

Домашнее задание 4. Проверка статистических гипотез

Ковалев Д., СКБ171

Критерий согласия хи-квадрат (Пирсона)

Проверяем гипотезу H_0 : случайная величина распределена по закону ξ с параметром θ .

Для проверки гипотезы с использованием этого критерия необходимо сгруппировать наблюдения. Допустим, есть n наблюдений, и мы разбиваем их по k интервалам с граничными точками $x_{(0)}, x_{(1)}, \dots, x_{(k)}$. Здесь $x_{(0)}$ — нижняя грань области распределения случайной величины, $x_{(k)}$ — верхняя грань.

Посчитаем теоретические вероятности попадания в i -й интервал: $p_i = F_\xi(x_i) - F_\xi(x_{i-1})$, $i = \overline{1, k}$. Пусть в i -й интервал попало n_i наблюдений, тогда для проверки гипотезы критерием хи-квадрат нужно измерить отклонение $\frac{n_i}{n}$ от p_i .

В данном критерии используется статистика $\chi^2 = n \sum_{i=1}^k \frac{\left(p_i - \frac{n_i}{n}\right)^2}{p_i}$. Для нахождения числа интервалов будем использовать формулу Старджесса: $k = \log_2 n + 1$ в случае непрерывных распределений, а для дискретных будем брать в качестве числа интервалов максимальное значение, которое принимает случайная величина

Эта статистика при $n \rightarrow \infty$ имеет распределение χ^2 с $r = k - 1$ степенями свободы.

Для выбранного уровня значимости α по таблице распределения χ^2 ищем критическое значение $\chi_{\alpha, r}^2$. Если посчитанная статистика больше критического значения, гипотеза H_0 отвергается, иначе — принимается.

Достоинство критерия: можно использовать как для дискретных, так и для непрерывных распределений.

Недостаток критерия: зависимость от объема выборки (рассматриваемая статистика имеет распределение χ^2 только при $n \rightarrow \infty$).

Критерий согласия хи-я сложной гипотезы

В этом случае у нас нет изначального предположения о параметре распределения θ , поэтому вместо него берется $\hat{\theta}$. Далее действовать нужно аналогично написанному ранее.

Критерий согласия Колмогорова-Смирнова (для абсолютно непрерывных распределений)

Проверяем гипотезу H_0 : случайная величина распределена по закону ξ с параметром θ .

В данном критерии используется статистика Колмогорова: $D_n = \max \left| \hat{F}_n(x) - F_\xi(x) \right|$

При $n \rightarrow \infty$ $\mathbb{P}(D_n \sqrt{n} \geq \lambda)$ стремится к распределению Колмогорова:

$$\mathbb{P}(\lambda) = 1 - \sum_{k=-\infty}^{+\infty} (-1)^k e^{-2k^2 \lambda^2}$$

Критерий: если $D_n \sqrt{n} \leq \lambda_\alpha$, то гипотеза принимается (α - уровень значимости, λ_α - табличное значение).

Достоинство критерия: критерий работает как для малых, так и для больших объемов выборок.

Недостаток критерия: критерий неприменим для дискретных распределений

Критерий согласия Колмогорова-Смирнова для сложной гипотезы (для абсолютно непрерывных распределений)

В этом случае у нас нет изначального предположения о параметре распределения θ , поэтому вместо него берется $\hat{\theta}$. Далее действовать нужно аналогично написанному ранее.

Критерий однородности Колмогорова-Смирнова

Для определения, принадлежат ли две выборки одному и тому же распределению, используется критерий однородности Колмогорова-Смирнова. Гипотеза H_0 : функции распределения двух выборок одинаковы.

Положим n, m — объемы двух сравниваемых выборок, \hat{F}_n, \hat{F}_m — эмпирические функции распределения, построенные по ним этих выборок, то статистика критерия имеет вид: $\lambda =$

$$= \sqrt{\frac{nm}{n+m}} \cdot \max |\hat{F}_n - \hat{F}_m|$$

Для уровня значимости α ищем λ_α — табличное критическое значение статистики. Критерий: если $\lambda \leq \lambda_\alpha$, гипотеза H_0 принимается.

Критерий применяется, если истинная функция распределения F_ξ непрерывна.

```
[1]: import os
import numpy as np
import pandas as pd
import tqdm
import matplotlib.pyplot as plt
from scipy import special, stats, optimize
from scipy.stats import pearsonr, mstats
from bisect import bisect_left, bisect_right
import functools
%matplotlib inline

[2]: plt.rcParams.update({'font.size': 16})
sample_count = [5, 10, 100, 10**3, 10**5]

[3]: def build_edf(emperical_data, return_y=False):
    length = len(emperical_data)
    if length == 0:
        raise ValueError("Emperical data must have length 1 or more")
    y_values = []
    prev = None
    counter, total = 0, 0
    for entry in emperical_data:
        if prev is None:
            prev = entry
        if entry == prev:
            counter += 1
```

```

    else:
        y_values += counter * [total / length]
        prev = entry
        counter = 1
    total += 1
    y_values += counter * [total / length]

    if return_y:
        return y_values

def edf(x, emperical_data=emperical_data, y_values=y_values):
    if x < emperical_data[0]:
        return 0.0
    return y_values[bisect_right(emperical_data, x) - 1]

return edf

```

```

[4]: def find_max_diff(data1, data2):
    i, j = 0, 0
    n, m = len(data1), len(data2)
    def find_next(i, j, n, m):
        new = -1
        if i < n and j < m:
            if data1[i] < data2[j]:
                new = data1[i]
                i += 1
            else:
                new = data2[j]
                j += 1
        elif i < n:
            new = data1[i]
            i += 1
        elif j < m:
            new = data2[j]
            j += 1
        while i < n and data1[i] == new:
            i += 1
        while j < m and data2[j] == new:
            j += 1
        return (i, j)
    ret = 0
    while (i, j) != (n, m):
        ret = max(ret, abs(i / n - j / m))
        i, j = find_next(i, j, n, m)
    return ret

```

Функция для генерации выборок

```
[5]: def generate_samples(function, **kwargs):  
    ret = {}  
    for size in sample_count:  
        ret[size] = [function(size=size, **kwargs) for i in range(5)]  
    return ret
```

```
[6]: chi2_quantiles = pd.read_csv("chi2_quantiles.csv", sep=';',  
    ↪index_col='count')
```

```
[7]: chi2_quantiles.head()
```

```
[7]:
```

	0.9	0.95	0.1	0.05
count				
1	0.015791	0.003932	2.705543	3.841459
2	0.210721	0.102587	4.605170	5.991465
3	0.584374	0.351846	6.251389	7.814728
4	1.063623	0.710723	7.779440	9.487729
5	1.610308	1.145476	9.236357	11.070498

```
[8]: def find_chi2(var_series, cdf, minimum, maximum, is_discrete=False):  
    n = len(var_series)  
    if is_discrete:  
        k = maximum - 1  
    else:  
        k = int(np.log2(n)) + 1  
    edges = np.linspace(minimum, maximum, k + 1)  
    p_arr, n_arr = [], []  
    for i in range(k):  
        p_arr.append(cdf(edges[i + 1]) - cdf(edges[i]))  
        n_arr.append(bisect_right(var_series, edges[i + 1]) -  
    ↪bisect_right(var_series, edges[i]))  
    p_arr, n_arr = np.array(p_arr), np.array(n_arr)  
    return {'k': k, 'chi2': n * np.sum((p_arr - n_arr / n)**2 /  
    ↪(p_arr))}
```

```
[9]: def find_dn(var_series, cdf):  
    ret = 0  
    n = len(var_series)  
    for i, entry in enumerate(var_series):  
        ret = max(ret, abs(cdf(entry) - (i + 1) / n), abs(cdf(entry) -  
    i / n))  
    return ret
```

```
[10]: lambdas = {'0.1': 1.22, '0.05': 1.36}
```

Распределение Ципфа (дискретное)

```
[11]: # функция вероятности
def zipf_pmf(x, s, N): return (1 / x**s) / np.sum(1 / np.arange(1,
↪N+1)**s)
# функция распределения
def zipf_cdf(x, s, N):
    if isinstance(x, int) or isinstance(x, float):
        return np.sum(zipf_pmf(np.arange(1, np.floor(x)+1), s, N))
    ret = np.zeros(x.shape)
    for i in range(x.shape[0]):
        ret[i] = np.sum(zipf_pmf(np.arange(1, np.floor(x[i])+1), s,
↪N))
    return ret
```

```
[12]: def zipf_random(s, N, size=1):
    uni_random = stats.uniform.rvs(size=size, )
    ticks = [0]
    ret = []
    for elem in uni_random:
        if elem <= ticks[-1]:
            ret.append(bisect_left(ticks, elem))
        else:
            cur = len(ticks)
            while ticks[-1] < elem:
                ticks.append(ticks[-1] + zipf_pmf(cur, s, N))
                cur += 1
            ret.append(cur - 1)
    return ret
```

Выберем для моделирования параметры $s = 1.5$ и $N = 10$

```
[13]: zipf_s = 1.5
zipf_N = 15
zipf_samples = generate_samples(zipf_random, s=zipf_s, N=zipf_N)
zipf_var_series = {}
for sample_size, samples in zipf_samples.items():
    zipf_var_series[sample_size] = [sorted(sample) for sample in
↪samples]
```

