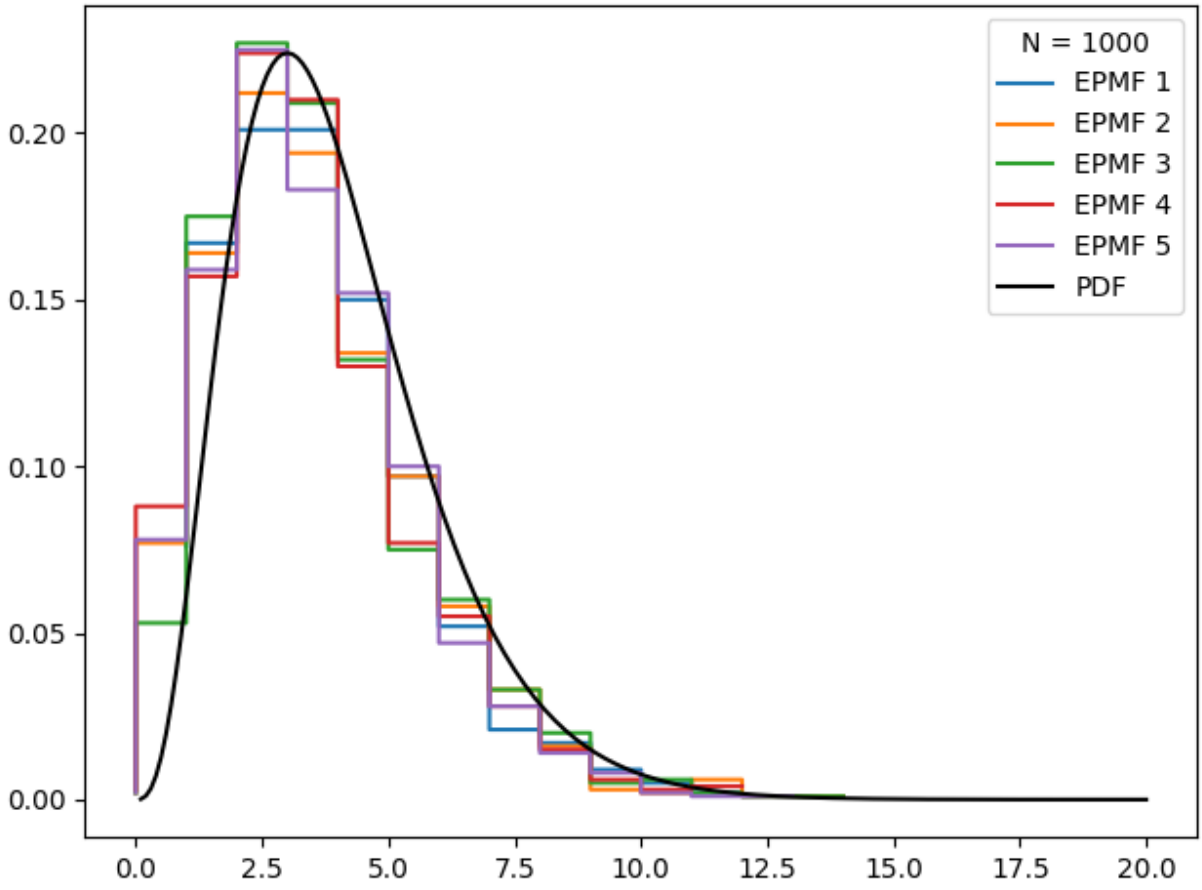

N = 100000
EHMF 1
EHMF 2
EHMF 3
EHMF 4
EHMF 5
40000
30000
20000
10000
0
0
5
10
15
20

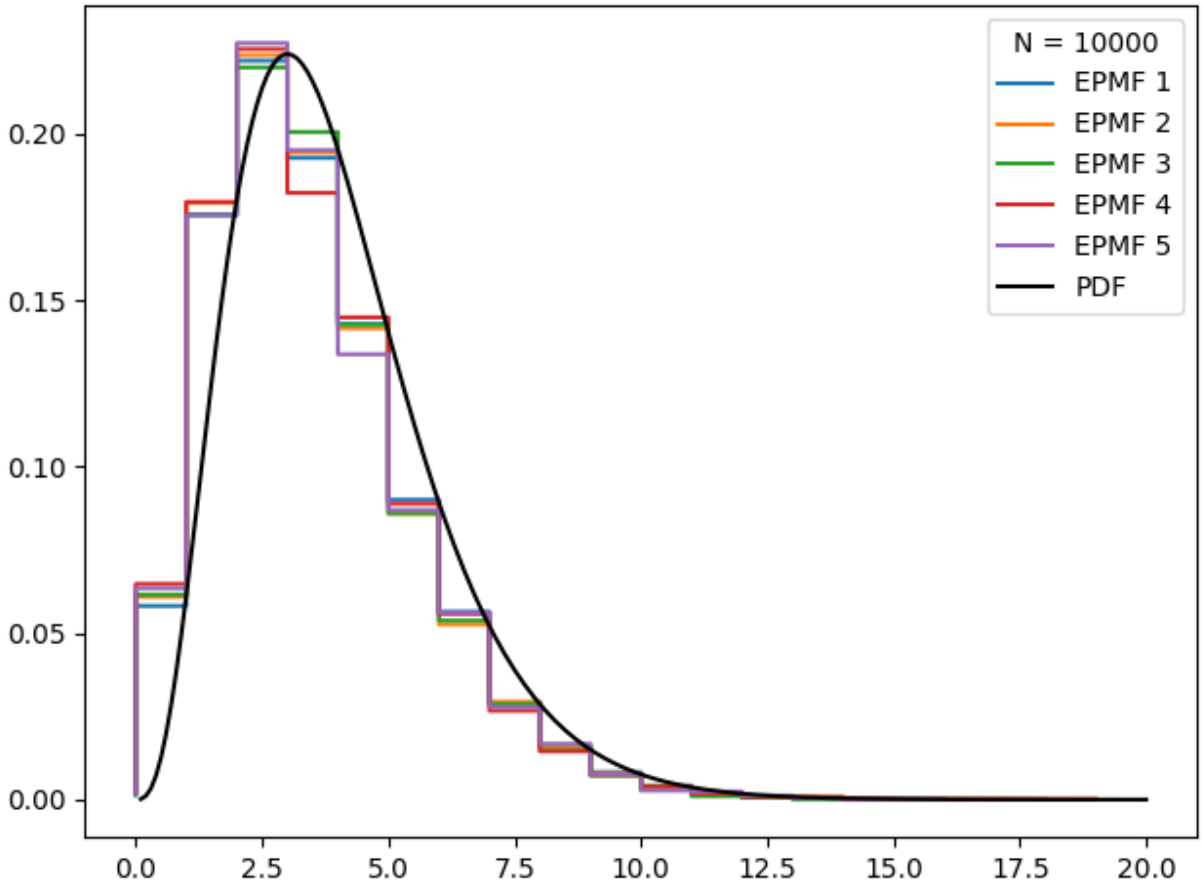
```
def rround (sampl):
    sampl1 = sampl
    for i in range(len(sampl)):
        for j in range(len(sampl1[i])):
            sampl1[i][j] = round(sampl1[i][j], 0)
    return sampl1

def EPMF (sampl):
    yv = arrey_creation(n, len(sampl))

    sss = rround(sampl)

    for i in range(len(sampl)):
        for j in range(len(sss[i])):
            yv[i][j] = (sampl[i]).count(sss[i][j]) / len(sampl[i])

    return yv

sampling_freq = rround(sampl)

yv = EPMF(sampl)


for i in range(5):
    plt.step(sampling_freq[i],yv[i], label='EPMF ' + str(i + 1))


xs=[]
ys=[]
zz=0
for i in range(200):
    zz += 0.1
    xs.append(zz)
for i in range(len(xs)):
    ys.append(PDF(xs[i],1,4))

plt.plot(xs,ys,color='black',label = 'PDF')


plt.legend(title='N = {}'.format(n))
plt.show()
```


N = 5
EPMF 1
EPMF 2
EPMF 3
EPMF 4
EPMF 5
PDF
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


N = 10
EPMF 1
EPMF 2
EPMF 3
EPMF 4
EPMF 5
PDF
0.5
0.4
0.3
0.2
0.1
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


N = 100
EPMF 1
EPMF 2
EPMF 3
EPMF 4
EPMF 5
PDF
0.25
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


N = 1000
EPMF 1
EPMF 2
EPMF 3
EPMF 4
EPMF 5
PDF
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


N = 10000
EPMF 1
EPMF 2
EPMF 3
EPMF 4
EPMF 5
PDF
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0