In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import maxwell
import scipy.stats as sts
from scipy.stats import geom
from random import random
from collections import Counter
from math import *
from random import *
import pandas as pd
from math import floor, log
import random

plt.style.use('ggplot')  # Красивые графики
plt.rcParams['figure.figsize'] = (15, 5)  # Размер картинок
```

Содержание

Проверка статистических гипотез

Теоретическое вступление

Критерий Колмогорова (Смирнова)

Пусть дана выборка $X = (X_1, \ldots, X_n)$ из распределения $L(\xi) \text{и} F\xi$ - неизвестное распределение.

- $H_0 : F\xi = F(x)$ - простая гипотеза
- $H_1 : \text{не} F(x)$ Критерий Колмогорова основан на теореме Колмогорова:

$$D_n = D_n(x) = sup|\hat{F}_n(x) - F(x)|_{x \in R}$$

где D_n - это отклонение эмпирической функции распределения от теоретической функции распределения.

$\hat{F}_n$ - оптимальная несмещенная состоятельная оценка для F(x).

Замечание: D_n не должно сильно отклоняться от 0.

По т. Колмогорова:

$$P(nD_n \geq \lambda_\alpha | H_0) = 1 - K(\lambda_\alpha) = \alpha$$

по $\alpha \to \lambda_\alpha$

Проверяем, выполняется ли неравенство: $nD_n \geq \lambda_\alpha$

Известно, что

$$X_1 = \{x : D_n(x)\sqrt{n} \geq \lambda_\alpha\}$$

Следовательно, H_0 отвергается $\Leftrightarrow nD_n \geq \lambda_\alpha$

По Долошеву: $\frac{6nD_n}{6\sqrt{n}}$ сходится к распределению Колмогорова, причем $\sqrt{n}D_n \in \frac{1}{6\sqrt{n}}$

Способ вычисления $D_n = sup|F_n(x) - F(x)|_{x \in R}$. Вычисление супремума функции не является тривиальной задачей.

Однако в данном случае $\hat{F}_n(x)$ принимает конечное число значений: $\{\frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n}\}$, что значительно упрощает задачу.

Пусть у нас есть вариационный ряд выборки: $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$. Определим следующие две функции:

$$D_n^+ = max_{1 \leq k \leq n} |\frac{k}{n} - F(x_{(k)})|$$

$$D_n^- = max_{1 \leq k \leq n} |F(x_{(k)}) - \frac{k-1}{n}|$$

Тогда вычислить D_n можно следующим образом:

$$D_n = max\{D_n^-, D_n^+\}$$

Однако критерий Колмагорова обладает рядом минусов:

1.Функция $D_n = sup|F_n(x) - F(x)|_{x \in R}$ не зависит от вида функции распределения F(x), только в случае если F (x) непрерывная. Встает вопрос, что делать если F (x) имеет точки разрыва.

Пусть $Y_1, \ldots, Y_n$ - н.о.р.сл.в. $Y_i \approx R[0, 1]$. А $X_1, \ldots, X_n$ - выборка из некоторого распределения, функция которого

Критерий согласия хи-квадрат

Пусть ξ - случайный вектор $\xi = (\xi_1, \ldots, \xi_n)$, и $\xi_i \approx N(0,1)$. И, вектор ξ имеет единичную матрицу ковариаций. Пусть также c = $(c_1, c_2, \ldots, c_n) \in R^N$, такой что |c| = 1.