4.1 Проверка гипотез о виде распределения

Критерий согласия хи-квадрат

```
[14]: for sample_size, var_series in zipf_var_series.items():
    print("Выборки объема {}".format(sample_size))
    for i, sample in enumerate(var_series):
        print(" " + str(i + 1) + " ", end="")
```

```

    ret = find_chi2(sample, lambda x: zipf_cdf(x, zipf_s,
→zipf_N), 1, zipf_N, True)
    k, chi2 = ret['k'], ret['chi2']
    for alpha in ["0.1", "0.05"]:
        print("{:20}".format(" д.у.з. " + alpha + " гипотеза "
→), end="")
        from_table = chi2_quantiles.loc[k - 1][alpha]
        if from_table > chi2:
            print(" принимается ({:6.3f} > {:6.3f})".
→format(from_table, chi2), end="")
        else:
            print("не принимается ({:6.3f} <={:6.3f})".
→format(from_table, chi2), end="")
        print("")

```

Выборки объема 5

```

1: д.у.з. 0.1 г. + (19.812 > 11.366), д.у.з. 0.05 г. + (22.362 > 11.366)
2: д.у.з. 0.1 г. + (19.812 > 13.671), д.у.з. 0.05 г. + (22.362 > 13.671)
3: д.у.з. 0.1 г. + (19.812 > 3.182), д.у.з. 0.05 г. + (22.362 > 3.182)
4: д.у.з. 0.1 г. - (19.812 <=33.763), д.у.з. 0.05 г. - (22.362 <=33.763)
5: д.у.з. 0.1 г. + (19.812 > 9.181), д.у.з. 0.05 г. + (22.362 > 9.181)

```

Выборки объема 10

```

1: д.у.з. 0.1 г. - (19.812 <=21.446), д.у.з. 0.05 г. + (22.362 > 21.446)
2: д.у.з. 0.1 г. + (19.812 > 11.033), д.у.з. 0.05 г. + (22.362 > 11.033)
3: д.у.з. 0.1 г. - (19.812 <=23.666), д.у.з. 0.05 г. - (22.362 <=23.666)
4: д.у.з. 0.1 г. + (19.812 > 2.937), д.у.з. 0.05 г. + (22.362 > 2.937)
5: д.у.з. 0.1 г. + (19.812 > 3.699), д.у.з. 0.05 г. + (22.362 > 3.699)

```

Выборки объема 100

```

1: д.у.з. 0.1 г. + (19.812 > 5.786), д.у.з. 0.05 г. + (22.362 > 5.786)
2: д.у.з. 0.1 г. + (19.812 > 12.797), д.у.з. 0.05 г. + (22.362 > 12.797)
3: д.у.з. 0.1 г. + (19.812 > 11.169), д.у.з. 0.05 г. + (22.362 > 11.169)
4: д.у.з. 0.1 г. + (19.812 > 12.326), д.у.з. 0.05 г. + (22.362 > 12.326)
5: д.у.з. 0.1 г. - (19.812 <=19.923), д.у.з. 0.05 г. + (22.362 > 19.923)

```

Выборки объема 1000

```

1: д.у.з. 0.1 г. + (19.812 > 15.714), д.у.з. 0.05 г. + (22.362 > 15.714)
2: д.у.з. 0.1 г. + (19.812 > 19.256), д.у.з. 0.05 г. + (22.362 > 19.256)
3: д.у.з. 0.1 г. + (19.812 > 13.202), д.у.з. 0.05 г. + (22.362 > 13.202)
4: д.у.з. 0.1 г. + (19.812 > 13.799), д.у.з. 0.05 г. + (22.362 > 13.799)
5: д.у.з. 0.1 г. + (19.812 > 3.770), д.у.з. 0.05 г. + (22.362 > 3.770)

```

Выборки объема 100000

```

1: д.у.з. 0.1 г. - (19.812 <=20.544), д.у.з. 0.05 г. + (22.362 > 20.544)
2: д.у.з. 0.1 г. - (19.812 <=27.690), д.у.з. 0.05 г. - (22.362 <=27.690)
3: д.у.з. 0.1 г. + (19.812 > 10.681), д.у.з. 0.05 г. + (22.362 > 10.681)
4: д.у.з. 0.1 г. + (19.812 > 19.684), д.у.з. 0.05 г. + (22.362 > 19.684)
5: д.у.з. 0.1 г. + (19.812 > 19.552), д.у.з. 0.05 г. + (22.362 > 19.552)

```

Критерий согласия хи-квадрат для сложной гипотезы

```

[15]: # вклад выборки
def zipf_score(s, N, sample):
    ticks = np.arange(1, N+1)
    powered = ticks**(-s)

```

```

    return np.sum(np.sum(powered * np.log(ticks)) / np.sum(powered) -
    ↪ np.log(sample))

```

```

[16]: for sample_size, var_series in zipf_var_series.items():
    print("Выборки объема {}".format(sample_size))
    for i, sample in enumerate(var_series):
        print(" " + str(i + 1) + " ", end=" ")
        s_est = optimize.bisect(lambda s: zipf_score(s, zipf_N,
        ↪ sample), 0, 5)
        ret = find_chi2(sample, lambda x: zipf_cdf(x, s_est, zipf_N),
        ↪ 1, zipf_N, True)
        k, chi2 = ret['k'], ret['chi2']
        for alpha in ["0.1", "0.05"]:
            print("{:20}".format(" д.у.з. " + alpha + " гипотеза "),
            ↪ end=" ")

            from_table = chi2_quantiles.loc[k - 1][alpha]
            if from_table > chi2:
                print(" принимается ({:6.3f} > {:6.3f})".
                ↪ format(from_table, chi2), end=" ")
            else:
                print("не принимается ({:6.3f} <={:6.3f})".
                ↪ format(from_table, chi2), end=" ")
            print(" ОМП: s = {:.3f}".format(s_est))

```

Выборки объема 5

```

1: д.у.з. 0.1 г. + (19.812 > 9.897), д.у.з. 0.05 г. + (22.362 > 9.897), ОМП: s = 1.396
2: д.у.з. 0.1 г. + (19.812 > 13.517), д.у.з. 0.05 г. + (22.362 > 13.517), ОМП: s = 1.774
3: д.у.з. 0.1 г. + (19.812 > 4.379), д.у.з. 0.05 г. + (22.362 > 4.379), ОМП: s = 2.056
4: д.у.з. 0.1 г. + (19.812 > 10.047), д.у.з. 0.05 г. + (22.362 > 10.047), ОМП: s = 0.562
5: д.у.з. 0.1 г. + (19.812 > 9.929), д.у.з. 0.05 г. + (22.362 > 9.929), ОМП: s = 1.587

```

Выборки объема 10

```

1: д.у.з. 0.1 г. + (19.812 > 13.196), д.у.з. 0.05 г. + (22.362 > 13.196), ОМП: s = 1.198
2: д.у.з. 0.1 г. + (19.812 > 13.892), д.у.з. 0.05 г. + (22.362 > 13.892), ОМП: s = 1.797
3: д.у.з. 0.1 г. + (19.812 > 10.080), д.у.з. 0.05 г. + (22.362 > 10.080), ОМП: s = 0.768
4: д.у.з. 0.1 г. + (19.812 > 1.636), д.у.з. 0.05 г. + (22.362 > 1.636), ОМП: s = 2.882
5: д.у.з. 0.1 г. + (19.812 > 2.952), д.у.з. 0.05 г. + (22.362 > 2.952), ОМП: s = 2.316

```

Выборки объема 100

```

1: д.у.з. 0.1 г. + (19.812 > 6.287), д.у.з. 0.05 г. + (22.362 > 6.287), ОМП: s = 1.568
2: д.у.з. 0.1 г. + (19.812 > 11.221), д.у.з. 0.05 г. + (22.362 > 11.221), ОМП: s = 1.419
3: д.у.з. 0.1 г. + (19.812 > 8.264), д.у.з. 0.05 г. + (22.362 > 8.264), ОМП: s = 1.350
4: д.у.з. 0.1 г. + (19.812 > 11.826), д.у.з. 0.05 г. + (22.362 > 11.826), ОМП: s = 1.439
5: д.у.з. 0.1 г. - (19.812 <=23.130), д.у.з. 0.05 г. - (22.362 <=23.130), ОМП: s = 1.617

```

Выборки объема 1000

```

1: д.у.з. 0.1 г. + (19.812 > 13.768), д.у.з. 0.05 г. + (22.362 > 13.768), ОМП: s = 1.458
2: д.у.з. 0.1 г. + (19.812 > 18.609), д.у.з. 0.05 г. + (22.362 > 18.609), ОМП: s = 1.487
3: д.у.з. 0.1 г. + (19.812 > 12.804), д.у.з. 0.05 г. + (22.362 > 12.804), ОМП: s = 1.493
4: д.у.з. 0.1 г. + (19.812 > 14.325), д.у.з. 0.05 г. + (22.362 > 14.325), ОМП: s = 1.527
5: д.у.з. 0.1 г. + (19.812 > 3.779), д.у.з. 0.05 г. + (22.362 > 3.779), ОМП: s = 1.509

```

Выборки объема 100000

```

1: д.у.з. 0.1 г. - (19.812 <=19.814), д.у.з. 0.05 г. + (22.362 > 19.814), ОМП: s = 1.496
2: д.у.з. 0.1 г. - (19.812 <=27.362), д.у.з. 0.05 г. - (22.362 <=27.362), ОМП: s = 1.502
3: д.у.з. 0.1 г. + (19.812 > 10.696), д.у.з. 0.05 г. + (22.362 > 10.696), ОМП: s = 1.500
4: д.у.з. 0.1 г. + (19.812 > 18.287), д.у.з. 0.05 г. + (22.362 > 18.287), ОМП: s = 1.492
5: д.у.з. 0.1 г. + (19.812 > 19.522), д.у.з. 0.05 г. + (22.362 > 19.522), ОМП: s = 1.498

```

Бета-распределение (абсолютно непрерывное)

```
[17]: # плотность вероятности
def beta_pdf(x, alpha, beta): return stats.beta.pdf(x, alpha, beta)
# функция распределения
def beta_cdf(x, alpha, beta): return special.betainc(alpha, beta, x)

[18]: def beta_random(a, b, size=1):
    ret = []
    l = a
    k = a + b - 1
    for _ in range(size):
        uni_set = stats.uniform.rvs(size=k)
        uni_set.sort()
        ret.append(uni_set[l-1])
    return np.array(ret)
```

Выберем для моделирования параметры $a = 5$ и $b = 2$

```
[19]: beta_a = 5
beta_b = 2
beta_samples = generate_samples(beta_random, a=beta_a, b=beta_b)
beta_var_series = {}
for sample_size, samples in beta_samples.items():
    beta_var_series[sample_size] = [sorted(sample) for sample in
    ↪ samples]
```

4.1 Проверка гипотез о виде распределения

Критерий согласия хи-квадрат

```
[20]: for sample_size, var_series in beta_var_series.items():
    print("Выборки объема {}".format(sample_size))
    for i, sample in enumerate(var_series):
        print(" " + str(i + 1) + " ", end=" ")
        ret = find_chi2(sample, lambda x: beta_cdf(x, beta_a,
    ↪ beta_b), 0, 1, False)
        k, chi2 = ret['k'], ret['chi2']
        for alpha in ["0.1", "0.05"]:
            print("{:20}".format(" д.у.з. " + alpha + " гипотеза "),
    ↪ end=" ")
            from_table = chi2_quantiles.loc[k - 1][alpha]
            if from_table > chi2:
                print(" принимается ({:6.3f} > {:6.3f})".
    ↪ format(from_table, chi2), end=" ")
            else:
                print("не принимается ({:6.3f} <={:6.3f})".
    ↪ format(from_table, chi2), end=" ")
            print(" ")
```


Выборки объема 5

1: д.у.з. 0.1 г. + (4.605 > 1.633), д.у.з. 0.05 г. + (5.991 > 1.633)
2: д.у.з. 0.1 г. + (4.605 > 2.706), д.у.з. 0.05 г. + (5.991 > 2.706)
3: д.у.з. 0.1 г. + (4.605 > 1.633), д.у.з. 0.05 г. + (5.991 > 1.633)
4: д.у.з. 0.1 г. + (4.605 > 0.532), д.у.з. 0.05 г. + (5.991 > 0.532)
5: д.у.з. 0.1 г. + (4.605 > 2.706), д.у.з. 0.05 г. + (5.991 > 2.706)

Выборки объема 10

1: д.у.з. 0.1 г. + (6.251 > 0.799), д.у.з. 0.05 г. + (7.815 > 0.799)
2: д.у.з. 0.1 г. + (6.251 > 0.799), д.у.з. 0.05 г. + (7.815 > 0.799)
3: д.у.з. 0.1 г. + (6.251 > 1.912), д.у.з. 0.05 г. + (7.815 > 1.912)
4: д.у.з. 0.1 г. + (6.251 > 0.276), д.у.з. 0.05 г. + (7.815 > 0.276)
5: д.у.з. 0.1 г. + (6.251 > 2.633), д.у.з. 0.05 г. + (7.815 > 2.633)

Выборки объема 100

1: д.у.з. 0.1 г. - (10.645 <=33.303), д.у.з. 0.05 г. - (12.592 <=33.303)
2: д.у.з. 0.1 г. + (10.645 > 3.614), д.у.з. 0.05 г. + (12.592 > 3.614)
3: д.у.з. 0.1 г. + (10.645 > 2.945), д.у.з. 0.05 г. + (12.592 > 2.945)
4: д.у.з. 0.1 г. + (10.645 > 7.780), д.у.з. 0.05 г. + (12.592 > 7.780)
5: д.у.з. 0.1 г. + (10.645 > 3.344), д.у.з. 0.05 г. + (12.592 > 3.344)

Выборки объема 1000

1: д.у.з. 0.1 г. + (14.684 > 7.806), д.у.з. 0.05 г. + (16.919 > 7.806)
2: д.у.з. 0.1 г. + (14.684 > 3.937), д.у.з. 0.05 г. + (16.919 > 3.937)
3: д.у.з. 0.1 г. + (14.684 > 6.159), д.у.з. 0.05 г. + (16.919 > 6.159)
4: д.у.з. 0.1 г. + (14.684 > 4.398), д.у.з. 0.05 г. + (16.919 > 4.398)
5: д.у.з. 0.1 г. + (14.684 > 2.647), д.у.з. 0.05 г. + (16.919 > 2.647)

Выборки объема 100000

1: д.у.з. 0.1 г. + (23.542 > 11.631), д.у.з. 0.05 г. + (26.296 > 11.631)
2: д.у.з. 0.1 г. + (23.542 > 12.441), д.у.з. 0.05 г. + (26.296 > 12.441)
3: д.у.з. 0.1 г. - (23.542 <=29.362), д.у.з. 0.05 г. - (26.296 <=29.362)
4: д.у.з. 0.1 г. + (23.542 > 10.552), д.у.з. 0.05 г. + (26.296 > 10.552)
5: д.у.з. 0.1 г. + (23.542 > 14.352), д.у.з. 0.05 г. + (26.296 > 14.352)

Критерий согласия хи-квадрат для сложной гипотезы

```
[21]: # вклад выборки
def beta_score(params, sample):
    sample = np.array(sample)
    a, b = params[0], params[1]
    n = len(sample)
    first = n * (special.psi(a+b)-special.psi(a)) + np.sum(np.
    ↪log(sample))
    second = n * (special.psi(a+b)-special.psi(b)) + np.sum(np.log(1 -
    sample))
    return (first, second)

# явно посчитанные частные производные 1 и 2 порядка
# (для более хорошей работы метода Ньютона)
def beta_score_fprime(params, n):
    a, b = params[0], params[1]
    first = n * (special.polygamma(1, a+b)-special.polygamma(1, a))
    second = n * (special.polygamma(1, a+b)-special.polygamma(1, b))
    return (first, second)

def beta_score_fprime2(params, n):
    a, b = params[0], params[1]
```

```

first = n * (special.polygamma(2, a+b)-special.polygamma(2, a))
second = n * (special.polygamma(2, a+b)-special.polygamma(2, b))
return (first, second)

```

```

[22]: for sample_size, var_series in beta_var_series.items():
    print("Выборки объема {}".format(sample_size))
    for i, sample in enumerate(var_series):
        print(" " + str(i + 1) + " ", end=" ")
        params_est = optimize.newton(lambda params:
        ↪beta_score(params, sample), (4.9, 2.1), fprime=lambda params:
        ↪beta_score_fprime(params, len(sample)),
                                fprime2=lambda params:
        ↪beta_score_fprime2(params, len(sample)), maxiter=1000)
        ret = find_chi2(sample, lambda x: beta_cdf(x, params_est[0],
        ↪params_est[1]), 0, 1, False)
        k, chi2 = ret['k'], ret['chi2']
        for alpha in ["0.1", "0.05"]:
            print("{:20}".format(" д.у.з. " + alpha + " гипотеза "),
            ↪end=" ")
            if from_table > chi2:
                print(" принимается ({:6.3f} > {:6.3f})".
            ↪format(from_table, chi2), end=" ")
            else:
                print("не принимается ({:6.3f} <={:6.3f})".
            ↪format(from_table, chi2), end=" ")
            print(" ОМП: a = {:.3f}, b = {:.3f}".format(*params_est))

```

Выборки объема 5

```

1: д.у.з. 0.1 г. + (26.296 > 0.564), д.у.з. 0.05 г. + (26.296 > 0.564), ОМП: a = 5.055, b = 2.577
2: д.у.з. 0.1 г. + (26.296 > 0.163), д.у.з. 0.05 г. + (26.296 > 0.163), ОМП: a = 16.154, b = 3.011
3: д.у.з. 0.1 г. + (26.296 > 0.192), д.у.з. 0.05 г. + (26.296 > 0.192), ОМП: a = 7.301, b = 4.788
4: д.у.з. 0.1 г. + (26.296 > 0.006), д.у.з. 0.05 г. + (26.296 > 0.006), ОМП: a = 15.514, b = 5.465
5: д.у.з. 0.1 г. + (26.296 > 0.147), д.у.з. 0.05 г. + (26.296 > 0.147), ОМП: a = 13.067, b = 2.052

```

Выборки объема 10

```

1: д.у.з. 0.1 г. + (26.296 > 1.155), д.у.з. 0.05 г. + (26.296 > 1.155), ОМП: a = 7.479, b = 2.805
2: д.у.з. 0.1 г. + (26.296 > 1.105), д.у.з. 0.05 г. + (26.296 > 1.105), ОМП: a = 5.566, b = 1.401
3: д.у.з. 0.1 г. + (26.296 > 2.098), д.у.з. 0.05 г. + (26.296 > 2.098), ОМП: a = 4.619, b = 1.881
4: д.у.з. 0.1 г. + (26.296 > 0.306), д.у.з. 0.05 г. + (26.296 > 0.306), ОМП: a = 4.741, b = 1.924
5: д.у.з. 0.1 г. + (26.296 > 0.301), д.у.з. 0.05 г. + (26.296 > 0.301), ОМП: a = 9.684, b = 2.769