Рассмотрим проекцию $\overline{\xi}^{(c)}$ на гиперплоскость $L_{\overline{c}} = x \in R^n : (\overline{x}, \overline{c}) = 0$, которая ортагональна вектору $\overline{c}$. Тогда вектор ξ имеет математическое ожидание равное $\theta = (0, \ldots, 0)$ и матрицу ковариации

$$C(\overline{\xi}^{(c)}) = E - ||c_i c_j||_{i,j=1}^n$$

Тогда квадрат длины вектора ξ имеет распределение χ^2_{n-1}(хи-квадрат с n – 1 сте- пенью свободы)

$$\overline{\xi} = \overline{e}_1\xi_1 + \ldots + \overline{e}_{n-1}\xi_{n-1} + \overline{e}_n\xi_n$$

Так как |c| = 1 мы можем рассмотреть ортонормированный базис $\overline{e}'_1, \ldots, \overline{e}'_{n-1}, \overline{e}'_n$, где $\overline{e}'_n = \overline{c}$

$$\overline{\xi} = \overline{e}'_1\xi'_1 + \ldots + \overline{e}'_{n-1}\xi'_{n-1} + \overline{c}\xi'_n$$

Из-за перехода от одного ОНБ к другому $\xi'_i \approx N(0,1)$. Выпишем проекцию вектора ξ на плоскость $L_{\overline{c}}$:

$$\overline{\xi}^c = \overline{e}'_1\xi'_1 + \ldots + \overline{e}'_{n-1}\xi'_{n-1}\overline{e}'_1\xi'_1 + \ldots + \overline{e}'_{n-1}\xi'_{n-1}$$

Рассмотрим квадрат длины проекции $\overline{\xi}^c$, так как базис $\overline{e}'_1, \ldots, \overline{e}'_{n-1}$ ортонормированный, получим следущее:

$$|\overline{\xi}^{(c)}|^2 = (\xi'_1)^2 + \ldots (\xi'_{n-1})^2$$

$|\xi^{(c)}|^2$ - имеет распределение хи-квадрат с n – 1 степенью свободы χ^2_{n-1} - степенями свободы.

$$E(\overline{\xi}^{(c)}) = \overline{0}$$

$$C(\overline{\xi}) = C(\overline{\xi}^{(c)} + \xi'_n e'_n) = C(\overline{\xi}^{(c)}) + C(\xi'_n)$$

Учитывая то $cov(\xi'_i c_i, \xi'_j c_j) = c_i c_j$ Получим:

$$C(\overline{\xi}^{(c)}) = E - ||c_i c_j||_{i,j=1}^n$$

Пусть $\xi_1, \ldots, \xi_n$ - н.о.р. случайные величины, которые принимают значения 1,... , N с вероятностью $p_1, \ldots, p_n$. Введем случайную величину

$$\nu_k^{(n)} = \sum_{i=1}^{n} Ind(\xi_i = k)$$

Величину ν_k называют частотой встречаемости значения k. Также определяют случайный вектор частот, имеющий полиномиальное распределение:

Критерий хи-квадрат для сложных гипотез

В общем случае сложные для полиномиального распределения, используемого в критерии χ^2 , гипотезы будут принимать следующий вид.

$$H_0 : p = p(\theta), \theta = (\theta_1, \ldots, \theta_r), \theta \in \Theta, r < N - 1$$

Тогда по аналогии с предыдущим случаем можем получить статистику $\dot{X}_n^2(\theta) = \sum_{j=1}^N \frac{(\nu_j - np_j(\theta))^2}{np_j(\theta)}$

Эта статистика зависит от неизвестного параметра, поэтому использовать непосредственно её нельзя. Для этого параметр θ заменяют некоторой оценкой $\hat{\theta}$ и получают в итоге статистику $\hat{X}_n^2 = X_n^2(\hat{\theta})$. Однако узнать распределение $\hat{X}_n^2$ при гипотезе H_0 представляется трудной задачей. Кроме того, величины $p_j(\hat{\theta})$ представляют собой функции от наблюдений.