```

Выборки объема 100

```

1: д.у.з. 0.1 г. + (26.296 > 9.692), д.у.з. 0.05 г. + (26.296 > 9.692), ОМП: a = 4.485, b = 2.178
2: д.у.з. 0.1 г. + (26.296 > 2.434), д.у.з. 0.05 г. + (26.296 > 2.434), ОМП: a = 5.914, b = 2.479
3: д.у.з. 0.1 г. + (26.296 > 2.164), д.у.з. 0.05 г. + (26.296 > 2.164), ОМП: a = 5.607, b = 2.250
4: д.у.з. 0.1 г. + (26.296 > 7.078), д.у.з. 0.05 г. + (26.296 > 7.078), ОМП: a = 5.601, b = 1.865
5: д.у.з. 0.1 г. + (26.296 > 3.567), д.у.з. 0.05 г. + (26.296 > 3.567), ОМП: a = 4.349, b = 1.631

```

Выборки объема 1000

```

1: д.у.з. 0.1 г. + (26.296 > 8.366), д.у.з. 0.05 г. + (26.296 > 8.366), ОМП: a = 4.971, b = 1.975
2: д.у.з. 0.1 г. + (26.296 > 3.905), д.у.з. 0.05 г. + (26.296 > 3.905), ОМП: a = 4.903, b = 1.964
3: д.у.з. 0.1 г. + (26.296 > 2.845), д.у.з. 0.05 г. + (26.296 > 2.845), ОМП: a = 5.162, b = 2.139
4: д.у.з. 0.1 г. + (26.296 > 2.841), д.у.з. 0.05 г. + (26.296 > 2.841), ОМП: a = 5.243, b = 2.084
5: д.у.з. 0.1 г. + (26.296 > 2.722), д.у.з. 0.05 г. + (26.296 > 2.722), ОМП: a = 5.103, b = 2.049

```

Выборки объема 100000

```

1: д.у.з. 0.1 г. + (26.296 > 11.827), д.у.з. 0.05 г. + (26.296 > 11.827), ОМП: a = 5.005, b = 2.002
2: д.у.з. 0.1 г. + (26.296 > 10.886), д.у.з. 0.05 г. + (26.296 > 10.886), ОМП: a = 5.017, b = 2.007
3: д.у.з. 0.1 г. - (26.296 <=28.626), д.у.з. 0.05 г. - (26.296 <=28.626), ОМП: a = 5.019, b = 2.004
4: д.у.з. 0.1 г. + (26.296 > 9.628), д.у.з. 0.05 г. + (26.296 > 9.628), ОМП: a = 5.003, b = 1.998
5: д.у.з. 0.1 г. + (26.296 > 14.273), д.у.з. 0.05 г. + (26.296 > 14.273), ОМП: a = 4.981, b = 1.994

```

Критерий согласия Колмогорова-Смирнова

```
[23]: for sample_size, var_series in beta_var_series.items():
    sqrt_n = np.sqrt(sample_size)
    print("Выборки объема {}".format(sample_size))
    for i, sample in enumerate(var_series):
        print(" " + str(i + 1) + " ", end=" ")
        dn = find_dn(sample, lambda x: beta_cdf(x, beta_a, beta_b))
        for alpha in ["0.1", "0.05"]:
            print("{:20}".format(" д.у.з. " + alpha + " гипотеза "),
→end=" ")
            from_table = chi2_quantiles.loc[k - 1][alpha]
            if dn * sqrt_n <= lambdas[alpha]:
                print(" принимается ({:5.3f} <={:5.3f})".format(dn
→* sqrt_n, lambdas[alpha]), end=" ")
            else:
                print("не принимается ({:5.3f} > {:5.3f})".format(dn
→* sqrt_n, lambdas[alpha]), end=" ")
            print(" ")
```

Выборки объема 5

```
1: д.у.з. 0.1 г. + (0.739 <=1.220), д.у.з. 0.05 г. + (0.739 <=1.360)
2: д.у.з. 0.1 г. + (1.090 <=1.220), д.у.з. 0.05 г. + (1.090 <=1.360)
3: д.у.з. 0.1 г. + (0.915 <=1.220), д.у.з. 0.05 г. + (0.915 <=1.360)
4: д.у.з. 0.1 г. + (0.538 <=1.220), д.у.з. 0.05 г. + (0.538 <=1.360)
5: д.у.з. 0.1 г. - (1.278 > 1.220), д.у.з. 0.05 г. + (1.278 <=1.360)
```

Выборки объема 10

```
1: д.у.з. 0.1 г. + (0.548 <=1.220), д.у.з. 0.05 г. + (0.548 <=1.360)
2: д.у.з. 0.1 г. - (1.243 > 1.220), д.у.з. 0.05 г. + (1.243 <=1.360)
3: д.у.з. 0.1 г. + (0.638 <=1.220), д.у.з. 0.05 г. + (0.638 <=1.360)
4: д.у.з. 0.1 г. + (0.596 <=1.220), д.у.з. 0.05 г. + (0.596 <=1.360)
5: д.у.з. 0.1 г. + (0.933 <=1.220), д.у.з. 0.05 г. + (0.933 <=1.360)
```

Выборки объема 100

```
1: д.у.з. 0.1 г. - (1.255 > 1.220), д.у.з. 0.05 г. + (1.255 <=1.360)
2: д.у.з. 0.1 г. + (0.723 <=1.220), д.у.з. 0.05 г. + (0.723 <=1.360)
3: д.у.з. 0.1 г. + (0.739 <=1.220), д.у.з. 0.05 г. + (0.739 <=1.360)
4: д.у.з. 0.1 г. - (1.430 > 1.220), д.у.з. 0.05 г. - (1.430 > 1.360)
5: д.у.з. 0.1 г. + (0.469 <=1.220), д.у.з. 0.05 г. + (0.469 <=1.360)
```

Выборки объема 1000

```
1: д.у.з. 0.1 г. + (0.654 <=1.220), д.у.з. 0.05 г. + (0.654 <=1.360)
2: д.у.з. 0.1 г. + (0.682 <=1.220), д.у.з. 0.05 г. + (0.682 <=1.360)
3: д.у.з. 0.1 г. + (0.992 <=1.220), д.у.з. 0.05 г. + (0.992 <=1.360)
4: д.у.з. 0.1 г. + (0.558 <=1.220), д.у.з. 0.05 г. + (0.558 <=1.360)
5: д.у.з. 0.1 г. + (0.709 <=1.220), д.у.з. 0.05 г. + (0.709 <=1.360)
```

Выборки объема 100000

```
1: д.у.з. 0.1 г. + (0.717 <=1.220), д.у.з. 0.05 г. + (0.717 <=1.360)
2: д.у.з. 0.1 г. + (0.713 <=1.220), д.у.з. 0.05 г. + (0.713 <=1.360)
3: д.у.з. 0.1 г. + (0.932 <=1.220), д.у.з. 0.05 г. + (0.932 <=1.360)
4: д.у.з. 0.1 г. + (0.752 <=1.220), д.у.з. 0.05 г. + (0.752 <=1.360)
5: д.у.з. 0.1 г. + (0.753 <=1.220), д.у.з. 0.05 г. + (0.753 <=1.360)
```

Критерий согласия Колмогорова-Смирнова для сложной гипотезы

```
[24]: for sample_size, var_series in beta_var_series.items():
    sqrt_n = np.sqrt(sample_size)
    print("Выборки объема {}".format(sample_size))
    for i, sample in enumerate(var_series):
        print(" " + str(i + 1) + " ", end=" ")
        params_est = optimize.newton(lambda params:
        ↪beta_score(params, sample), (4.9, 2.1), fprime=lambda params:
        ↪beta_score_fprime(params, len(sample)),
                                fprime2=lambda params:
        ↪beta_score_fprime2(params, len(sample)), maxiter=1000)
        dn = find_dn(sample, lambda x: beta_cdf(x, params_est[0],
        ↪params_est[1]))
        for alpha in ["0.1", "0.05"]:
            print("{:20}".format(" д.у.з. " + alpha + " гипотеза "),
            ↪end=" ")
            if dn * sqrt_n <= lambdas[alpha]:
                print(" принимается ({:5.3f} <={:5.3f})".format(dn
            ↪* sqrt_n, lambdas[alpha]), end=" ")
            else:
                print("не принимается ({:5.3f} > {:5.3f})".format(dn
            ↪* sqrt_n, lambdas[alpha]), end=" ")
            print(" ")
```

Выборки объема 5

```
1: д.у.з. 0.1 г. + (0.569 <=1.220), д.у.з. 0.05 г. + (0.569 <=1.360)
2: д.у.з. 0.1 г. + (0.522 <=1.220), д.у.з. 0.05 г. + (0.522 <=1.360)
3: д.у.з. 0.1 г. + (0.525 <=1.220), д.у.з. 0.05 г. + (0.525 <=1.360)
4: д.у.з. 0.1 г. + (0.337 <=1.220), д.у.з. 0.05 г. + (0.337 <=1.360)
5: д.у.з. 0.1 г. + (0.723 <=1.220), д.у.з. 0.05 г. + (0.723 <=1.360)
```