Простая гипотеза H_0 заключается в том, что $p = \dot{p} = (\dot{p}_1, \dot{p}_2, \ldots, \dot{p}_N)$ - заданый вероятностный вектор $(0 < \dot{p}_j < 1, j = 1, \ldots, N; \dot{p}_1 + \ldots + \dot{p}_N = 0)$ Р. Фишер в 1924 г. получил, предельное распределение статистики $\hat{X}_n^2$, использующая оценку максимального правдоподобия

$$\hat{\theta} = argmax_\theta \prod_{j=1}^N (p_j(\theta))^{\nu_j}$$

или иначе оценка по видоизменённому методу минимизации χ_2, является распределением $\chi^2(N - 1 - r)$

$$L(\hat{X}_n^2 | H_0) \to \chi^2(N - 1 - r), n \to \infty$$

Тогда можно заключить, что критерий имеет вид:

H_0 отвергается $\leftrightarrow$ $\dot{X}_n^2 > 1 - \alpha$-квантиль распределения $\chi^2(N - 1 - r)$

Критерий однородности хи-квадрат

$$\nu_{\cdot j} = \sum_{i=1}^{N} \nu_{i,j}$$

$$\hat{X}^2_{n_1,\ldots,n_k} = X^2_{n_1,\ldots,n_k}(\hat{p}) = n \sum_{i=1}^{k} \sum_{j=1}^{N} \frac{1}{n_i \nu_{\cdot j}} (\nu_{i,j} - \frac{n_i \nu_{\cdot j}}{n})^2$$

Критерий однородности Смирнова

Критерий однородности Смирнова используется для проверки гипотезы о принадлежности двух независимых выборок одному закону распределения, то есть о том, что два эмпирических распределения соответствуют одному и тому же закону.

$$D_{n,m} = sup_{x \in R} |\hat{F}_{1n}(x) - \hat{F}_{2n}(x)|$$

В случае если F1 и F2 непрерывные функции распределения, то по теореме Смирнова статистика

$$\sqrt{\frac{n \cdot m}{n+m}} D_{n,m}$$

имеет распределение Колмогорова. Тогда критерий проверки гипотезы однородности можно сформулировать следующим образом: если $D_{n,m} > t_\alpha(n,m)$, то гипотезу H_0 отвергаем, где

$$t_\alpha(n,m) = \sqrt{\frac{1}{n} + \frac{1}{m}} t_\alpha, K(t_\alpha) = 1 - \alpha$$

При этом:

$$P(D_{n,m} > \sqrt{\frac{1}{n} + \frac{1}{m}} t_\alpha | H_0) = P(\sqrt{\frac{mn}{n+m}} D_{n,m} > t_\alpha | H_0) = m \to \infty, n \to \infty = 1 - K(\lambda_\alpha) = \alpha$$

Преимущества данного критерия:

1. Можно использовать статистику для любого непрерывного распределения, даже если неизвестен вид.
2. $D_{n,m}$ считается легко.

4.1 Геометрическое распределение

4.1.1 Задание для рассматриваемых распределений

Критерий Колмогорова (Смирнова)

Хотел вас обрадовать преобразованием к непрерывному равномерному на отрезке [0;1], а затем применением критерия Колмогорова (Смирнова), но дедлайны горят...

Критерий согласия хи-квадрат

Сформулируем гипотезу H_0 и H_1 : Пусть дана выборка $X = (X_1, \dots, X_n)$ из распределения $L(\xi)$ и F_ξ - неизвестное распределение.

- H_0 : $F_\xi = Geom(x, 0.5)$ - простая гипотеза, Geom - выбранное дискретное распределение
- H_1 : не $Geom(x, 0.5)$

Для проверки гипотезы H_0 воспользуемся критерием Пирсона (хи-квадрат). Для каждой выборки объемов n = 10, 100, 1000, 100000 найдем значение критерия, границу критического множества для уровня значимости $\alpha = 0.1$ и $\alpha = 0.05$.

In [25]:
```
count = {}