Выборки объема 10

```
1: д.у.з. 0.1 г. + (0.468 <=1.220), д.у.з. 0.05 г. + (0.468 <=1.360)
2: д.у.з. 0.1 г. + (0.578 <=1.220), д.у.з. 0.05 г. + (0.578 <=1.360)
3: д.у.з. 0.1 г. + (0.601 <=1.220), д.у.з. 0.05 г. + (0.601 <=1.360)
4: д.у.з. 0.1 г. + (0.620 <=1.220), д.у.з. 0.05 г. + (0.620 <=1.360)
5: д.у.з. 0.1 г. + (0.466 <=1.220), д.у.з. 0.05 г. + (0.466 <=1.360)
```

Выборки объема 100

```
1: д.у.з. 0.1 г. + (0.616 <=1.220), д.у.з. 0.05 г. + (0.616 <=1.360)
2: д.у.з. 0.1 г. + (0.617 <=1.220), д.у.з. 0.05 г. + (0.617 <=1.360)
3: д.у.з. 0.1 г. + (0.652 <=1.220), д.у.з. 0.05 г. + (0.652 <=1.360)
4: д.у.з. 0.1 г. - (1.253 > 1.220), д.у.з. 0.05 г. + (1.253 <=1.360)
5: д.у.з. 0.1 г. + (0.811 <=1.220), д.у.з. 0.05 г. + (0.811 <=1.360)
```

Выборки объема 1000

```
1: д.у.з. 0.1 г. + (0.718 <=1.220), д.у.з. 0.05 г. + (0.718 <=1.360)
2: д.у.з. 0.1 г. + (0.663 <=1.220), д.у.з. 0.05 г. + (0.663 <=1.360)
3: д.у.з. 0.1 г. + (0.514 <=1.220), д.у.з. 0.05 г. + (0.514 <=1.360)
4: д.у.з. 0.1 г. + (0.359 <=1.220), д.у.з. 0.05 г. + (0.359 <=1.360)
5: д.у.з. 0.1 г. + (0.680 <=1.220), д.у.з. 0.05 г. + (0.680 <=1.360)
```

Выборки объема 100000

```
1: д.у.з. 0.1 г. + (0.735 <=1.220), д.у.з. 0.05 г. + (0.735 <=1.360)
2: д.у.з. 0.1 г. + (0.533 <=1.220), д.у.з. 0.05 г. + (0.533 <=1.360)
```

3: д.у.з. 0.1 г. + (0.662 <=1.220), д.у.з. 0.05 г. + (0.662 <=1.360)
 4: д.у.з. 0.1 г. + (0.469 <=1.220), д.у.з. 0.05 г. + (0.469 <=1.360)
 5: д.у.з. 0.1 г. + (0.684 <=1.220), д.у.з. 0.05 г. + (0.684 <=1.360)

```
[25]: def lambda_smirnov(n, m, max_diff):
    return np.sqrt(n * m / (n + m)) * max_diff
for size_i, size in enumerate(sample_count):
    for i in range(len(beta_var_series[size])):
        for j in range(i + 1, len(beta_var_series[size])):
            sm_stat = lambda_smirnov(size, size,
            ↪find_max_diff(beta_var_series[size][i], beta_var_series[size][j]))
            print("Выборки размером {:6} и {:6}".format(size, size),
            ↪end="")

            for alpha in ["0.1", "0.05"]:
                print(" д.у.з. " + alpha + " гипотеза ", end="")
                if sm_stat <= lambdas[alpha]:
                    print(" принимается ({:.3f} <={:.3f})".
                    ↪format(sm_stat, lambdas[alpha]), end="")
                else:
                    print("не принимается ({:.3f} > {:.3f})".
                    ↪format(sm_stat, lambdas[alpha]), end="")
                print("", end="")
            print("")
            for size_j in range(size_i + 1, len(sample_count)):
                size2 = sample_count[size_j]
                for sample in beta_var_series[size2]:
                    sm_stat = lambda_smirnov(size, size2,
                    ↪find_max_diff(beta_var_series[size][i], sample))
                    print("Выборки размером {:6} и {:6}".format(size,
                    ↪size2), end="")

                    for alpha in ["0.1", "0.05"]:
                        print(" д.у.з. " + alpha + " гипотеза ", end="")
                        if sm_stat <= lambdas[alpha]:
                            print(" принимается ({:.3f} <={:.3f})".
                            ↪format(sm_stat, lambdas[alpha]), end="")
                        else:
                            print("не принимается ({:.3f} > {:.3f})".
                            ↪format(sm_stat, lambdas[alpha]), end="")
                        print("", end="")
                    print("")
```

В. р. 5 и 5: д.у.з. 0.1 г. + (0.949 <=1.220), д.у.з. 0.05 г. + (0.949 <=1.360)
 В. р. 5 и 5: д.у.з. 0.1 г. + (0.632 <=1.220), д.у.з. 0.05 г. + (0.632 <=1.360)
 В. р. 5 и 5: д.у.з. 0.1 г. + (0.632 <=1.220), д.у.з. 0.05 г. + (0.632 <=1.360)
 В. р. 5 и 5: д.у.з. 0.1 г. + (0.949 <=1.220), д.у.з. 0.05 г. + (0.949 <=1.360)
 В. р. 5 и 10: д.у.з. 0.1 г. + (0.730 <=1.220), д.у.з. 0.05 г. + (0.730 <=1.360)
 В. р. 5 и 10: д.у.з. 0.1 г. + (1.095 <=1.220), д.у.з. 0.05 г. + (1.095 <=1.360)
 В. р. 5 и 10: д.у.з. 0.1 г. + (0.548 <=1.220), д.у.з. 0.05 г. + (0.548 <=1.360)
 В. р. 5 и 10: д.у.з. 0.1 г. + (0.730 <=1.220), д.у.з. 0.05 г. + (0.730 <=1.360)

V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.913	<=1.220),	д.у.з.	0.05 г.	+	(0.913	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(0.546	<=1.220),	д.у.з.	0.05 г.	+	(0.546	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(0.676	<=1.220),	д.у.з.	0.05 г.	+	(0.676	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(0.676	<=1.220),	д.у.з.	0.05 г.	+	(0.676	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(1.004	<=1.220),	д.у.з.	0.05 г.	+	(1.004	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(0.764	<=1.220),	д.у.з.	0.05 г.	+	(0.764	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.734	<=1.220),	д.у.з.	0.05 г.	+	(0.734	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.745	<=1.220),	д.у.з.	0.05 г.	+	(0.745	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.705	<=1.220),	д.у.з.	0.05 г.	+	(0.705	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.754	<=1.220),	д.у.з.	0.05 г.	+	(0.754	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.734	<=1.220),	д.у.з.	0.05 г.	+	(0.734	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.744	<=1.220),	д.у.з.	0.05 г.	+	(0.744	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.737	<=1.220),	д.у.з.	0.05 г.	+	(0.737	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.742	<=1.220),	д.у.з.	0.05 г.	+	(0.742	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.741	<=1.220),	д.у.з.	0.05 г.	+	(0.741	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.738	<=1.220),	д.у.з.	0.05 г.	+	(0.738	<=1.360)
V. p.	5 и	5:	д.у.з.	0.1 г.	-	(1.265	> 1.220),	д.у.з.	0.05 г.	+	(1.265	<=1.360)
V. p.	5 и	5:	д.у.з.	0.1 г.	+	(0.949	<=1.220),	д.у.з.	0.05 г.	+	(0.949	<=1.360)
V. p.	5 и	5:	д.у.з.	0.1 г.	+	(0.632	<=1.220),	д.у.з.	0.05 г.	+	(0.632	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.913	<=1.220),	д.у.з.	0.05 г.	+	(0.913	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.365	<=1.220),	д.у.з.	0.05 г.	+	(0.365	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.913	<=1.220),	д.у.з.	0.05 г.	+	(0.913	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.913	<=1.220),	д.у.з.	0.05 г.	+	(0.913	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.730	<=1.220),	д.у.з.	0.05 г.	+	(0.730	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	-	(1.266	> 1.220),	д.у.з.	0.05 г.	+	(1.266	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(1.178	<=1.220),	д.у.з.	0.05 г.	+	(1.178	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(1.113	<=1.220),	д.у.з.	0.05 г.	+	(1.113	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(1.047	<=1.220),	д.у.з.	0.05 г.	+	(1.047	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(0.982	<=1.220),	д.у.з.	0.05 г.	+	(0.982	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(1.117	<=1.220),	д.у.з.	0.05 г.	+	(1.117	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(1.077	<=1.220),	д.у.з.	0.05 г.	+	(1.077	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(1.131	<=1.220),	д.у.з.	0.05 г.	+	(1.131	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(1.097	<=1.220),	д.у.з.	0.05 г.	+	(1.097	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(1.104						

V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.504	<=1.220),	д.у.з.	0.05 г.	+	(0.504	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.520	<=1.220),	д.у.з.	0.05 г.	+	(0.520	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.575	<=1.220),	д.у.з.	0.05 г.	+	(0.575	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.544	<=1.220),	д.у.з.	0.05 г.	+	(0.544	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	+	(0.560	<=1.220),	д.у.з.	0.05 г.	+	(0.560	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.538	<=1.220),	д.у.з.	0.05 г.	+	(0.538	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.538	<=1.220),	д.у.з.	0.05 г.	+	(0.538	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.533	<=1.220),	д.у.з.	0.05 г.	+	(0.533	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.533	<=1.220),	д.у.з.	0.05 г.	+	(0.533	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	+	(0.542	<=1.220),	д.у.з.	0.05 г.	+	(0.542	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(1.095	<=1.220),	д.у.з.	0.05 г.	+	(1.095	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.365	<=1.220),	д.у.з.	0.05 г.	+	(0.365	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(1.095	<=1.220),	д.у.з.	0.05 г.	+	(1.095	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(1.095	<=1.220),	д.у.з.	0.05 г.	+	(1.095	<=1.360)
V. p.	5 и	10:	д.у.з.	0.1 г.	+	(0.913	<=1.220),	д.у.з.	0.05 г.	+	(0.913	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	-	(1.418	> 1.220),	д.у.з.	0.05 г.	-	(1.418	> 1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	-	(1.397	> 1.220),	д.у.з.	0.05 г.	-	(1.397	> 1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	-	(1.309	> 1.220),	д.у.з.	0.05 г.	+	(1.309	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	-	(1.266	> 1.220),	д.у.з.	0.05 г.	+	(1.266	<=1.360)
V. p.	5 и	100:	д.у.з.	0.1 г.	+	(1.200	<=1.220),	д.у.з.	0.05 г.	+	(1.200	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	-	(1.300	> 1.220),	д.у.з.	0.05 г.	+	(1.300	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	-	(1.283	> 1.220),	д.у.з.	0.05 г.	+	(1.283	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	-	(1.336	> 1.220),	д.у.з.	0.05 г.	+	(1.336	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	-	(1.305	> 1.220),	д.у.з.	0.05 г.	+	(1.305	<=1.360)
V. p.	5 и	1000:	д.у.з.	0.1 г.	-	(1.285	> 1.220),	д.у.з.	0.05 г.	+	(1.285	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	-	(1.281	> 1.220),	д.у.з.	0.05 г.	+	(1.281	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	-	(1.281	> 1.220),	д.у.з.	0.05 г.	+	(1.281	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	-	(1.277	> 1.220),	д.у.з.	0.05 г.	+	(1.277	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	-	(1.277	> 1.220),	д.у.з.	0.05 г.	+	(1.277	<=1.360)
V. p.	5 и	100000:	д.у.з.	0.1 г.	-	(1.278	> 1.220),	д.у.з.	0.05 г.	+	(1.278	<=1.360)
V. p.	10 и	10:	д.у.з.	0.1 г.	+	(0.894	<=1.220),	д.у.з.	0.05 г.	+	(0.894	<=1.360)
V. p.	10 и	10:	д.у.з.	0.1 г.	+	(0.671	<=1.220),	д.у.з.	0.05 г.	+	(0.671	<=1.360)
V. p.	10 и	10:	д.у.з.	0.1 г.	+	(0.447	<=1.220),	д.у.з.	0.05 г.	+	(0.447	<=1.360)
V. p.	10 и	10:	д.у.з.									

V. p.	10	и	100:	д.у.з.	0.1	г.	+	(1.025	<=1.220),	д.у.з.	0.05	г.	+	(1.025	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.	+	(0.603	<=1.220),	д.у.з.	0.05	г.	+	(0.603	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.648	<=1.220),	д.у.з.	0.05	г.	+	(0.648	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.636	<=1.220),	д.у.з.	0.05	г.	+	(0.636	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.557	<=1.220),	д.у.з.	0.05	г.	+	(0.557	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.661	<=1.220),	д.у.з.	0.05	г.	+	(0.661	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.620	<=1.220),	д.у.з.	0.05	г.	+	(0.620	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.643	<=1.220),	д.у.з.	0.05	г.	+	(0.643	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.638	<=1.220),	д.у.з.	0.05	г.	+	(0.638	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.641	<=1.220),	д.у.з.	0.05	г.	+	(0.641	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.639	<=1.220),	д.у.з.	0.05	г.	+	(0.639	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.633	<=1.220),	д.у.з.	0.05	г.	+	(0.633	<=1.360)
V. p.	10	и	10:	д.у.з.	0.1	г.	+	(0.671	<=1.220),	д.у.з.	0.05	г.	+	(0.671	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.	+	(0.874	<=1.220),	д.у.з.	0.05	г.	+	(0.874	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.	+	(0.633	<=1.220),	д.у.з.	0.05	г.	+	(0.633	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.	+	(0.663	<=1.220),	д.у.з.	0.05	г.	+	(0.663	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.	+	(0.362	<=1.220),	д.у.з.	0.05	г.	+	(0.362	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.	+	(0.573	<=1.220),	д.у.з.	0.05	г.	+	(0.573	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.570	<=1.220),	д.у.з.	0.05	г.	+	(0.570	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.604	<=1.220),	д.у.з.	0.05	г.	+	(0.604	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.664	<=1.220),	д.у.з.	0.05	г.	+	(0.664	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.573	<=1.220),	д.у.з.	0.05	г.	+	(0.573	<=1.360)
V. p.	10	и	1000:	д.у.з.	0.1	г.	+	(0.632	<=1.220),	д.у.з.	0.05	г.	+	(0.632	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.595	<=1.220),	д.у.з.	0.05	г.	+	(0.595	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.596	<=1.220),	д.у.з.	0.05	г.	+	(0.596	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.588	<=1.220),	д.у.з.	0.05	г.	+	(0.588	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.593	<=1.220),	д.у.з.	0.05	г.	+	(0.593	<=1.360)
V. p.	10	и	100000:	д.у.з.	0.1	г.	+	(0.602	<=1.220),	д.у.з.	0.05	г.	+	(0.602	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.	+	(1.146	<=1.220),	д.у.з.	0.05	г.	+	(1.146	<=1.360)
V. p.	10	и	100:	д.у.з.	0.1	г.									

V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.868	<=1.220),	д.у.з.	0.05	г.	+	(0.868	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.877	<=1.220),	д.у.з.	0.05	г.	+	(0.877	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.610	<=1.220),	д.у.з.	0.05	г.	+	(0.610	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.763	<=1.220),	д.у.з.	0.05	г.	+	(0.763	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.696	<=1.220),	д.у.з.	0.05	г.	+	(0.696	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.739	<=1.220),	д.у.з.	0.05	г.	+	(0.739	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.735	<=1.220),	д.у.з.	0.05	г.	+	(0.735	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.741	<=1.220),	д.у.з.	0.05	г.	+	(0.741	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.756	<=1.220),	д.у.з.	0.05	г.	+	(0.756	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.727	<=1.220),	д.у.з.	0.05	г.	+	(0.727	<=1.360)
V. p.	100	и	100:	д.у.з.	0.1	г.	+	(1.131	<=1.220),	д.у.з.	0.05	г.	+	(1.131	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	-	(1.325	> 1.220),	д.у.з.	0.05	г.	+	(1.325	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	-	(1.373	> 1.220),	д.у.з.	0.05	г.	-	(1.373	> 1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	-	(1.611	> 1.220),	д.у.з.	0.05	г.	-	(1.611	> 1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	-	(1.335	> 1.220),	д.у.з.	0.05	г.	+	(1.335	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	-	(1.468	> 1.220),	д.у.з.	0.05	г.	-	(1.468	> 1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	-	(1.412	> 1.220),	д.у.з.	0.05	г.	-	(1.412	> 1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	-	(1.431	> 1.220),	д.у.з.	0.05	г.	-	(1.431	> 1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	-	(1.422	> 1.220),	д.у.з.	0.05	г.	-	(1.422	> 1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	-	(1.428	> 1.220),	д.у.з.	0.05	г.	-	(1.428	> 1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	-	(1.443	> 1.220),	д.у.з.	0.05	г.	-	(1.443	> 1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.572	<=1.220),	д.у.з.	0.05	г.	+	(0.572	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.553	<=1.220),	д.у.з.	0.05	г.	+	(0.553	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.715	<=1.220),	д.у.з.	0.05	г.	+	(0.715	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.553	<=1.220),	д.у.з.	0.05	г.	+	(0.553	<=1.360)
V. p.	100	и	1000:	д.у.з.	0.1	г.	+	(0.477	<=1.220),	д.у.з.	0.05	г.	+	(0.477	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.479	<=1.220),	д.у.з.	0.05	г.	+	(0.479	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.483	<=1.220),	д.у.з.	0.05	г.	+	(0.483	<=1.360)
V. p.	100	и	100000:	д.у.з.	0.1	г.	+	(0.464	<=1.220),	д.у.з.	0.05	г.	+	(0.464	<=1.360)
V. p.															

В. р. 100000 и 100000: д.у.з. 0.1 г. + (0.622 <=1.220), д.у.з. 0.05 г. + (0.622 <=1.360)
В. р. 100000 и 100000: д.у.з. 0.1 г. + (0.760 <=1.220), д.у.з. 0.05 г. + (0.760 <=1.360)
В. р. 100000 и 100000: д.у.з. 0.1 г. + (1.125 <=1.220), д.у.з. 0.05 г. + (1.125 <=1.360)
В. р. 100000 и 100000: д.у.з. 0.1 г. + (1.020 <=1.220), д.у.з. 0.05 г. + (1.020 <=1.360)

4.2 Проверка параметрических гипотез

4.2.1 Выбор данных

В качестве данных возьмем выборки из распределения Ципфа с параметрами $s = 1.5$, $N = 15$, а также сгенерируем второй набор выборок с параметрами $s = 1.7$, $N = 15$.

```
[26]: zipf_s2 = 2.
zipf_samples2 = generate_samples(zipf_random, s=zipf_s2, N=zipf_N)
zipf_var_series2 = {}
for sample_size, samples in zipf_samples2.items():
    zipf_var_series2[sample_size] = [sorted(sample) for sample in
    ↪samples]
```

4.2.2 Постановка задачи

Дана реализация выборки из какого-либо распределения, вид распределения считаем известным. Рассматриваются две гипотезы: H_0 — реализация выборки соответствует распределению с параметром θ_0 , H_1 — реализация выборки соответствует распределению с параметром θ_1 . Необходимо выбрать одну из этих гипотез и отвергнуть другую.