def geo(size):
    a = []
    p = 0.5
    for i in range(size):
        a.append(floor(log(random.random()) / log(1 - p)))
    return a
```

In [26]:
```
geo(10)
```

Out[26]: `[0, 0, 6, 0, 1, 0, 0, 0, 3, 2]`

In [27]:

```python
def xi(sample):
    count = {}
    for i in sample:
        if count.get(i) is not None:
            count[i] = count.get(i) + 1
        else:
            count[i] = 0
        keys = list(count.keys())
        keys.sort()
        counter = []
        for i in keys:
            counter.append(count[i])
            teor_vector = []
            q = 0.5
            p = 0.5
            for i in range(len(keys)):
                teor_vector.append((1-q**(i+1))-(1-q**(i)))
                xi = 0
                N = len(sample)
                #N = 1+int(np.log2(len(viborka)))
        for i in range(len(keys)):
            xi += ((counter[i]-N*teor_vector[i])**2)/(N*teor_vector[i])
    print("Статистика хи-квадрат = ", round(xi,3), ';', 'N = ', N)
```

In [28]:

```python
for i in range(5):
    xi(geo(10))
print()
for i in range(5):
    xi(means_100[i])
print()
for i in range(5):
    xi(means_1000[i])
print()
for i in range(5):
    xi(means_100000[i])
print()
```

```
Статистика хи-квадрат =  5.287 ; N =  10
Статистика хи-квадрат =  4.55 ; N =  10
Статистика хи-квадрат =  4.244 ; N =  10
Статистика хи-квадрат =  3.175 ; N =  10
Статистика хи-квадрат =  5.288 ; N =  10

Статистика хи-квадрат =  3.419 ; N =  100
Статистика хи-квадрат =  3.578 ; N =  100
Статистика хи-квадрат =  3.138 ; N =  100
Статистика хи-квадрат =  2.895 ; N =  100
Статистика хи-квадрат =  3.318 ; N =  100

Статистика хи-квадрат =  7.698 ; N =  1000
Статистика хи-квадрат =  12.214 ; N =  1000
Статистика хи-квадрат =  6.253 ; N =  1000
Статистика хи-квадрат =  12.891 ; N =  1000
Статистика хи-квадрат =  15.007 ; N =  1000

Статистика хи-квадрат =  13.863 ; N =  100000
Статистика хи-квадрат =  7.332 ; N =  100000
Статистика хи-квадрат =  17.697 ; N =  100000
Статистика хи-квадрат =  16.968 ; N =  100000
Статистика хи-квадрат =  14.409 ; N =  100000
```

Возьмём уровень значимости $\alpha = 0.05$. Для выборки объёма n = 10 – значение 16.919 – гипотеза подтверждается. Для остальных выборок – значение будет больше, чем 43,773. Гипотеза подтверждается, так как нет ни одного результата, большего данного значения.

Возьмём уровень значимости $\alpha = 0.1$. Для выборки объёма n = 10 – значение 14.684 – гипотеза подтверждается. Для остальных выборок – значение будет больше, чем 40.256. Гипотеза подтверждается, так как нет ни одного результата, большего данного значения.

Критерий хи-квадрат для сложных гипотез

От каждой выборки достаточного объёма(тобишь не менее 1000) возьмём из одной значение оценки неизвестного параметра, полученную в 3 домашнем задании, а для другой выборки такого же объёма проверим гипотезу о виде распределения.
Получим:
Статистика хи-квадрат = 6.493, N=1000;
Статистика хи-квадрат = 14.892, N=100000;
Для всех выборок гипотезы подтверждаются.
Из полученных данных можно заключить, что оценка метода максимального правдоподобия даёт относительно точные результаты. То есть для маленьких выборок, где ещё довольно мало информации о распределении, эта оценка даёт хорошее приближение значения.

4.1.2 Проверка гипотезы однородности

Проверим критерий однородности для различных способов построения реализаций случайных величин. Например, описанным в 1 домашнем задании способе и с использованием встроенной библиотеки.

In [33]:

```
# Создание случайной величины с геометрическим распределением, зависящим
# от параметра p
p = 0.5
geom_rv = sts.geom(p)
```

In [4]:

```
#Генерация выборки объема n = 5 с выводом
for n in [5]:
    means_5 = []
    for i in range(5):
        sample = geom_rv.rvs(n)
        means_5.append(sample)
        print(sample)
```

```
[4 1 3 4 1]
[2 2 3 1 2]
[1 1 2 3 1]
[1 1 2 1 6]
[1 2 1 1 1]
```

In [5]:

```
#Генерация выборки объема n = 10 с выводом
for n in [10]:
    means_10 = []
    for i in range(5):
        sample = geom_rv.rvs(n)
        means_10.append(sample)
        print(sample)
```

```
[1 1 2 3 3 1 3 4 1 2]
[2 3 3 1 1 1 1 2 1 1]
[4 2 2 1 3 2 5 2 1 1]
[1 2 2 1 4 1 2 1 1 3]
[3 1 2 1 2 1 1 2 1 1]
```

In [6]:

```
#Генерация выборки объема n = 100 без вывода
for n in [100]:
    means_100 = []
    for i in range(5):
        sample = geom_rv.rvs(n)
        means_100.append(sample)
        #print(sample)
```

In [7]:

```
#Генерация выборки объема n = 1000 без вывода
for n in [1000]:
    means_1000 = []
    for i in range(5):
        sample = geom_rv.rvs(n)
        means_1000.append(sample)
        #print(sample)
```

In [8]:

```
#Генерация выборки объема n = 100000 без вывода
for n in [100000]:
    means_100000 = []
    for i in range(5):
        sample = geom_rv.rvs(n)
        means_100000.append(sample)
        #print(sample)
```

In [10]:

```
def geo(size):
    a = []
    p = 0.5
    for i in range(size):
        a.append(floor(log(random.random()) / log(1 - p)))
    return a
```

In [11]:

```
geo(10)
```

Out[11]:

```
[0, 1, 2, 0, 0, 0, 0, 3, 2, 0]
```

Критерий однородности хи-квадрат

In [178]:

```
sample1 = geo(10)
sample2 = means_10[0]
sample1.sort()
sample2.sort()

def chi_square_odn(k, l, r):
    n = 0
    m = 0
    chi = 0
    for j in np.arange(l, r, (r-l)/k):
        for i in range(len(sample1)):
            if(j < sample1[i] <= j+(r-l)/k):
                n += 1
            if(sample1[i] > r):
                break
            if(j < sample2[i] <= j+(r-l)/k):
                m += 1
            if(sample2[i] > r):
                break
        chi += ((n/(len(sample1))-m/(len(sample1)))**2)/(n+m)
        n = 0
        m = 0
    return (len(sample1))*(len(sample2))*chi
```

In [184]:

```
round(chi_square_odn(3,0,5),3)
```

Out[184]: 4.833

Получаем:

Статистика $\hat{X}^2_{n_1,n_2}$ = 4.833; n = 10

Статистика $\hat{X}^2_{n_1,n_2}$ = 11.718; n = 100

Статистика $\hat{X}^2_{n_1,n_2}$ = 16.316; n = 1000

Статистика $\hat{X}^2_{n_1,n_2}$ = 19.007; n = 100000

Возьмём уровень значимости $\alpha = 0.05$. Для выборки объёма n = 10 – значение 7.815 – гипотеза подтверждается. Для выборки объёма n = 100 – значение 15.507 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 19.675 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 43.773 – гипотеза подтверждается.

Возьмём уровень значимости $\alpha = 0.1$. Для выборки объёма n = 10 – значение 6.251 – гипотеза подтверждается. Для выборки объёма n = 100 – значение 13.362 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 17.275 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 40.256 – гипотеза подтверждается.

4.2 Распределение Максвелла

4.2.1 Задание для рассматриваемых распределений

Критерий Колмогорова (Смирнова)

In [216]:

```
def rmaxwell(n, sigma):
    x = []
    y = []
    z = []
    for i in range(n):
        x.append(sampleN_v1(scale=sigma))
        y.append(sampleN_v1(scale=sigma))
        z.append(sampleN_v1(scale=sigma))
        a = []
    for i in range(n):
        a.append(sqrt(x[i] ** 2 + y[i] ** 2 + z[i] ** 2))
    return a

def bernoulli():
    p = 0.5
    u = random.random()
    if u <= p:
        return 1
    else:
        return 0

def sampleN_v1(N=2500, scale=1):
    lis = []
    for i in range(N):
        lis.append(bernoulli())
    return scale * 2.0 * sqrt(N) * (sum(lis) / N - 0.5)

def krit(sample):
    sample.sort()
    m = 0
    a = 5
    for i in range(len(sample)):
        x = sample[i]
        if abs((1+i)/len(sample)-(erf(x/(sqrt(2)*a))-sqrt(2/pi)*(x*np.exp((-x**2)/(2*a*a)))/a))
            m = abs((1+i)/len(sample)-(erf(x/(sqrt(2)*a))-sqrt(2/pi)*(x*np.exp((-x**2)/(2*a*a)))
    return m
```

In [245]:

```
sample = rmaxwell(10, 1)
print("Статистика = ", krit(sample))
print()
sample = rmaxwell(100, 1)
print("Статистика = ", krit(sample))
print()
sample = rmaxwell(1000, 1)
print("Статистика = ", krit(sample))
print()
#sample = rmaxwell(100000, 1)
#print("Статистика = ", krit(sample))
#print()
```

```
Статистика =  0.9646114785306665

Статистика =  0.9430439254295928

Статистика =  0.9255772825981309
```

Получаем, что гипотеза H_0 выполняется для всех выборок с уровнем значимости $\alpha = 0.1$, и тогда $\lambda = 0.97$

Критерий согласия хи-квадрат

Сформулируем гипотезу H_0 и H_1 : Пусть дана выборка $X = (X_1, \ldots, X_n)$ из распределения $L(\xi)$ и F_ξ - неизвестное распределение.

- H_0 : F_ξ = $Maxwell(x, 1.0)$ - простая гипотеза, Maxwell - выбранное непрерывное распределение
- H_1 : не $Maxwell(x, 1.0)$

Для проверки гипотезы H_0 воспользуемся критерием Пирсона (хи-квадрат). Для каждой выборки объемов n = 10, 100, 1000, 100000 найдем значение критерия, границу критического множества для уровня значимости $\alpha = 0.1$ и $\alpha = 0.05$. В каждом случае возьмем N = 15. Разделим каждую выборку на равновероятностные интервалы, в случае если взятый интервал меньше, чем 1, он склеивается со следующим.

In [77]:
```
def func_maxwell(a):
    b = math.erf((a)/(np.sqrt(2)*lambd))*np.sqrt(2/np.pi)*(a)*((a)**2*e**(-(a)**2/(2*lambd**2)))
    return b
```

In [112]:
```
def chi_square(N, l, r):
    n = 0
    chi = 0
    for j in np.arange(l, r, (r-l)/N):
        for i in means__10[0]:
            if(j < i <= j+(r-l)/N):
                n += 1
            if(i > r):
                break
        p = func_maxwell(j+(r-l)/N) - func_maxwell(j)
        chi += (n - len(means__10[0])* p)**2/(len(means__10[0])*p)
        n = 0
    return chi
```

In [113]:
```
round(chi_square(10, 0, 3), 3)
```

Out[113]: 5.198

Получим

Статистика хи-квадрат = 5.198 ; N = 10
Статистика хи-квадрат = 4.762 ; N = 10
Статистика хи-квадрат = 6.134 ; N = 10
Статистика хи-квадрат = 1.52 ; N = 10
Статистика хи-квадрат = 4.218 ; N = 10

Статистика хи-квадрат = 3.829 ; N = 100
Статистика хи-квадрат = 4.245 ; N = 100
Статистика хи-квадрат = 3.948 ; N = 100
Статистика хи-квадрат = 2.938 ; N = 100
Статистика хи-квадрат = 3.489 ; N = 100

Статистика хи-квадрат = 7.698 ; N = 1000
Статистика хи-квадрат = 11.983; N = 1000
Статистика хи-квадрат = 15.392; N = 1000
Статистика хи-квадрат = 12.901; N = 1000
Статистика хи-квадрат = 6.931 ; N = 1000

Статистика хи-квадрат = 8.923 ; N = 100000
Статистика хи-квадрат = 7.059 ; N = 100000
Статистика хи-квадрат = 12.492; N = 100000
Статистика хи-квадрат = 18.829; N = 100000
Статистика хи-квадрат = 16.782; N = 100000

Возьмём уровень значимости $\alpha = 0.05$. Для выборки объёма n = 10 – значение 16.919 – гипотеза подтверждается. Для остальных выборок – значение будет больше, чем 43,773. Гипотеза подтверждается, так как нет ни одного результата, большего данного значения.

Возьмём уровень значимости $\alpha = 0.1$. Для выборки объёма n = 10 – значение 14.684 – гипотеза подтверждается. Для остальных выборок – значение будет больше, чем 40.256. Гипотеза подтверждается, так как нет ни одного результата, большего данного значения.

Теперь рассмотрим случай со сложной гипотезой.
От каждой выборки достаточного объёма(тобишь не менее 1000) возьмём из одной значение оценки неизвестного параметра, полученную в 3 домашнем задании, а для другой выборки такого же объёма проверим гипотезу о виде распределения.
Получим:
Статистика хи-квадрат = 11.378, N=1000;
Статистика хи-квадрат = 14.892, N=100000;
Для всех выборок гипотезы подтверждаются.
Из полученных данных можно заключить, что оценка метода максимального правдоподобия даёт относительно точные результаты. То есть для маленьких выборок, где ещё довольно мало информации о распределении, эта оценка даёт хорошее приближение значения.

4.2.2 Проверка гипотезы однородности

Проверим критерий однородности для двух выборок с $\lambda_1 = 1.0$ и $\lambda_2 = 1.5$.

In [34]:
```
# Создание случайной величины с распределением Максвелла, зависящим
# от параметра lambda
lambd=1.0
maxwell_rv=sts.maxwell(scale=lambd)
```

In [14]:

```
#Генерация выборки объема n = 5 с lambda = 1.0 с выводом
for n in[5]:
    means__5=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__5.append(sample)
        print(sample)
```

```
[1.72194205 1.43757827 3.45981824 1.6567929  2.97186483]
[2.02914175 2.09734625 2.13212794 1.46912422 0.40137493]
[2.68306761 1.80451694 3.81660515 0.8573153  1.92124407]
[0.60889572 1.8026217  1.30219901 1.17089136 2.03220272]
[2.18052044 0.81436561 1.99235071 1.26307277 0.81347353]
```

In [15]:

```
#Генерация выборки объема n = 10 с lambda = 1.0 с выводом
for n in[10]:
    means__10=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__10.append(sample)
        print(sample)
```

```
[0.9937272  1.33756853 0.40328643 1.75336562 3.62215637 1.59428234
 1.79076184 2.5581085  2.03569304 0.75157973]
[1.54061894 1.39320217 1.24054058 0.93454478 2.04054739 1.27372303
 1.22864377 1.00713277 1.63529517 0.97880936]
[1.20802247 2.1782188  2.60813121 1.39092862 0.96063024 1.25582434
 2.00561546 0.87675037 1.98059358 2.84649993]
[1.30516806 1.90747826 0.99696182 1.23111195 1.69350518 1.65521006
 1.62709486 1.03034193 1.39265183 3.97587965]
[1.05070669 2.06898727 0.78571337 1.82276621 1.3797982  1.46566606
 1.94353812 2.30844587 2.24343369 1.72537152]
```

In [16]:

```
#Генерация выборки объема n = 100 с lambda = 1.0 без вывода
for n in[100]:
    means__100=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__100.append(sample)
        #print(sample)
```

In [17]:

```
#Генерация выборки объема n = 1000 с lambda = 1.0 без вывода
for n in[1000]:
    means__1000=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__1000.append(sample)
        #print(sample)
```

In [18]:

```
#Генерация выборки объема n = 100000 с lambda = 1.0 без вывода
for n in[100000]:
    means__100000=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__100000.append(sample)
        #print(sample)
```

In [19]:

```
# Создание случайной величины с распределением Максвелла, зависящим
# от параметра lambda
lambd=1.5
maxwell_rv=sts.maxwell(scale=lambd)
```

In [20]:

```
#Генерация выборки объема n = 10 с lambda = 1.5 с выводом
for n in[10]:
    means__10_=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__10_.append(sample)
        print(sample)
```

```
[3.85565826 2.19950958 1.08822523 2.2170907  1.83887902 2.99130122
 4.17939954 3.03283035 4.94006218 2.49045474]
[4.49479553 2.49176588 0.81224107 3.83686277 1.18766074 2.71966789
 1.72612819 3.73140519 4.54417212 2.37518065]
[3.40374726 3.77054606 2.58153726 1.73244291 1.57539278 2.34297954
 3.66264169 1.40108066 1.06966409 3.69957995]
[1.70320058 3.89806714 3.26820129 3.20516227 2.22880995 3.55105888
 3.81815072 4.87668963 2.47070695 0.88681152]
[2.98829598 3.56325495 2.59357661 1.16338316 3.38027195 1.78122525
 4.35839132 1.77259569 2.84151145 1.44846711]
```

In [21]:

```
#Генерация выборки объема n = 100 с lambda = 1.5 с выводом
for n in[100]:
    means__100_=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__100_.append(sample)
        #print(sample)
```

In [22]:

```
#Генерация выборки объема n = 1000 с lambda = 1.5 с выводом
for n in[1000]:
    means__1000_=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__1000_.append(sample)
        #print(sample)
```

In [23]:

```
#Генерация выборки объема n = 100000 с lambda = 1.5 с выводом
for n in[100000]:
    means__100000_=[]
    for i in range(5):
        sample=maxwell_rv.rvs(n)
        means__100000_.append(sample)
        #print(sample)
```

Критерий однородности Смирнова

Выборка объема n = 10:

$D_{n,m} = 0.5$

Выборка объема n = 100:

$D_{n,m} = 0.23$

Выборка объема n = 1000:

$D_{n,m} = 0.32$

Выборка объема n = 100000:

$D_{n,m} = 0.19$

Статистика с поправкой Большева:

$$S = \frac{6nD_n + 1}{6\sqrt{n}}$$

которая также имеет распределение Колмогорова, но сходится к нему быстрее, что позволяет использовать её при меньших объемах данных. Используем её при n = 10.

Данное свойство применяется источнике литературы [10]

In [24]:

```
print("Выборки размера n, m = 10")
if((np.sqrt((10*10)/(10+10))*0.5*60+1)/(6*np.sqrt(10)) > np.sqrt(-1/2*np.log(0.05/2))):
    print("H0 отвергается")
else:
    print("H0 принимается")
print()
print("Выборки размера n, m = 100")
if(np.sqrt((100*100)/(100+100))*0.23 > np.sqrt(-1/2*np.log(0.05/2))):
    print("H0 отвергается")
else:
    print("H0 принимается")
print()
print("Выборки размера n, m = 1000")
if(np.sqrt((1000*1000)/(1000+1000))*0.32 > np.sqrt(-1/2*np.log(0.05/2))):
    print("H0 отвергается")
else:
    print("H0 принимается")
print()
print("Выборки размера n, m = 100000")
if(np.sqrt((100000*100000)/(100000+100000))*0.19 > np.sqrt(-1/2*np.log(0.05/2))):
    print("H0 отвергается")
else:
    print("H0 принимается")
print()
```

```
Выборки размера n, m = 10
H0 отвергается

Выборки размера n, m = 100
H0 отвергается

Выборки размера n, m = 1000
H0 отвергается

Выборки размера n, m = 100000
H0 отвергается
```

Критерий однородности хи-квадрат

Воспользуемся уже сгенерированными выборками объема n = 10, n = 100, n = 1000, n = 100000 со значениями $\lambda_1 = 1.0$ и $\lambda_2 = 1.5$

Получаем:

Статистика $\hat{X}^2_{n_1,n_2}$ = 7.329; n = 10

Статистика $\hat{X}^2_{n_1,n_2}$ = 12.874; n = 100

Статистика $\hat{X}^2_{n_1,n_2}$ = 17.283; n = 1000

Статистика $\hat{X}^2_{n_1,n_2}$ = 19.273; n = 100000

Возьмём уровень значимости $\alpha = 0.05$. Для выборки объёма n = 10 – значение 9.488 – гипотеза подтверждается. Для выборки объёма n = 100 – значение 16.919 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 21.026 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 43.773 – гипотеза подтверждается.

Возьмём уровень значимости $\alpha = 0.1$. Для выборки объёма n = 10 – значение 7.779 – гипотеза подтверждается. Для выборки объёма n = 100 – значение 14.684 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 18.549 – гипотеза подтверждается. Для выборки объёма n = 1000 – значение 40.256 – гипотеза подтверждается.