Необходимо выработать критерий — правило, по которому по реализации выборки мы принимаем либо гипотезу H_0 , либо гипотезу H_1 . Этот критерий делит выборочное пространство \mathbb{X} на \mathbb{X}_0 и \mathbb{X}_1 , соответствующие гипотезам H_0 и H_1 соответственно.

В частности, в нашем случае:

- H_0 : $s = 1.5$
- H_1 : $s = 3$

Ошибка первого рода α — отвержение гипотезы H_0 при ее истинности; ошибка второго рода β — принятие гипотезы H_0 при ее ложности.

$$\alpha = \mathbb{P}\{x \in \mathbb{X}_1 | H_0\}$$

$$\beta = \mathbb{P}\{x \in \mathbb{X}_0 | H_1\}$$

Функция мощности — это вероятность отвергнуть гипотезу H_0 при заданном распределении наблюдений \mathbb{Y} , т.е. функция мощности — это функция от распределения наблюдений \mathbb{Y} :

$$g(\mathbb{Y}) = \mathbb{P}\{x \in \mathbb{Y} | H_0\}$$

В частности, при $\mathbb{Y} = \mathbb{X}_1$ получаем $g(\mathbb{X}_1) = \alpha$

4.2.3 Вычисление функции отношения правдоподобия

$$l(\vec{x}) = \frac{L(\vec{x}, \theta_1)}{L(\vec{x}, \theta_0)}$$

Гипотеза H_0 принимается при выполнении $l(\vec{x}) < c$

Пусть \vec{x} имеет размер n , N - второй параметр распределения Ципфа, тогда:

$$L(\vec{x}, s_i) = \prod_{i=1}^n \frac{x_i^{-s}}{\sum_{j=1}^N j^{-s}} = \left[\sum_{j=1}^N j^{-s} \right]^{-n} \prod_{i=1}^n x_i^{-s}$$

$$\ln L(\vec{x}, s_i) = -n \ln \left[\sum_{j=1}^N j^{-s} \right] - s \sum_{i=1}^n \ln x_i$$

4.2.4 Вычисление критической области/количества материала

$$\begin{aligned} \ln l(\vec{x}) &= \ln L(\vec{x}, s_1) - \ln L(\vec{x}, s_0) = -n \ln \left[\sum_{j=1}^N j^{-s_1} \right] - s_1 \sum_{i=1}^n \ln x_i + n \ln \left[\sum_{j=1}^N j^{-s_0} \right] + s_0 \sum_{i=1}^n \ln x_i = \\ &= n \ln \left[\frac{\sum_{j=1}^N j^{-s_0}}{\sum_{j=1}^N j^{-s_1}} \right] + (s_0 - s_1) \sum_{i=1}^n \ln x_i < \ln c = c' \end{aligned}$$

Полагая $s_0 < s_1$:

$$\sum_{i=1}^n \ln x_i > \left[c' - n \ln \left(\frac{\sum_{j=1}^N j^{-s_0}}{\sum_{j=1}^N j^{-s_1}} \right) \right] \frac{1}{s_0 - s_1}$$

По центральной предельной теореме при $n \rightarrow \infty$: $\frac{S_n - \mu n}{\sigma \sqrt{n}} \sim N(0, 1)$ (μ, σ^2 - мат. ожидание и дисперсия, S_n - сумма выборки)

Посчитаем матожидание и дисперсию случайной величины ξ , которая принимает значения $\ln 1, \ln 2, \dots, \ln N$ с вероятностями, соответствующими распределению Ципфа:

$$\mathbb{P}\{\xi = \ln k\} = \frac{k^{-s}}{\sum_{j=1}^N j^{-s}}$$

$$\mathbb{E}\xi = \sum_{k=1}^N \left[\ln k \cdot \frac{k^{-s}}{\sum_{j=1}^N j^{-s}} \right] = \frac{\sum_{k=1}^N k^{-s} \ln k}{\sum_{j=1}^N j^{-s}};$$

$$\mathbb{E}(\xi^2) = \sum_{k=1}^N \left[\ln^2 k \cdot \frac{k^{-s}}{\sum_{j=1}^N j^{-s}} \right] = \frac{\sum_{k=1}^N k^{-s} \ln^2 k}{\sum_{j=1}^N j^{-s}}$$

$$\mathbb{D}\xi = \mathbb{E}(\xi^2) - (\mathbb{E}\xi)^2 = \frac{\sum_{k=1}^N k^{-s} \ln^2 k}{\sum_{j=1}^N j^{-s}} - \left[\frac{\sum_{k=1}^N k^{-s} \ln k}{\sum_{j=1}^N j^{-s}} \right]^2$$

Итак, при принятии гипотезы H_0 :

$$\sum_{i=1}^n \ln x_i > \left[c' - n \ln \left(\frac{\sum_{j=1}^N j^{-s_0}}{\sum_{j=1}^N j^{-s_1}} \right) \right] \frac{1}{s_0 - s_1}$$

$$\frac{\sum_{i=1}^n \ln x_i - n \cdot \mathbb{E}_0 \xi}{\sqrt{n \cdot \mathbb{D}_0 \xi}} \sim N(0, 1)$$

$$\frac{\sum_{i=1}^n \ln x_i - n \cdot \mathbb{E}_0 \xi}{\sqrt{n \cdot \mathbb{D}_0 \xi}} > \frac{\left[c' - n \ln \left(\frac{\sum_{j=1}^N j^{-s_0}}{\sum_{j=1}^N j^{-s_1}} \right) \right] \frac{1}{s_0 - s_1} - n \cdot \mathbb{E}_0 \xi}{\sqrt{n \cdot \mathbb{D}_0 \xi}}$$

$$\text{Принятие гипотезы } H_0 \text{ происходит в случае } t = \frac{\sum_{i=1}^n \ln x_i - n \cdot \mathbb{E}_0 \xi}{\sqrt{n \cdot \mathbb{D}_0 \xi}} < t_\alpha, \text{ где } t_\alpha \text{ — квантиль}$$

нормального распределения уровня α . Отсюда можно посчитать c' и c .

При принятии H_0 отвергается H_1 ; при отвержении H_0 принимается H_1 . Точного равенства квантилю произойти не может (т.к. происходит переход к абсолютно непрерывному нормальному распределению), но если бы такое случилось, нужно было бы подбросить монетку.

```
[27]: norm_quantiles = {"0.05": 1.645, "0.1": 1.282}

def find_expected(s):
    values = np.arange(1, zipf_N + 1)
    return np.sum(values ** (-s) * np.log(values)) / np.sum(values **
    ↪ (-s))

def find_variance(s):
    values = np.arange(1, zipf_N + 1)
    return np.sum(values ** (-s) * (np.log(values)) ** 2) / np.
    ↪ sum(values ** (-s)) - find_expected(s) ** 2

def find_t(sample, s):
    n = len(sample)
    expected = find_expected(s)
    variance = find_variance(s)
    return (np.sum(np.log(np.array(sample))) - n * expected) / np.
    ↪ sqrt(n * variance)
```

Для реализаций выборок, сгенерированных с параметром $s_0 = 1.5$ (звездочкой помечены ошибки первого рода):

```
[28]: for sample_size, var_series in zipf_var_series.items():
    print("Выборки объема {}".format(sample_size))
    for i, sample in enumerate(var_series):
        print(" " + str(i + 1) + " ", end=" ")
        t = abs(find_t(sample, zipf_s))
        for alpha in ["0.1", "0.05"]:
```

```

        print("{:20}".format(" д.у.з. " + alpha + " принимается
→"), end="")
        if t < norm_quantiles[alpha]:
            print("H0 ({:6.3f} < {:5.3f}) ".format(t,
→norm_quantiles[alpha]), end="")
        else:
            print("H1 ({:6.3f} > {:5.3f}) * |".format(t,
→norm_quantiles[alpha]), end="")
        print("")

```

Выборки объема 5

1: д.у.з. 0.1 H0 (0.199 < 1.282)		д.у.з. 0.05 H0 (0.199 < 1.645)
2: д.у.з. 0.1 H0 (0.459 < 1.282)		д.у.з. 0.05 H0 (0.459 < 1.645)
3: д.у.з. 0.1 H0 (0.834 < 1.282)		д.у.з. 0.05 H0 (0.834 < 1.645)
4: д.у.з. 0.1 H1 (2.025 > 1.282) *		д.у.з. 0.05 H1 (2.025 > 1.645) *
5: д.у.з. 0.1 H0 (0.156 < 1.282)		д.у.з. 0.05 H0 (0.156 < 1.645)

Выборки объема 10

1: д.у.з. 0.1 H0 (0.857 < 1.282)		д.у.з. 0.05 H0 (0.857 < 1.645)
2: д.у.з. 0.1 H0 (0.699 < 1.282)		д.у.з. 0.05 H0 (0.699 < 1.645)
3: д.у.з. 0.1 H1 (2.220 > 1.282) *		д.у.з. 0.05 H1 (2.220 > 1.645) *
4: д.у.з. 0.1 H1 (2.086 > 1.282) *		д.у.з. 0.05 H1 (2.086 > 1.645) *
5: д.у.з. 0.1 H1 (1.555 > 1.282) *		д.у.з. 0.05 H0 (1.555 < 1.645)

Выборки объема 100

1: д.у.з. 0.1 H0 (0.553 < 1.282)		д.у.з. 0.05 H0 (0.553 < 1.645)
2: д.у.з. 0.1 H0 (0.689 < 1.282)		д.у.з. 0.05 H0 (0.689 < 1.645)
3: д.у.з. 0.1 H1 (1.299 > 1.282) *		д.у.з. 0.05 H0 (1.299 < 1.645)
4: д.у.з. 0.1 H0 (0.514 < 1.282)		д.у.з. 0.05 H0 (0.514 < 1.645)
5: д.у.з. 0.1 H0 (0.928 < 1.282)		д.у.з. 0.05 H0 (0.928 < 1.645)

Выборки объема 1000

1: д.у.з. 0.1 H0 (1.103 < 1.282)		д.у.з. 0.05 H0 (1.103 < 1.645)
2: д.у.з. 0.1 H0 (0.353 < 1.282)		д.у.з. 0.05 H0 (0.353 < 1.645)
3: д.у.з. 0.1 H0 (0.193 < 1.282)		д.у.з. 0.05 H0 (0.193 < 1.645)
4: д.у.з. 0.1 H0 (0.691 < 1.282)		д.у.з. 0.05 H0 (0.691 < 1.645)
5: д.у.з. 0.1 H0 (0.233 < 1.282)		д.у.з. 0.05 H0 (0.233 < 1.645)

Выборки объема 100000

1: д.у.з. 0.1 H0 (1.054 < 1.282)		д.у.з. 0.05 H0 (1.054 < 1.645)
2: д.у.з. 0.1 H0 (0.397 < 1.282)		д.у.з. 0.05 H0 (0.397 < 1.645)
3: д.у.з. 0.1 H0 (0.038 < 1.282)		д.у.з. 0.05 H0 (0.038 < 1.645)
4: д.у.з. 0.1 H1 (2.219 > 1.282) *		д.у.з. 0.05 H1 (2.219 > 1.645) *
5: д.у.з. 0.1 H0 (0.527 < 1.282)		д.у.з. 0.05 H0 (0.527 < 1.645)

Для реализаций выборок, сгенерированных с параметром $s_1 = 2$ (звездочкой помечены ошибки второго рода):

```

[29]: for sample_size, var_series in zipf_var_series2.items():
        print("Выборки объема {}".format(sample_size))
        for i, sample in enumerate(var_series):
            print("" + str(i + 1) + "", end="")
            t = abs(find_t(sample, zipf_s))
            for alpha in ["0.1", "0.05"]:
                print("{:20}".format(" д.у.з. " + alpha + " принимается
→"), end="")
                if t < norm_quantiles[alpha]:

```

```

        print("H0 ({:6.3f} < {:5.3f}) * |".format(t,
↪norm_quantiles[alpha]), end="")
    else:
        print("H1 ({:6.3f} > {:5.3f}) ".format(t,
↪norm_quantiles[alpha]), end="")
    print("")

```

Выборки объема 5

```

1: д.у.з. 0.1 H1 (1.365 > 1.282) | д.у.з. 0.05 H0 (1.365 < 1.645) *
2: д.у.з. 0.1 H0 (0.020 < 1.282) * | д.у.з. 0.05 H0 (0.020 < 1.645) *
3: д.у.з. 0.1 H0 (0.990 < 1.282) * | д.у.з. 0.05 H0 (0.990 < 1.645) *
4: д.у.з. 0.1 H0 (0.156 < 1.282) * | д.у.з. 0.05 H0 (0.156 < 1.645) *
5: д.у.з. 0.1 H0 (1.089 < 1.282) * | д.у.з. 0.05 H0 (1.089 < 1.645) *

```

Выборки объема 10

```

1: д.у.з. 0.1 H0 (0.323 < 1.282) * | д.у.з. 0.05 H0 (0.323 < 1.645) *
2: д.у.з. 0.1 H1 (1.470 > 1.282) | д.у.з. 0.05 H0 (1.470 < 1.645) *
3: д.у.з. 0.1 H0 (0.293 < 1.282) * | д.у.з. 0.05 H0 (0.293 < 1.645) *
4: д.у.з. 0.1 H1 (1.821 > 1.282) | д.у.з. 0.05 H1 (1.821 > 1.645)
5: д.у.з. 0.1 H0 (0.870 < 1.282) * | д.у.з. 0.05 H0 (0.870 < 1.645) *

```

Выборки объема 100

```

1: д.у.з. 0.1 H1 (2.765 > 1.282) | д.у.з. 0.05 H1 (2.765 > 1.645)
2: д.у.з. 0.1 H1 (4.113 > 1.282) | д.у.з. 0.05 H1 (4.113 > 1.645)
3: д.у.з. 0.1 H1 (4.292 > 1.282) | д.у.з. 0.05 H1 (4.292 > 1.645)
4: д.у.з. 0.1 H1 (3.051 > 1.282) | д.у.з. 0.05 H1 (3.051 > 1.645)
5: д.у.з. 0.1 H1 (4.433 > 1.282) | д.у.з. 0.05 H1 (4.433 > 1.645)

```

Выборки объема 1000

```

1: д.у.з. 0.1 H1 10.718 > 1.282 | д.у.з. 0.05 H1 (10.718 > 1.645)
2: д.у.з. 0.1 H1 11.329 > 1.282 | д.у.з. 0.05 H1 (11.329 > 1.645)
3: д.у.з. 0.1 H1 11.119 > 1.282 | д.у.з. 0.05 H1 (11.119 > 1.645)
4: д.у.з. 0.1 H1 10.121 > 1.282 | д.у.з. 0.05 H1 (10.121 > 1.645)
5: д.у.з. 0.1 H1 10.908 > 1.282 | д.у.з. 0.05 H1 (10.908 > 1.645)

```

Выборки объема 100000

```

1: д.у.з. 0.1 H1 108.607 > 1.282 | д.у.з. 0.05 H1 (108.607 > 1.645)
2: д.у.з. 0.1 H1 108.242 > 1.282 | д.у.з. 0.05 H1 (108.242 > 1.645)
3: д.у.з. 0.1 H1 109.144 > 1.282 | д.у.з. 0.05 H1 (109.144 > 1.645)
4: д.у.з. 0.1 H1 108.436 > 1.282 | д.у.з. 0.05 H1 (108.436 > 1.645)
5: д.у.з. 0.1 H1 109.736 > 1.282 | д.у.з. 0.05 H1 (109.736 > 1.645)

```

$$\frac{\sum_{i=1}^n \ln x_i - n \cdot \mathbb{E}_0 \xi}{\sqrt{n \cdot \mathbb{D}_0 \xi}} < t_\alpha, \text{ откуда } \sum_{i=1}^n \ln x_i < t_\alpha \sqrt{n \cdot \mathbb{D}_0 \xi} + n \cdot \mathbb{E}_0 \xi = A$$

$$\frac{\sum_{i=1}^n \ln x_i - n \cdot \mathbb{E}_1 \xi}{\sqrt{n \cdot \mathbb{D}_1 \xi}} < \frac{A - n \cdot \mathbb{E}_1 \xi}{\sqrt{n \cdot \mathbb{D}_1 \xi}} = t_\beta$$

$$t_\beta = \frac{A - n \cdot \mathbb{E}_1 \xi}{\sqrt{n \cdot \mathbb{D}_1 \xi}} = t_\alpha \sqrt{\frac{\mathbb{D}_0 \xi}{\mathbb{D}_1 \xi}} + \sqrt{n} \frac{\mathbb{E}_0 \xi - \mathbb{E}_1 \xi}{\sqrt{\mathbb{D}_1 \xi}}$$

$$n = \left\lceil \left(t_\beta - t_\alpha \sqrt{\frac{\mathbb{D}_0 \xi}{\mathbb{D}_1 \xi}} \right)^2 \frac{\mathbb{D}_1 \xi}{(\mathbb{E}_0 \xi - \mathbb{E}_1 \xi)^2} \right\rceil$$
 — количество материала, необходимое для различения гипотез H_0 и H_1 с заданными ошибками 1 и 2 рода

```
[30]: def find_n(s0, s1, t_alpha, t_beta):
        return int(np.ceil(
            ((t_beta - t_alpha * np.sqrt(find_variance(s0) /
→find_variance(s1))) ** 2)
            * find_variance(s1) / ((find_expected(s0) -
→find_expected(s1)) ** 2)
            ))

[31]: for s1 in (zipf_s + np.logspace(-3, -0.1, 10)):
        n = find_n(zipf_s, s1, norm_quantiles["0.05"], norm_quantiles["0.
→1"])
        print("Параметры {} и {:.3f}, для различения с a = {} и b = {}
→нужно кол-во материала {}".format(zipf_s, s1, 0.05, 0.1, n))
```

```
Пар-ы 1.5 и 1.501, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 193600
Пар-ы 1.5 и 1.502, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 44041
Пар-ы 1.5 и 1.504, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 10054
Пар-ы 1.5 и 1.509, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 2313
Пар-ы 1.5 и 1.519, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 541
Пар-ы 1.5 и 1.541, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 131
Пар-ы 1.5 и 1.586, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 34
Пар-ы 1.5 и 1.680, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 11
Пар-ы 1.5 и 1.878, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 4
Пар-ы 1.5 и 2.294, для разл-я с a = 0.05 и b = 0.1 нужно кол-во материала 3
```